Responding to the Call of Things
A Conversational Approach to 3D Animation Software

A project submitted in fulfilment of the requirements for the degree of Doctor of Philosophy.

Georgina Alicia Moore

BA (Fine Arts), Curtin University, Perth
Postgraduate Diploma (Animation and Interactive Media), RMIT University,
Melbourne Master of Arts (by Research Project), RMIT University, Melbourne

School of Media and Communication

College of Design and Social Context

RMIT University

October 2016
Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the project is the result of work which has been carried out since the official commencement date of the approved research program; any editorial work, paid or unpaid, carried out by a third party is acknowledged; and, ethics procedures and guidelines have been followed.

Georgina Moore
October, 2016.
Acknowledgements

I would like to thank my supervisor Dr Chris Barker for his generous encouragement and support over the past 4 years. One of the (many) enjoyable aspects of this research has been getting to know Chris. I would also like to thank my associate supervisor Dr Adam Nash for his insightful feedback on drafts of this document. Dr Stefan Greuter was my first associate supervisor and his generous support in the early stages of this research is greatly appreciated.

For their friendship and inspiring conversation I would like to thank all my research colleagues at RMIT including Amani Naseem, Robert Cercos, Betty Sargeant, Sara Jane Pell, Tom Penney, Matthew Riley, Andy Buchanan and Helmut Munz. For the duration of this research my workspace has been situated in the Exertion Games lab at RMIT University and I am grateful to Floyd Meuller and everyone in the Lab for providing a collegial working environment.

I would like to thank all my friends and family for their encouragement and support over the past four years and will mention some of them by name. As a friend and an experienced researcher, Dr Robyn Barnacle’s support and advice on various issues have been invaluable throughout the course of this research. Dr Pia Ednie-Brown is also friend and a researcher and, although intermittent, our discussions over many years have always been inspiring and influential. In the early days of this research Goran Popovski shared with me some of his extensive knowledge of computer programming, helping and inspiring me to pursue a greater technical understanding of 3D software.

I would especially like to thank my parents, Donna Dean and Colin Moore, who have been consistently supportive of my creative projects and personal endeavours throughout my entire life. This research project wouldn’t have been possible without them. A special thank you also to my partner, Hazel Porter, for her ongoing support and for showing me by example how much can be achieved with hard work and dedication.

There are several people whose efforts have directly influenced the final form of this document. I would like to acknowledge Michael Day who provided copy-editing; Verity Campbell for spending several enjoyable hours discussing my research themes and how best to communicate them; and Chris Knowles for his technical expertise, helping me to lay out this document in html.

At the start of this research I looked around me for an object that could serve as the inspiration for a number of different 3D animations projects. I could have chosen a car, a teapot, or a billiard ball but instead I chose my dog, a whippet called Ginger. It was a convenient choice because Ginger has been with me most of the time throughout this research. Ginger died a few months ago and, as I finish writing this document, she is no longer beside me in her bed. One of many collaborative partners involved in this research, this document is dedicated to her.
# Table of contents

- Abstract - page 6
- 1. Introduction - page 7
  - Research aims and methods
  - Background
  - Document outline
  - Ginger and Default Whippet
    - The sketching process
    - Drawing as a way of seeing
    - A typical 3D workflow
    - Turning away from the local and contingent
    - Filtering out complex and unexpected encounters
    - The importance of chance and accident
  - From physical artefacts to symbolic representations
    - The reductive capacity of digital technology
    - Creation replaced by selection from a menu
    - The progression toward realism
  - 3D animation and technological Enframing
    - Digital animation before and after software
- 2. Context of practice - page 20
  - Early computer graphics research
  - Emergence of commercial software
  - Technical Rationality
  - Back to basics and the low-poly aesthetic
  - Media transparency
    - NPR
    - Photorealism
  - Showing the making process
  - Towards a transparent interface
  - The glitch aesthetic
    - Glitches as a creative opportunities
  - Becoming alive to the accident
  - Context of practice: Conclusion
- 3. Things themselves - page 30
  - Objects and things
  - Being present to things
  - The thingliness of digital objects
  - Beginning in poverty
  - Forgetting the objects before you
  - Accidents and essences
  - Object Oriented Programming
  - Wonder in the face of the world
  - Objects in the act of appearing
  - The world of perception
  - The arbitration of conflict
  - Things Themselves: Conclusion
- 4. Experimental animations - page 40
  - Research methods
    - Reflection on prior practice
    - Taking notes during production
  - Technical notes
    - Description of hardware and software
Scripting
  - Description of practice-led enquiry
    - Playing with Software
    - Playing with history
    - Traditional animation principles: Conformity and disruption
    - Animation as relation
    - Modelling as animation
    - Working from sketches
    - Working from Life
    - Colour as Light and Geometry as Shadows

5. Outcomes and findings - page 66
  - A conversational approach
  - The thingliness of digital objects
  - Beginning in poverty and seeing essences
  - The world of perception
  - Mapping process to outcome

6. Discussion - page 70
  - The paradox of expression
  - Practice as a stylistic exchange
  - Tolerance for the incomplete

7. Conclusion - page 76

Bibliography - page 77

List of Figures - page 83

Definition of key terms - page 86

Appendices - page 87
Abstract

Makers of Zbrush, one of many popular 3D computer animation packages, say that their software “allows you to create limited only by your imagination” (Pixologic, 2015). The idea that 3D software is a creative tool at the service of a proficient user is common in the 3D animation community, but it is a sentiment that downplays the agency of the software. Drawing from a range of theoretical sources, and through practical investigations, I examine and challenge pernicious notions of transparency, representation and the status of digital media objects.

Standard approaches to 3D software emphasise efficiency and control. This research seeks a more empathic mode of engagement; an actively receptive comportment which I refer to as a “conversational” approach. How can a 3D animator adopt a conversational approach to 3D tools to extend the creative capacities of the medium and move beyond its implicit design biases?

To answer this question I have created a series of short experimental animations using commercial tools in unusual ways. I have also coded custom tools that enable and encourage unorthodox practices. Rather than being designed in advance, experimental animations emerge from my response to everyday physical things, including my response to computer glitches (which I regard as aesthetic suggestions).

Phenomenological enquiry is my starting point, but the finished animations embody a distinctly non-human gesture reminiscent of digital objects and object oriented programming (OOP). Findings from this research indicate that, rather than a means to an end, 3D animation is better conceived as a stylistic exchange between a variety of things – including the software, the hardware, the user and other objects. Embracing 3D animation as a conversational process reveals capabilities of the software that extend beyond its design, suggesting that (like all things) digital objects have the capacity to continually surprise.

As computer software becomes an increasingly prevalent and powerful force in our lives, it becomes increasingly important to recognise that algorithmic models are not simply tools for organising or explaining; they are (potentially reductive) ways of revealing the world. The challenge facing 3D animation practitioners is how to stay in the messy zone where situations and entities are encountered in their complexity. My research finds that exploring the generative potential of material practices can help us avoid the reductive potential inherent in digital tools.
1. Introduction

This research follows the process of discovery and reflection that proceeded from my first use of 3D computer animation software in 2001, and my subsequent familiarisation with the software.

Research aims and methods

The aim of this research is to expand the practices and visual styles associated with computer graphics (CG) software by developing approaches to 3D animation that avoid reductive frameworks (such as standardised workflows and abstract mathematical models).

Using theoretical and practical investigations, I explore assumptions and biases implicit in the design, marketing and orthodox use of 3D animation tools. Informed by these investigations, as well as by reflection on my prior practice, the research proceeds via a practice-led enquiry that involves creating a series of short experimental 3D animations, as well as several customised 3D animation tools.

Background

After many years using traditional art media such as oil paint, charcoal and pencil, in 2001, while studying at RMIT University, I was introduced to 3D computer animation software. During this time I learnt how to use 3D Studio Max and made a short animated film called Drive (Moore, 2001). Making this film was a learning process and I didn’t have much control over the animations that I produced. I followed some tutorials, but mostly I experimented/played with the 3D tools and responded to imagery that the software came up with. Even if it wasn’t what I expected, I might accept what I saw, revise it or reject it altogether.

The fact that I had to compromise my “artistic vision” because of my lack of skill didn’t, at the time, seem like a good thing. But looking back on it now it seems that my unorthodox use of tools, my loose pictorial goals, and the way that my focus was on responding to what was given combined to form an approach to animation which sometimes resulted in imagery more intriguing than I could have designed in advance.

Figure 1.1: Stills from “Drive” (Moore, 2001).

Excited by the seemingly infinite potential of this new medium, I spent subsequent years working professionally in the 3D animation industry and learnt standard 3D animation procedures. Eventually, I felt equipped to convincingly illustrate any subject matter. However, this mastery, autonomy and empowerment came at a cost. Having gained a high level of technical proficiency, my 3D animations no longer felt like the result of a collaborative process because I was now the one calling the shots. It seemed that the software was now at the service of my creative design, my ideas and my imagination. However, instead of being full of discovery, the making process had become a necessary (and often tedious) set of steps required to reach a specified goal. A finished work either achieved this goal or it fell short; the work seldom exceeded or altered my expectations. Over time, I became aware of something else happening, which was indistinct, but disquieting, and unnerving.

My hard-earned knowledge of standard 3D procedures seemed to define my experience of the software, making it hard to play or experiment. It also influenced the way that I engaged with the world around me.

This research follows the process of discovery that proceeded from that initial spark of disquiet, inviting me to look below the software’s glossy surface.
Document outline

Section 1: Introduction explains the aims, methods and motivation for my research. This section introduces themes relevant to my intuition that computer software can have a reductive role in creative practice.

Section 2: Context of practice discusses a selection of 3D users and digital artists who expand the visual styles associated with computer graphics (CG) software by exploring qualities inherent in the medium and by being open to opportunities that emerge in the process of making. Unlike orthodox approaches to CG software, these artists don’t seek media transparency, either in their tools or in their work. We hear from artists using traditional media (such as Francis Bacon and Frederick Franck) who emphasise the importance of fostering an actively receptive comportment through their work, and who adopt particular creative strategies to do so. The work of digital artists discussed in this section indicates that an actively receptive comportment might also be important for those aiming to expand practices and visual styles associated with 3D software.

Section 3: Things themselves further examines the idea of an actively receptive comportment through a discussion of phenomenology. The basic project of phenomenology is to avoid the imposition of reductive abstract models and to “return to the things themselves” (Husserl, 2001, p. 168). This is a call to explore the inexhaustible richness of our everyday experience of things and notice what can’t be described in words, explained using scientific concepts, or represented using computational [algorithmic] models. The things with which my research is concerned include everyday physical things in the world around me as well as 3D software and the 3D animations which this software can produce. How, and to what extent, can I approach these things without being limited by standard practices and algorithmic models? The work of phenomenologist philosophers Edmund Husserl, Martin Heidegger and Maurice Merleau-Ponty together with that of the artists discussed in Section 2 and 3, suggests a number of approaches that will inform my practice-led enquiry.

In Section 4: Experimental animations I describe a practice-led enquiry aimed at developing new approaches to 3D computer graphics software. Drawing on themes outlined in previous sections, this practical enquiry involves creating a series of short 3D animations which cultivate an actively receptive comportment. Rather than focusing on achieving predictable and repeatable results, in these experimental animations I often work with computer glitches, treating them as aesthetic suggestions, to which I intuitively respond.

In many experimental animations I use 3D software to explore perceptual experience; sometimes working from pencil sketches and sometimes working directly from life. In addition, I explore ways to remap the user’s actions to a visible outcome so that activities involved in making a 3D animation are evident to viewers of a finished work.

Section 5: Outcomes and findings discusses outcomes and findings from the practice-led enquiry, covering 3D animations, customised 3D animation tools, and a number of creative strategies used throughout the project. These custom tools and creative strategies all involve a shift in focus from control to response. Some tools and strategies involve responding primarily to suggestions from the software (including suggested form and suggested movement) while others respond primarily to everyday physical things in my local environment. Whether responding primarily to digital things or to physical things, 3D animation practice is appreciated as a type of conversation (or collaboration) between the user, the software and other physical things. In addition to 3D animations, custom tools and creative strategies described in this document, a major contribution of this research is to conceptualise a conversational approach to 3D animation practice which involves a shift in comportment from control to response.

Section 6: Discussion outlines an expanded conception of style, arguing that all things exhibit a style and it is useful to think of 3D animation as a stylistic exchange between the style of an animator and the style of other things. Reflecting on the experimental animations created in this research I notice that, in addition to my style of seeing, finished works also embody a distinctly non-human gesture reminiscent of digital objects and object oriented programming (OOP). This insight leads to a short discussion of object oriented ontology (OOO) which conceptually situates humans on the same level as other things (including physical, digital, real and imagined things). Like an expanded conception of style, OOO encourages us to think in terms of collectives and entanglements.

Ginger and Default Whippet
Throughout this document standard 3D animation practices and procedures are primarily discussed through an account of Default Whippet (Figure 1.2), a 3D animation completed at the outset of this research.

Like many of the animations discussed in this document, Default Whippet is based on my dog, a whippet called Ginger. My constant companion throughout this research, Ginger is lying beside me in her bed as I type this document and every now and then I turn away from the screen and move my hand from the keyboard to stroke her fur.

The sketching process

Before commencing Default Whippet, I took my dog Ginger to the park and sketched her as she wandered around or lay on the grass. Observational sketching is not a typical way to start a 3D animation project and ultimately my drawings of Ginger were not very useful in the creation of the Default Whippet 3D animation. However, as a way of interrogating activities involved in standard 3D practices, observational sketching provides a useful point of comparison.

Requiring only basic tools – paper and pencil or pen – sketching is typically a simple activity but it can reveal a world of infinite complexity. While sketching, I often notice the way that shifts in attention or mood change what I see. I also notice that the activity of sketching is itself a way of seeing and it is different from how I normally see the world. While sketching I pay attention to what things look like rather than moving among them and hardly noticing them at all. And if I were to swap my pencil for paint brush, I would also see the dog differently (e.g. I might notice the texture, colour or tonal variation across Ginger’s coat rather than focusing on her silhouette). While sketching Ginger in the park, I visually explore her contours through the pencil which has become an extension of my body, i.e. my eyes travel across her form as I intuitively make marks on the page. I am mostly seeing and responding to the dog but I also look down at the page and respond to the sketch as an emerging figure or gestalt. Ginger moves around, but even when she’s still I notice her muscles twitching and her nose sniffing the air; it’s obvious that she’s making sense of a world that I can’t even imagine.

Drawing can be a way of seeing or a mode of contemplation – but what about the resulting artefact; the drawing itself? The activity of sketching can open onto a world which is so complex, elastic and inexhaustibly rich that there’s no question of trying to pin it down or “accurately” represent it with graphite on paper. Sketches are often sketchy and this is especially true when, like Ginger, the objects of study are in constant motion. The finished sketch is indicative or allusive; it is an attempt to capture something which is obviously out of reach or beyond my grasp. In perceptual experience there is always more to see and I have the feeling that what I am seeing could always be different.
When it comes to the drawing as an artefact, qualities of sketchy incompleteness can hint at more than is explicitly depicted and hence leave the work open to interpretation. The sketch, like perception, is always incomplete.

Drawing as a way of seeing

In his book *The Zen of Seeing: Seeing/Drawing as Meditation* visual artist Frederick Franck describes his drawing practice as a mode of contemplation in which the world is experienced afresh at each moment. He says, “Drawing is the discipline by which I constantly rediscover the world. I have learned that what I have not drawn I have never really seen, and that when I start drawing an ordinary thing I realize how extraordinary it is.” (Franck, 1973, p.6) Franck says that each of his drawings is an “adventure” for which he can’t predict the outcome (Franck, 1973, p.6). He believes that it’s important to avoid drawing systems or recipes (such as those found in a book on “how-to” draw horses) because, although they may be useful for a “picture manufacturer”, they deprive you of knowing what things really look like; they stop you from really seeing (Franck, 1973, p.55). He says:

> In order to draw a horse, draw horses until you practically become a horse – not “horses in general” but that particular horse you are drawing at a given moment. Until you feel the tense curving of its neck in your own neck! (Franck, 1973, p.55)

![Figure 1.5: Drawing of a horse by Frederick Frank. Sourced from “The Zen of Seeing” (Franck, 1973, pp. 56–57); Copyright 1973, Frederick Frank.](image)

Of importance to Franck is the particular state of awareness that he cultivates through his practice and it’s clear from the quote above that, for him, drawing involves an empathetic kind of engagement. Franck calls his practice Seeing/Drawing and insists that for him seeing and drawing are not two distinct activities but rather “one single undivided act” (Franck, 1973, p.ix). We could say that Franck sees (understands, describes or explores) things (such as a horse, a landscape, a human figure) through his medium (pencil and paper). A 3D animator also sees things through their medium but it seems to involve a different kind of seeing.

A typical 3D workflow

*Default Whippet* (Figure 1.2) was made using a 3D character animation workflow which involves several distinct steps including *modelling, texturing, rigging* and animation (see Figure 1.6).

![Figure 1.6: This diagram is comprised of stills from Default Whippet and illustrates some of the steps involved in a standard 3D character animation workflow.](image)

Observational sketches of Ginger were ultimately not very useful in the creation of the *Default Whippet* 3D animation because they are idiosyncratic, inaccurate and incomplete. In accordance with the standard 3D character animation workflow (illustrated in Figure 1.6), I needed to create the dog at the origin of the Cartesian grid and in a generic or default pose. This means that the dog needs to be looking straight ahead with limbs straight and body symmetrical. This symmetrical pose can be seen in Figure 1.7 which shows the finished *Default Whippet* model.
To achieve this, I engaged the help of a colleague who photographed Ginger while I physically held her in position. I then manipulated the photographs using Photoshop, creating a series of composite images which depict Ginger in a symmetrical pose as seen from the side, top and back (Figure 1.8). Collectively, reference images such as these are commonly known as a model sheet, and they act like a set of architectural plans or blueprints for the creation of a virtual mesh.

Sometimes imported into the software’s viewport, model sheet images are often used as a guide for building a mesh. When sketching, drawing and seeing are one activity and each drawing is an adventure. When animating with 3D software, by contrast, we are often working toward a predefined image.

As I progressed with Default Whippet I soon found that my photos of Ginger didn’t give me all the detail that I needed and so, like most 3D projects, I used the Google search engine to download a collection of source material from the internet. Figure 1.9 shows a sample of this collection which includes digital photographs, movies and anatomy diagrams. I referred to these images while modelling the dog and also during the rigging phase of the project, when I felt the need to study dog skeletons, musculature and anatomy as well as a variety of common whippet poses. Upon reflection, it seems strange that, throughout Default Whippet, I constantly referred to my vast collection of digital files (including downloaded images and movies as well as my digital photos of Ginger), but I hardly looked directly at the dog who lay right beside me as I worked.

Turning away from the local and contingent

Instead of responding to the dog in front of me, in Default Whippet I built a virtual world (or at least a virtual dog) based on digital reference. As described by Media theorist Douglas Rushkoff, this is a move “away from the local and toward dislocation” (Rushkoff, 2010, p. 41). Rushkoff says that this is one of the biases implicit in digital media. He explains that digital networks:

work from far away, exchanging intimacy for distance. This makes them terrifically suitable for long-distance communication and activities, but rather awful for engaging with what – or who – is right in front of us. (Rushkoff, 2010, p.41)
As an example of digital media’s bias away from the local, Rushkoff describes our obsession with smart phones and social networking sites like Facebook; he was one of the first to offer a now-familiar warning that these digital media have the capacity to alienate us from real world friendships (Rushkoff, 2010, p.43). Rushkoff’s warnings are relevant to contemporary creative practitioners because digital media (including computer software) impacts upon the way that practitioners engage with things in their local environment.

In addition to my collection of digital images, the production of Default Whippet was guided by the software’s help menu and information gained via online forums and tutorials. Throughout production I worked in a variety of locations; sometimes at university using a desktop computer, other times at home, and occasionally I took my laptop to a local cafe. My location made little difference to the outcome of a particular working session; it didn’t matter what was around me because I was absorbed in the screen world and was largely oblivious to my physical surroundings. For 3D users, as long as the software is installed and there is a stable internet connection, it’s easy to conclude that our physical location and context is not relevant.

**How can a 3D user engage with their local physical environment? Instead of manipulating a dog into a default pose and then turning away from her completely, what other options are available?**

As well as turning away from the immediate physical environment in favour of internet images, 3D animation devalues the local and the particular by assuming that processes should be repeatable and that each work session should be exchangeable with the next. This assumption downplays contingent and contextual features of a particular practice situation including computer glitches, or surprise outcomes of any kind. As with all complex workflows, I encountered a number of surprising and intriguing images while working on Default Whippet. Some were the result of auxiliary processes (e.g. the mosaic design surrounding the digitally painted texture map in Figure 1.10), while others were the result of glitches or mistakes (e.g. the wildly deformed model shown in Figure 1.11 is the result of a skinning error, i.e. an error which arises when the mesh is not sufficiently bound to the underlying joint). These surprising images felt significant because they were not entirely random and they were also not entirely preconceived or designed in advance. I could not easily make sense of the images but they intrigued me.

![Figure 1.10: Screenshot from Default Whippet 3D animation showing digitally painted texture map.](image)

Surprising images (and movements) almost always occur in 3D animation production and they are often ambiguous – it’s hard to say exactly what they depict. It’s also hard to say where these images and movements come from: we can’t easily attribute them to the animator, to the software or to the computer hardware. It is common for a 3D animator to be dismissive of these outcomes and to focus instead on imagery that is deliberate, repeatable and easy to understand.

*Promoting the assumption that surprising and ambiguous outcomes are mistakes to be avoided is one of the ways in which 3D software filters out complex and unexpected encounters (for both users and for viewers).*

![Figure 1.11: Screenshot from Default Whippet 3D animation showing skinning error.](image)

Filtering out complex and unexpected encounters

Pariser’s book examines the phenomena of web personalisation experienced by users of internet search engines such as Google, and social networking sites such as Facebook. According to Pariser, the result of web personalisation is that people are increasingly unlikely to encounter views other than their own (Pariser, 2011). Pariser worries that “In the filter bubble there’s less room for chance encounters that bring insight and learning” (Pariser, 2011, p.15).

As media consumers, we have always sort information that feeds our own interests and, prior to the ubiquity of digital media, we probably subscribed to a particular newspaper and changed the TV channel to avoid content that didn’t interest us. So what, if anything, makes the internet filter bubble different to prior media? For Pariser the reductive capacity of the filter bubble is more insidious than prior media because it is invisible (i.e. we’re not aware of its workings) and it is automatic (i.e. we don’t chose to turn it on) (Pariser, 2011, p.10). Collectively these features make it easy for us to assume that digital media are neutral, unbiased or transparent.

It seems that the filter bubble can cocoon us in a cosy and familiar world and, as long as we find what we (think we) are looking for, we probably feel that the world is at our fingertips. For users, 3D software can also provide a kind of cocoon because a turn away from the local physical world and adherence to standardised workflows can mean that the user is not open to complex and unexpected encounters (i.e. unexpected turns in their work). The 3D animator usually aims for an efficient production process and avoids surprises but, for many creative practitioners, surprise outcomes are crucial to their work (Bolt, 2007; Bond, 2012; Dijk, 2012; Kentridge, 2012; Sylvester, 1975).

The importance of chance and accident

![Figure 1.12: Francis Bacon “Three Studies of Lucian Freud” (detail). Oil on canvas.](image)

Emphasising the importance of chance and accident in his work, twentieth century painter Francis Bacon says “Half my painting activity is disrupting what I can do with ease” (Sylvester, 1975, p.91). Bacon uses existing painting techniques but is “trying to make out of them something that is radically different to what those techniques have made before” (Sylvester, 1975, p.107) and his paintings walk a fine line between abstraction and figuration. He often paints the human figure but he never renders it in detail and never in a conventional manner, instead Bacon’s figures are contorted almost beyond recognition and they seem to emerge out of and twist back into the paint. Bacon’s aim is to present the viewer with recognisable imagery while avoiding illustration and, rather than conceiving of the outcome in advance and working toward a clear goal, it’s important for Bacon that he stumble’s toward an image (Sylvester, 1975, p.17). According to Bacon, it’s important for an artist not to proceed in an entirely predefined, formulaic or rational manner “because the moment you know what to do, you’re making just another form of illustration” (Sylvester, 1975, p.59) and, in order to disrupt his own illustrative habits, Bacon employs a variety of methods such as using a very large brush (Sylvester, 1975, p. 16), throwing paint at the canvas or working while he’s frustrated or drunk (Sylvester, 1975, p.91). Through an exploration of qualities inherent in the paint, he elicits surprising outcomes and then responds to these outcomes in an intuitive manner. Like Franck, Bacon approaches each work as an adventure. Franck pays particular attention to things in the world around him (people, animals, and the landscape), while Bacon is more obviously attentive to the physical properties of paint.

For both Franck and Bacon, deliberately avoiding established recipes or conventions and being open to the complexities of material interactions helps them escape habitual ways of seeing the world.

*Can 3D animators aiming to extend the language of their (digital) medium learn something from practitioners such as Bacon and Franck?*

When Franck talks about really seeing, he is alluding to an empathetic kind of engagement where he is responding to things based on feel rather than calculation. This type of response is difficult in 3D animation because standard workflows often involve creating a mesh out of context and according to a strict plan.
I created a model of Ginger at the “centre of the world”, in component parts and in a default pose (Figure 1.7). This ordered approach sucked the life out of the dog, but what other alternatives were available? In a pencil sketch or a painting, traces of an artist’s intuitive actions are often visible as multiple overlapping outlines or brushstrokes. It is harder for a 3D user to respond to things in an intuitive manner and (even if this is achieved) intuitive and iterative responses are not usually visible in a finished work.

3D software is a digital medium and a 3D animation doesn’t (by default) provide a record of the artist’s actions in the same way that a drawing or painting does.

From physical artefacts to symbolic representations

In a 2013 article “Media after Software”, media theorist Lev Manovich wrote:

While earlier reproduction technologies such as woodblock printing, moveable type printing, lithography, and photography stored media in ways accessible to bare senses … the electronic media technologies of the late 19th century abandoned these formats in favour of an electrical signal. Simultaneously, they also introduced a fundamentally new dimension of media – interface (i.e. the ways to represent and control the signal). (Manovich, 2013, abstract).

The shift to digital data and media software 100 years later extends this principle further. With digital computers, data is encoded as sets of numbers. This data is only really accessible via software applications because the software translates the data into sensory representations.

With computer graphics software, the way that a user’s actions are mapped in real-time to a visible screen based image is largely defined by human-computer interaction (HCI) researchers, software designers and programmers. This means that the “feel” of the medium, i.e. the “properties” that we experience when we use digital media (including 3D animation software) are largely defined by the software interface and, ultimately, by software and hardware developers.

Unlike a chemical based photograph, a pencil sketch or an oil painting, a digital image or computer animation has no causal or indexical link to the “real”, physical world (Mitchell, 2010, p.43). We can think of a photograph as the trace of light on film, a sketch as the trace of a pencil across a page, and a painting as the result of paint having been physically applied to a canvas. But with a digital image the link between a physical process and a visible outcome is looser, malleable – one could even argue, non-existent. For a 3D user the arbitrary mapping between process and outcome is perhaps best exemplified by the ongoing development of non-photorealistic rendering (NPR) techniques.

Operating in the same manner as stylistic filters found in image editing programs such as Photoshop, NPR rendering algorithms are often designed to achieve the look of traditional art media such as pastel, pencil or paint (Botkin, 2009; Meier, 1996; Shugrina, Betke, & Collomosse, 2006). NPR algorithms allow the 3D user to render the same virtual mesh in a variety of different ways which means that the same production process (i.e. the same physical actions performed by the user) can achieve entirely different visual outcomes. In this way the style of a finished 3D render is the result of aesthetic decisions/choices made by the user, rather than the inevitable outcome of particular physical actions or activities.

The sheer versatility of 3D software (and the fact that it can simulate the look of other media), coupled with an emphasis on deliberate user choices/decisions, encourages us to regard the production process as a means to an end, rather than as a series of intuitive responses to a complex and evolving practice situation.

With this conception of the role of process comes the assumption that it’s the artist’s ideas or explicit intentions that count.
The reductive capacity of digital technology

Makers of ZBrush, one of many 3D animation packages, say that ZBrush allows you (the user) to “create limited only by your imagination” (Pixologic, 2015) and renowned computer animation studio Pixar proudly suggest that they use technology only as a means to an end (Jones, 2014). These sentiments of empowerment are echoed throughout the 3D animation community, from large companies to sole practitioners, but they downplay the agency of the software and promote the idea that computer software is at the service of a proficient user.

As 3D users we are encouraged by software manufacturers to feel empowered, to fully express our “creativity” (c.f. Digital-Tutors, 2014), but this narrative of empowerment masks the complexity of our relationship with digital tools.

When we use a smart phone, tablet, laptop or desktop computer we are almost always using software which has been carefully designed to make us feel like we are in control (Autodesk, n.d.-b; Microsoft, n.d.-a). Collectively, digital hardware and software are marketed as products that empower us to achieve our goals whether they are to stay in touch, to hook up, to learn or to create (Autodesk, n.d.-a; Microsoft, n.d.-b) There’s no doubt that advances in digital technology enable us to do things that we couldn’t do before. However, the computer software and hardware that we use don’t simply open us to new experiences and expand our capabilities; their effects also flow in the other direction, impacting the way that we see the world.

N. Katherine Hayles' theory of technogenesis suggests that humans have coevolved with tools and technologies (Hayles, 2012). Hayles suggests that it’s never been about simply using technology with the users remaining unchanged.

Theories of distributed and extended cognition suggest that human cognition doesn’t just involve the brain or even just the body (Hollan, Hutchins, & Kirsh, 2000). These theories describe how we think with and through technologies and media. Whether typing this document on my laptop computer or sketching Ginger with pencil and paper in the park, it’s difficult to say exactly where my mind or my body ends and the technology begins.

As a 3D animator with experience in drawing, painting and sculpture, I intuitively feel that digital technology has an unparalleled capacity to reduce our experience of the world while purporting to expand it.

This intuition accords with the work of contemporary theorists who point out the reductive capacity of digital tools (Hayles, 2012; Lanier, 2011; Pariser, 2011; Rushkoff, 2010; Turkle, 2011). Like me, these theorists have mixed feelings about digital media; they are excited by its vast potential and also wary of its inherent dangers.

Rather than assuming that they serve as a means to an end, Hayles questions how the ubiquity of digital computers might be changing our practices and changing the way that we think. She suggests that our interaction with networked digital technology has rewired our brains for “hyper attention”, making it more difficult for us to engage in sustained, contemplative attention, or what Hayles refers to as “deep attention” (Hayles, 2007). Similarly, Rushkoff says that while computers copy our intellectual processes, they discourage more complex human thought processes (Rushkoff, 2010, p.23). Whatever the nature of the changes, it seems that we can’t just offload certain types of tasks to machines and remain unchanged ourselves because our interactions with machines (on an individual, societal and historical level) involve certain activities and encourage certain ways of seeing and thinking.

According to Rushkoff, digital technology is replete with simplified models of reality from restaurant recommendations and Google Maps to virtual reality communities such as Second Life (Rushkoff, 2010, pp.61-71). Rushkoff explains that “Because digital simulations are numerical models, many choices about them must
have been made in advance. Models are necessarily reductive; they are limited by design. This does not negate their usefulness; it merely qualifies it” (Rushkoff, 2010, p. 71).

The problem is not that abstract and simplified models of the world exist; the problem is that if we don’t question them, they can simplify the way that we see the world. For a 3D animator the problem of simplification manifests in a variety of ways, including our propensity to approach real world things (such as a physical plant or a dog) according to simplified models provided by computer graphics researchers. These simplified models come in the form of 3D animation presets (such as Paint Effects trees illustrated in Figure 1.15) as well as standard 3D workflows (such as the character animation workflow used in Default Whippet and illustrated in Figure 1.6, above).

![Figure 1.15: Images showing how to choose a birch tree from a number of presets and how to tweak parameters. These image from online tutorial “Designing and Animating a Birch Tree in Maya using Paint Effects” (Kumar, 2013) Copyright 2016 Envato Pty Ltd.](image)

In response to the claim that Second Life will eventually be indistinguishable from real life, Rushkoff says, “I doubt there is a computer simulation on the horizon capable of accurately representing all the activity in a single cubic centimetre of soil or the entire sensory experience of clipping one toenail” (Rushkoff, 2010, p. 70). While sketching Ginger in the park I could shift my attention from the trees in the background to the shape of Ginger’s leg, or the hairs on her back, and seeing her includes the feeling of stroking her fur. I therefore agree with Rushkoff’s sentiment because an experience as ordinary as this is simply too complex and elastic to describe in mathematical or algorithmic terms. Rushkoff hopes that “By acknowledging the bias of the digital toward a reduction in complexity, we regain the ability to treat its simulations as models occurring in a vacuum rather than accurate depictions of our world” (Rushkoff, 2010, p.62). We can read about biases inherent in digital technologies but how can we experience these biases for ourselves?

**Creation replaced by selection from a menu**

In his 2001 Book, *Understanding New Media*, Manovich voices the concerns of many when he says that “in computer culture authentic creation has been replaced by selection from a menu” (Manovich, 2001, p.124).

Software manufacturers such as Autodesk and Pixelogic (Autodesk, 2012; Pixologic, 2015) want us to feel like we can make anything with 3D software but Manovich suggests that using digital media (or “new media”) involves choosing from a limited number of available options and that “New media objects are rarely created completely from scratch; usually they are assembled from ready-made parts ... In the process of creating a new media object, the designer selects from libraries of 3D models and texture maps” (Manovich, 2001, p.124). Manovich is referring to the way that 3D animators often work with a ready-made mesh that has been created by someone else. He is also referring to that way that we use simulation and mesh creation presets which allow us to quickly create predesigned virtual objects (such as trees as in Figure 1.15 above) as well as phenomena such as smoke, fire or ocean waves (as illustrated in Figure 1.16).

![Figure 1.16: Screenshots showing how to create ocean waves using Maya's Ocean Shader and how to create smoke using Maya's Fluid Effects.](image)

In *Default Whippet* I chose to build the dog mesh myself, and I didn’t use simulation or mesh creation presets, but the process still involved using pre-designed digital objects such as shaders, lights, etc. This is typical; using 3D software always involves creating pre-designed objects, customising and combining them to suit a project’s specific needs.
Like the drawing recipes that Franck avoids, digital objects can act like a kind of prototype, providing a basis for the way that 3D users describe things in the world.

The progression toward realism

Manovich describes how the development of 3D computer graphics is presented as a progression toward “realism”, and how for the computer graphics community realism means “the ability to simulate any object in such a way that its computer image is indistinguishable from a photograph” (Manovich, 2001, p.168). Comparing 3D graphics with photography and cinema, Manovich makes the point that, in photography a camera records an already existing reality, but in 3D computer graphics we need to construct a virtual reality “from scratch” in order to photograph it with a virtual camera” (Manovich, 2001, p.175).

Constructing a photorealistic virtual world is an enormous task and this goal continues to drive a vast array of computer graphics (CG) research (Yuksel, Kaldor, James, & Marschner, 2012). With the ongoing development of new simulation, lighting and rendering algorithms, 3D software is becoming increasingly sophisticated while also becoming easier to use. For computer graphics pioneers, using a computer meant knowing how to program but today there’s no need to understand the code behind our increasingly “realistic” creations. Does the progress of CG research and the corresponding increase in user choice mean that users become increasingly empowered? Or does the ongoing consolidation of “synthetic realism” (Manovich, 2001, p. 168) mean that the software’s implicit assumptions and simplified models are more difficult to discern, more difficult to question and more difficult to disrupt? Heidegger’s concept of technological Enframing suggests that the latter is true.

3D animation and technological Enframing

More than 50 years before the work of contemporary media theorists such as Rushkoff and Pariser, German philosopher Martin Heidegger wrote The Question Concerning Technology (Heidegger, 1977), urging readers to question assumptions and biases implicit in the technologies that we use. Heidegger warned that technology has the capacity to dictate how we see the world and that if we consider it as a means to an end or as neutral then this capacity is hidden from view. He said:

We are delivered over to [technology] in the worst possible way if we regard it as something neutral; for this conception of it, to which today we particularly like to do homage, makes us utterly blind to the essence of technology. (Heidegger, 1977, p.4)

Heidegger explains that the late modern era is haunted by Enframing which is the “essence of technology” (Heidegger, 1977). The word Enframing (Ge­stell in German) suggests a type of framework according to which we interpret everything that exists. For Heidegger, Enframing is a type of compulsion that "sets upon man to order the real as standing-reserve" (Heidegger, 1977, p.19). In other words, Enframing compels us to consider everything (including ourselves) as quantifiable and exchangeable resources which can be endlessly ordered, optimised and stockpiled ready for use. As a 3D software user, the notion of interpreting the world according to an existing framework – reducing things to what is calculable and exchangeable – is particularly relevant.

Standardized workflows, lighting models and simulation tools can all provide frameworks which have the effect of strictly guiding the 3D user’s attention and actions.

3D animation software is a digital medium and a 3D animation is comprised of numerical representations rather than physical artefacts. This means that a 3D work, as a digital file, can be easily duplicated exactly and infinitely. It also means that the work’s constituent parts (i.e. the data or digital objects within a digital file) can be “switched about ever anew” (Heidegger, 1977, p.16).

Figure 1.17: Screenshots from Default Whippet.
In creating a 3D scene (or virtual world) the user tends to duplicate data wherever possible because it keeps the digital file in order and it reduces the animator’s workload, often mitigating the need to do things twice. For example, when creating Default Whippet, I worked on half the dog mesh and the other half was a mirrored duplicate (see Figure 1.17a). If I wanted to create a scene depicting many dogs, each dog would likely be a copy of the first (see Figure 1.17b). As a virtual form, I can share the dog among projects, among users or even sell it on the internet.

Digital animation before and after software

Images in Figure 1.18 illustrate the state of digital animation in 1972. These were the early days of CG research, when interacting with a computer meant programming. The images on the left show a group of Ivan Sutherland’s students at the University of Utah in the process of digitising a VW car. Operating like a manual 3D scanner, the students painted lines over the car and measured the location of resulting coordinates with a yardstick (“Computer History Museum,” n.d.). By entering these coordinates into the computer, they created the mesh shown in images on the right.

Figure 1.18: Images of Sutherland’s students measuring the coordinates of a VW car, and renders of the resulting car mesh. Image sourced from computerhistory.org.

In the process of digitisation, there is a lot about the car and its context that is left out (e.g. the particular shade of green experienced on a sunny day or the car’s significance to its owner, Sutherland’s wife) and this fact is probably obvious to Sutherland, to his students and also to viewers of their work. To our eyes today, this VW mesh is rudimentary; it’s more like a sketch of a car than a convincing representation. But in the 40 years since this digitisation project, there have been many advances in computer graphics software (and in networked digital computing more generally).

Today the processes used to create a mesh are more intuitive and the outcomes are more convincing. Rather than entering Cartesian coordinates into the computer via the keyboard, contemporary 3D users can “sculpt” a car of up to a billion polygons using a process “that feels incredibly natural” (Pixologic, 2015). With the aid of various lighting and shading tools, a contemporary digital representation of a car can (more) easily approach photorealism. Today we can create photorealistic CG cars. In fact, contemporary advertising often prefers to render virtual cars than to photograph real cars (Southern, 2012). Journalist Alex Southern’s online article from 2012, “Real or Rendered? How 3D Imagery Is Changing the Way You Shop” (Southern, 2012) describes the growing trend toward the use of computer graphics within the advertising industry. Southern states that in 2012, 80% of automotive advertising was created digitally (Southern, 2012).

Figure 1.19: A car advertisement which uses a computer generated car (Southern, 2012).

Car manufacturers were one of the first to use digitally created products instead of photographing the real thing, but this trend is not limited to the automotive industry. A 2012 article in the Wall Street Journal predicted that in 2013 furniture manufacturer IKEA would be creating 25% of its content for catalogues and brochures using 3D software (Hansegard & Jens, 2012).

For companies such as IKEA, computer graphics provides a way to save money and also to achieve visually flawless images (Hansegard & Jens, 2012; Southern, 2012). As a contemporary 3D user, if there is anything about cars that my virtual models leave out then this omission (i.e. the sense that there is an omission) is no longer obvious to me or to viewers of my work.
As described by Heidegger scholar Graham Harman, Enframing “drags … things into a colossal gridwork and reduces them to calculable and manipulable surfaces … Things now have meaning only insofar as they are subjected to this universal grid of presence” (G. Harman, 2009, p.6). Harman’s use of the words "surface" and "presence" allude to the dogmatic and totalising nature of Enframing, i.e. the way that it obscures other understandings and fails to acknowledge the existence of meanings that don’t come from us.

Harman’s description of the totalising nature of Enframing doesn’t seem entirely applicable to activities undertaken by Sutherland and his students or to their “boxy” digital car because of the novelty (i.e. the “non-naturalness”) of this 3D scanning process and because of the obvious inadequacy of the outcome (i.e. the “sketchy” car model). But

*Harman’s description of Enframing does seem applicable to practices and outcomes associated with contemporary 3D software because, as 3D users, we feel like we can make anything and our creations can be very convincing.*

Writing more than 60 years ago, the technologies that Heidegger referred to in his essay include nuclear power plants, mechanised farming and hydroelectric dams. He contrasted these modern technologies with simpler technologies such as an old wooden bridge (Heidegger, 1977, p. 16). Subjected to the demands of the hydroelectric dam, a river becomes a source of power and “appears as something at our command” (Heidegger, 1977, p. 16). Like the hydroelectric dam, a wooden bridge is also an intervention into the landscape but, according to Heidegger, it doesn’t reduce the river in the same way. Unlike the dam, the bridge allows the river that it spans to flow unimpeded and the bridge also highlights some of the river’s intrinsic qualities, bringing them to our attention.

In an essay from 1951, “Building, Dwelling Thinking”, Heidegger describes how a bridge:

> does not just connect banks that are already there. The banks emerge as banks only as the bridge crosses the stream. The bridge designedly causes them to lie across from each other. One side is set off against the other by the bridge. (Heidegger, 1971, p. 150)

The problem with modern technology, as Heidegger sees it, is its capacity to reduce our experience of the world in a way that prior technologies never did. A bridge allows the river’s intrinsic qualities to shine forth, bringing them to our attention, while a hydroelectric dam reduces the river to watts.

3D computer graphics are now a ubiquitous part of the networked digital world (Guha, 2015, p. 3). For animators, 3D software has the capacity to reduce our experience of the world in a way that prior creative media never did. Along with our growing dependence on 3D graphics, reasons for this include the complexity and versatility of 3D software and the fact that its imagery can be convincingly “real” (Konnikova, 2015; Rushkoff, 2010, p. 69). For 3D users, it’s easy to assume that the secrets of the visual world are at our fingertips and, that with enough knowledge and dedication, we can make images as convincing as real world things. By contrast, an activity such as sketching (as outlined above) can open onto a world of inexhaustible complexity - using a pencil to both explore and describe things simply highlights the impossibility of an exhaustive description. Simple technologies such as paint and brushes, or pencil and paper can call attention to the inexhaustible complexity of real things and, in this way, they are like Heidegger’s wooden bridge. 3D software, on the other hand, is more like a hydroelectric dam, reducing real things to data.
2. Context of practice

As suggested by Manovich (2001; see above (introduction.html#creation-replaced-by-selection-from-a-menu)), 3D animation practice often seems to involve choosing from ready-made solutions provided by CG researchers. But there are a number of 3D animators and digital artists who do more than simply utilise existing tools and implement standard workflows. Although they use commercial software, the work of these artists moves beyond conventions and visual styles associated with 3D software and with digital media more generally. This section discusses the work of some of these artists and asks, how do they do it? What tools and methods do they use? In order to contextualise the work of these digital artists, this section discusses computer graphics research, Technical Rationality (Schön, 1991), media transparency and orthodox approaches to 3D software.

Early computer graphics research

In the early 1970s, before the development of commercial 3D software packages, creating a virtual 3D mesh was a less intuitive and a more technical process than it is today. In those days, 3D modelling involved activities such as calculating coordinates and entering them as text onto a computer. As described above (introduction.html#digital-animation-before-and-after-software), this was the approach taken by Sutherland and his students while digitising a VW car. It was also the approach taken by Ed Catmull (also a student at the university of Utah) to create a digital model of his own hand (Figure 2.1; Catmull, 2014, p. 14).

Figure 2.1: Catmull’s plaster cast of his own hand; a wireframe render of his digital model; and a smooth shaded render of the same model. Images sourced from Catmull’s 2014 book (Catmull, 2014, pp. 14, 15) and from maacindia.com.

Catmull chose to digitise a hand because he wanted to render a complex object with a curved surface (Catmull, 2014, pp. 14–15). After making a plaster cast of his left hand, he drew polygons onto the plaster model. He then measured the co-ordinates defining each polygon and entered that data into a 3D animation program that he had written. His first renders were faceted and “boxy” but with the help of “smooth shading” techniques developed by other students he was able to render a smooth, more organic looking form.

Another method for calculating coordinates was to draw an object onto graph paper. Figure 2.2 shows an original sketch by Martin Newell created during the course of his PhD research project undertaken at the University of Utah in 1975 (computerhistory.org, n.d.). Newell sketched a teapot onto graph paper and then converted the sketch into Bezier curves on a computer. Apparently (computerhistory.org, n.d.; wikipedia.org, n.d-b.) the teapot that Newell sketched was his own, and it seems safe to assume that the teapot was significant to Newell in a variety of ways (e.g. practical, financial and perhaps even emotional). However, in digitising the teapot, Newell was not interested in its personal significance or even its physical proximity; he was interested only in properties that can be described mathematically. Specifically, he was interested in the teapot’s form, i.e. its extension in space.

Figure 2.2: Detail of Newell’s Utah Teapot sketch (computerhistory.org, n.d.); wireframe model of Newell’s Utah teapot created by University of Utah researchers (computerhistory.org, n.d.); and a render of Newell’s teapot.

As a 3D mesh, Newell’s virtual teapot exists in a kind of vacuum, devoid of context. In contrast to the original teapot, the digital model (existing as data) can be easily manipulated; it can be scaled, moved, rotated or deformed and it can also be duplicated and shared. Newell made the dataset publically available and the Utah
teapot quickly became a standard for the development of various computer lighting models (wikipedia.org, n.d.-b; simplymaya.com, n.d.).

One such algorithmic model was developed in 1975 by Phong Bui Tuong, also a student at University of Utah (Guha, 2015, p. 425; wikipedia.org; n.d.-a). Commonly referred to as Phong shading, this lighting and shading model is based on Phong’s observation that objects with a matte surface (such as a piece of chalk) have a large area of highlight which falls off gradually, while shiny objects (such as a billiard ball) have a smaller area of highlight which is brighter and falls off more acutely (wikipedia.org, n.d.-a). Drawing on his own experience of everyday things, Phong was able to develop a multi-purpose lighting and shading model which can be used to depict a variety of different surface types.

![Figure 2.3: Visual illustration of the Phong equation. (Wikipedia.org, n.d.-a).](image)

**Emergence of commercial software**

Throughout the 1970s individuals or teams of CG researchers/practitioners purchased computer hardware and had to develop reusable programs themselves (Carlson, 2003). Due to the cost of hardware and the expertise required, these activities were largely restricted to University and industry-supported research institutions. But innovations by researchers like Sutherland, Newell and Phong were soon converted into marketable products and throughout the 1980s 3D computer graphics software became available to a larger number and greater diversity of users (Carlson, 2003; Guha, 2015, p. 8). By the late 1980s it became possible for users to create 3D graphics with little or no knowledge of the code behind their creations. Phong’s is currently the most widely used lighting model in 3D computer graphics (Guha, 2015, p. 425) and Newell’s teapot dataset is available in many 3D software packages with the click of a button (“Why is there a teapot in 3DS Max? - The origin of common 3D models,” n.d.).

![Figure 2.4: Illustration showing the way that a Phong reflection model can be used to shade the surface of a 3D mesh. This image combines screenshots from TurboSquid (“TurboSquid,” n.d.) with a screenshot from Maya’s user interface (UI).](image)

Figure 2.4 illustrates how a Phong reflection model might be used by a contemporary 3D software user to shade ceramic and metallic surfaces. Based on observations of either a real coffee cup or of photographic reference, the user enters values for ambient, diffuse and specular parameters; these are the parameters specified for user input by the algorithm’s designer. The user manipulates values either by dragging a slider or entering numeric values via the keyboard. Test renders are made and values are tweaked until the desired result is achieved.

Contemporary 3D graphics programs offer a wide variety of increasingly complex shading algorithms, and most of them work in a similar fashion: i.e. user-entered values are processed according to predesigned algorithms. The images in Figure 2.4 are sourced from [TurboSquid>As stated on their website, the goal of TurboSquid is to ‘save artists the time of making a great model, and instead let them add their own personality to their creations’ (“TurboSquid,” n.d.), one of many online mesh repositories. Existing as a 3D dataset, this virtual coffee cup can be downloaded for a fee. It is one of many products available to 3D users who want to select from a library of existing models and textures.

21
Technical Rationality

CG research continues today as a sub-field of computer science (Guha, 2015, p. 7) and, compared to the early days of computer graphics, today there is a clearer distinction between software developers and software users. This delineation between research and practice resembles philosopher and academic Donald Schön’s description of “Technical Rationality” as outlined in his 1983 book, *The Reflective Practitioner* (Schön, 1991).

According to Schön, Technical Rationality promotes a model of professional knowledge which considers skillful practice as “instrumental problem solving made rigorous by the application of scientific theory and technique” (Schön, 1991, p. 21). Technical Rationality stems from a positivist epistemology which is characterised by the idea that “scientific knowledge is the only source of positive knowledge of the world” (Schön, 1991, p. 32) and “real knowledge lies in the theories and techniques of basic and applied science” (Schön, 1991, p. 27). In short, Technical Rationality assumes that scientific research can, and should, be the basis for almost all types of intelligent practice. For Technical Rationality, skillful practice means discerning the problem and applying the appropriate available solutions – and this sounds very much like standard approaches to 3D animation.

As a model of professional knowledge, Technical Rationality emphasises the need for standardisation, and it has been used to great effect in many areas of human endeavour including medicine and engineering. However, according to Schön, the model of Technical Rationality can be dangerously reductive because it fails to account for a practitioner’s ability to deal with divergent situations, and in some cases it precludes this ability. Schön points out that practitioners of all persuasions routinely deal with complexity, uncertainty, instability, uniqueness and value-conflict, and a Technical Rationalist model of skillful practice can’t account for this fact. He warns that if practitioners try to operate according to a Technical Rationalist paradigm, then they might be cutting a complex situation to fit prescribed solutions. In this case a practitioner might become “selectively inattentive to data that fall outside their categories” (Schön, 1991, p. 44) or they may “try to force the situation into a mold which lends itself to the use of available techniques” (Schön, 1991, pp. 44–45). Schön’s warnings are relevant to 3D practitioners because we work with solutions designed by CG researchers and software developers. Schön’s thought indicates that if we focus on the correct implementation of standard procedures (i.e. using the right 3D tools in the right way) then we might be cutting a specific practice situation to fit the design of the software. In other words:

*By employing a Technical Rationalist model, we might be missing features of a practice situation which are complex, uncertain, unique or unstable.*

In the Technical Rationalist paradigm, knowledge is formed by abstracting from particular situations to find general rules. These general rules are then used in practice to deal with specific situations. Researchers such as Newell, Phong and Sutherland study particular phenomena and develop general rules in order to design mathematical computer models. With the practices of these researchers in mind, we can see how the logic of Technical Rationality plays out for developers and users of 3D software. Along with Heidegger, Rushkoff and Manovich, Schön’s warnings suggest that our approach to 3D animation could be levelled (or Enframed) if we implicitly accept a Technical Rationalist paradigm.

One artist who overtly critiques this paradigm is award-winning 3D animator, David O’Reilly.

Back to basics and the low-poly aesthetic

![Shitmaker toolkit v1](image)

O’Reilley’s Shitmaker tool.

O’Reilley critiques our propensity to blindly accept CG solutions with his 3D software script “Shitmaker toolkit v1” (O’Reilly, 2016). Available for download from his website, “Shitmaker” is a “free and useless script for Maya” which allows you to “fuck up your favourite 3D models” (O’Reilly, 2016). Like this irreverent tool,
O’Reilley’s 3D animation work consistently breaks with convention and uses commercial software in surprising ways.

Figure 2.6: Still from Please Say Something, by David O'Reilly (2009).

O’Reilley takes a “back to basics” approach to 3D software and by this I mean that he avoids many of the newer and more complex computer graphics solutions which have been designed to achieve photorealism. For example, O’Reilley avoids lighting and rendering techniques such as subsurface scattering and global illumination. His 2009 film Please Say Something (O’Reilley, 2009), which won the Golden Bear for Best Short Film at the Berlinale, avoids software rendering altogether and is instead created entirely from preview renders. In an essay called “Basic Animation Aesthetics” (O’Reilley, 2010), O’Reilley shares some thoughts regarding the making of this film. He says, “The film makes no effort to cover up the fact that it is a computer animation, it holds an array of artefacts which distance it from reality, which tie it closer to the software it came from” (O’Reilley, 2010, p. 2). As well as exposing the the way the work was created to the viewer, another result of a back-to-basics approach is that the “production pipeline [is shortened] to a bare minimum” (O’Reilley, 2010, p. 3). O’Reilley says that because the process is quicker and less rigid, it is easier to make changes and to improvise.

A 3D mesh often starts as a simple geometric form, but the visual style adopted in most 3D animations hides this fact from the viewer. O’Reilley is one of a number of 3D users who have embraced a simple geometric, "low-poly" aesthetic that doesn't smooth the models to create organic-looking forms – instead leaving them faceted and “boxy”. Artist and animator Eran Hilleli is another proponent of this simple geometric style.

Figure 2.7: Still from Between Bears by Eran Hilleli (Hilleli, 2010).

A beautiful example of Hilleli’s work is Between Bears (Hilleli, 2010), an award winning short film created while Hilleli was still a student at the Bezalel Academy of Art and Design. By utilising qualities inherent in 3D animation, Hilleli’s film achieves some of the same qualities as a painting that doesn’t hide its brushstrokes. Specifically, it allows the process of its production to be evident in the final work and it achieves a visual interplay between abstract shapes and figurative (or recognisable) forms. The minimal detail, muted palette, and ambiguous storyline contribute to this film’s poetic and lyrical tone.

There is an increasing interest in the low-poly aesthetic which can also be seen in animations such as Pivot (Megens et al. 2009) and Dropp (Nagatsuka 2011), and in illustrations such as those by Mordi Levi (Levi, 2016), and computer games such as CatDammit! (Fir & Flams, 2014).

Figure 2.8: Detail of still from Pivot (Kevin Megens, Floris Vos, Amo de Grijis, 2009); Detail of still from Dropp by Hajime Nagatsuka (Nagatsuka, 2011); Illustration by Mordi Levi; Detail of screenshot from CatDammit! by Polish game designers Fir & Flams (2014).

Proponents of a low-poly style and a back-to-basics approach utilise qualities inherent in 3D animation software and don’t attempt to hide facts about the way the work was created from a viewer. These strategies can be contrasted with an approach to 3D software which aims to simulate prior media or to simulate photography.
**Media transparency**

**NPR**

Figure 2.9: Detail of still from *Matatoro* (Mauro Carraro Jérémy Pasquet, 2010), a 3D animation with a hand-made aesthetic.

Whether simulating the look of drawing, painting or photography, many 3D animators aim to hide their production process from the viewer, and many of their creations succeed in looking like they were created using a camera, pencils or a brush. As briefly discussed in the previous section, non-photorealistic rendering (NPR) research develops techniques for rendering 3D images that look like traditional art media such as paint, pencil or pastel.

The short animated film *Mata Toro* (Mauro Carraro Jérémy Pasquet, 2010) is a successful example of the NPR genre. Each frame of *Mata Toro* looks like it was created with pencil, watercolour or pastel. It has a hand-made feel which obscures the fact that it was created primarily using 3D animation software. As this piece shows, it is possible for an animation created using digital tools to simulate the inconsistency and texture of pencil or paint on paper. More recent examples of an NPR aesthetic include Disney Studio’s *Paperman* (Kahrs, 2012) which combines a 3D animation with 2D animation techniques (Oatley, 2013).

There is a small but growing consumer market for animations which present a hand-made aesthetic, but the pursuit of photorealism remains the primary concern in Computer Graphics research.

**Photorealism**

Figure 2.10: Still from *The Third and the Seventh* by Alex Roman (2009).

Alex Roman is a 3D animator who creates still images and short films that are indistinguishable from film and photography. The level of detail and precision that Roman has achieved in his highly acclaimed movie *The Third and The Seventh* (Roman, 2009) represents thousands of hours of work and the results are an aesthetically beautiful piece which could easily be mistaken for live action. It is difficult for a viewer of this film to imagine that, like Hilleli and O’Reilly’s models, many of the virtual objects in this movie started as a simple geometric form which became more complex as Roman added more and more detail.

Roman is one of many CG artists who strives for photorealism and this requires that he adds lots of detail, smooths his models, and uses complex lighting and shading algorithms. His aim is to leave no trace of the production process visible in the finished work. Roman’s approach to his work is consistent with what media theorists David Bolter and Jay Grusin refer to as the ongoing quest for media “transparency” (Bolter & Grusin, 2000).

To speak of a tool or a medium as *transparent* is to suggest that it (the brush, the painted surface, the software interface or the computer screen) is not explicitly experienced, i.e. that it doesn’t become an object of study.

*To speak of transparency implies that the medium is somewhat neutral and doesn’t intervene: that it transparently facilitates, communicates or conveys meaning.*
In their 1999 book, *Remediation: Understanding New Media*, Bolter and Grusin suggest that 3D software can be seen as a continuation of the tradition of representation that began with Renaissance painting (Bolter & Grusin, 2000, p. 21). According to them, it is the quest for media transparency that drives the renaissance painter’s use of pictorial perspective. Instead of making the viewer aware of the painting as an object, the use of perspective transports the viewer into the picture’s represented space. Bolter and Grusin state that, “if executed properly the surface of the painting dissolved and presented to the viewer the scene beyond” (Bolter & Grusin, 2000, p. 25).

As well as perfectly executed perspective, dissolution of a painted surface requires that there are no visible brush strokes. Bolter and Grusin describe how renaissance painters “concealed and denied the process ... in favour of a perfected product” (Bolter & Grusin, 2000, p. 25). The same can be said of computer graphics artists such as Roman. 3D animators get mathematically perfect perspective, shading and shadows “for free”, but they still have to work hard to hide the constructed nature of the image – i.e. the fact that it was created on a computer – from a viewer. Default Whippet took hundreds of hours to make – but it would need many more hours of work before a viewer could mistake it for live action photography.

Bolter and Grusin point out that when an artist is skilled and hard-working enough to successfully hide the process of production from a viewer, the irony is that their expertise goes unnoticed (Bolter & Grusin, 2000, p. 25). This potential lack of recognition might be the impetus behind some of the many “making of” videos shared by 3D artists.

**Showing the making process**

![Figure 2.11: a) “Making of a Bruce Lee 3D portrait”, YouTube video by Alexander Beim.; Published Dec 18, 2012 ; 478,976 views; b) “Tywin Lannister Sculpt Timelapse”, YouTube video by by Adam Fisher. ; Published Aug 28, 2013 ; 295,195 views.; c) “Character Modeling Demo Reel”, YouTube video by Patrick Kilcher.; Published Feb 3, 2014. 70,625 views.](image1)

Currently there are countless 3D animation “making of” videos (including modelling, rigging and texturing reels) available on YouTube and Vimeo (see e.g. Figures 2.11 a, b and c). Created and uploaded by large production teams as well as individual users, these videos are not considered finished works in their own right but are designed to reveal to viewers the process of production behind a finished work. These videos are often interesting because they are proof that the images were computer generated. Their popularity might also reflect a fascination with seeing work in which actions and decisions made throughout production are evident. Roman’s “making of” video, along with his finished film, are available on the internet (Roman, 2010; see Figure 2.12). He has also published a book, *The Third and The Seventh: From Bits to the Lens*, which describes his production process in pictures and in words (Roman, 2013).

![Figure 2.12: Two screenshots from *The Third & The Seventh – Compositing Breakdown* by Alex Roman (Roman, 2010); Followed by an image showing Roman’s book, *From Bits to Lens* (sourced from evermotion.org).](image2)

In their 2010 paper, “A Theory of Digital Objects”, Kallinikos, Aaltonen and Marton define digital objects as “digital technologies and devices and digital cultural artefacts such as music, video or image” (Kallinikos, Aaltonen, & Marton, 2010). With this definition in mind we can see that digital artists (including 3D animators) are simultaneously users (consumers) and producers of digital objects.

Whenever we use a computer we are using a variety of digital objects including (but not limited to) a computer operating system, a specific software application and that application’s various tools. Whether making a 3D animation using 3D software or viewing the resulting animation, the logic of media transparency is therefore equally relevant. 3D animators such as Roman aim for a level of finish that renders the digital medium...
transparent to a viewer of his work and, according to Bolter and Grusin, the trajectory of software interface design reveals the same aim (Bolter & Grusin, 2000, p. 31). Bolter and Grusin explain that the replacement of a text-based interface with a graphical user interface (GUI) was the result of an ongoing quest to make digital media transparent; the aim is for an interface that makes the user feel that they are not confronting a medium at all (Bolter & Grusin, 2000, pp. 32–33). Recent incarnations of this quest speak of a “Natural User Interface” (Boulos et al., 2011; Bruder, Steinicke, & Hinrichs, 2009; Lee, 2016; Park, Lee, Lee, Chang, & Kwak, 2016; Petersen & Stricker, 2009; Subramanian, 2015).

Towards a transparent interface

In 3D animation, the trend toward a transparent interface becomes evident when we compare activities (such as the text based entry and manipulation of mesh coordinates) performed by the CG pioneers described above, with contemporary mesh-creation tools.

As mentioned above (introduction.html#the-reductive-capacity-of-digital-technology), ZBrush is a contemporary 3D package which allows users to interactively “sculpt” virtual geometry in a manner that feels “natural” (Pixologic, 2015). The makers of ZBrush urge users to “Leave technical hurdles and steep learning curves behind, as you sculpt and paint with familiar brushes and tools” (Pixologic, 2015). Rather than interacting primarily via a keyboard or even a mouse, 3D sculpting in ZBrush is often done with a stylus which is held like a pencil. The ZBrush interface, even more than most 3D packages, is based on traditional painting and sculpting metaphors.

Figure 2.13: Screenshot showing the ZBrush interface.

The logic behind “artist-friendly” software (see Autodesk, 2016; Pixologic, 2015; Sadeghi, Pritchett, Jensen, & Tamstorf, 2010), such as ZBrush, is to hide algorithmic workings from a user so that the software feels intuitive and easy to use. As explained on the ZBrush website, “ZBrush creates a user experience that feels incredibly natural while simultaneously inspiring the artist within. With the ability to sculpt up to a billion polygons, ZBrush allows you to create limited only by your imagination” (Pixologic, 2015). Similar sentiments are expressed by other 3D software manufacturers. In a recent promotional video by Autodesk called, "Imagine, Design, Create!", a young man enthusiastically describes Autodesk products as the bridge between what you imagine and what you make (Autodesk, 2012). The bridge metaphor suggesting that the software provides a transparent or neutral passage between imagining something and making it happen.

Claims made by ZBrush and Autodesk seem to reflect the ultimate dream of a completely transparent medium, i.e. a medium that doesn’t interfere with an artist’s intention or ideas.

But could such a medium ever really exist? And would we want to use it even if it did? O’Reilly and Hilleli’s work suggests that their answer to these questions would be “no”, because neither of these artists approach their medium as if it were a neutral facilitator, or simply a means to a preconceived end. The distinctive visual styles achieved by O’Reilly and Hilleli have evidently emerged through an appreciation of properties inherent in the software. In other words, it is through an exploration of possibilities suggested by the medium itself that they have each managed to develop a unique voice.

The glitch aesthetic

Instead of seeking media transparency, there are many digital artists who celebrate the ways in which digital media fails to disappear. These artists want to expose the (modernist) myth of media transparency by embracing accidents, unexpected outcomes and computer glitches. One such artist is Curt Cloninger who insists that a computer glitch can be political because “it reminds us that technologies are not neutral tools, but rather are symptoms of our worldview and cultural norms” (Cloninger, 2014, pp. 13–14). Cloninger defines a glitch as either an interruption in a system or as an exposure of that system (Cloninger, 2014, p. 13). Like other
proponents of Glitch (Fell, 2013; Menkman, 2011), Cloninger is interested in how the contours of a system are revealed upon its interruption. He draws upon Heidegger’s philosophy to conceptualise the interplay between interruption and revelation.

While many philosophers emphasise the human capacity for rational reflection, Heidegger points out that in our normal, everyday activities we are not explicitly aware of the things that we are dealing with – it is usually not until they fail us that things become entities, or objects with properties. He famously explains this by describing our typical relationship with a hammer. In use, the hammer usually goes unnoticed and, in a manner of speaking, the hammer is transparent because we “see through” the hammer to the task beyond. For example, when hammering a nail, we are likely to be focused on a task such as hanging a picture; we are unlikely to be focused on the hammer itself (i.e. we don’t explicitly notice its shape, its weight or the colour of its handle). It is only if the hammer breaks or if we miss the nail and hit our finger that the hammer becomes an object that we are explicitly aware of. Heidegger called the way of being of the unnoticed hammer in use, “ready-to-hand” (Heidegger, 1962, pp. 98–99) and the surprising hammer or broken hammer (which becomes an object of attention or deliberate theoretical enquiry) “present-at-hand” (Heidegger, 1962, p. 101).

In his 1927 book Being and Time, Heidegger says “The ready-to-hand is not grasped theoretically at all ... The peculiarity of what is proximally ready-to-hand is that, in its readiness-to-hand, it must, as it were, withdraw in order to be ready-to-hand quite authentically” (Heidegger, 1962, p. 99). Heidegger suggests that our ready-to-hand relations with things are different from, and prior to, a conscious analysis of them.

Following Heidegger’s explanation, Cloninger reasons that “the contours of the system are revealed upon its interruption” (Cloninger, 2014, p. 14). When the system breaks, it’s like the revelation of the broken (present-at-hand) hammer. This revelation (e.g. noticing the hammer) makes us look at everything afresh; all the things with which we have been unconsciously interacting. “Heidegger’s broken hammer causes us to stop and examine the entire world ... a world we had been using implicitly, a world with which we were entangled unawares” (Cloninger, 2014, p. 14). Along with co-author Nick Briz, Cloninger’s 2014 publication Sabotage! Glitch Politix Man[ual/iifesto] is a call to break hammers so that we become aware of the technologies that we usually interact with and take for granted.

In “A Theory of Digital Objects”, Kallinikos et al. outline several features that make digital objects distinctly different from physical objects. One of these defining features is that digital objects are editable. This means that it is always possible to act upon and modify them (at least in principle). Another feature is that digital objects are interactive, which means that they offer users (or viewers) choices of how to act (e.g. navigating a website or choosing items from 3D software menus). Another feature discussed in this paper is openness, which means that “digital objects are possible to access and to modify by means of other digital objects, as when picture-editing software is used to bring changes to digital images” (Kallinikos et al., 2010).

Given that editability, interactivity and openness are qualities inherent in digital objects, it is interesting to note that I feel confined, reduced and constrained by standard approaches to 3D software. Digital artist Rosa Menkman suggests that this sense of confinement is linked to media transparency. In her online publication “Glitch Studies Manifesto”, she writes that “The quest for complete transparency has changed the computer system into a highly complex assemblage that is often hard to penetrate and sometimes even completely closed off” (Menkman, 2010b, p. 3). In her manifesto Menkman says that “The glitch is a wonderful experience of an interruption that shifts an object away from its ordinary form and discourse. For a moment I am shocked, lost and in awe” (Menkman, 2010b, p. 5). Menkman suggests that computer glitches encourage us to see that the typical use of a computer is based on a “genealogy of conventions”, while in reality computer software can be used in many different ways (Menkman, 2010b, p. 3).

**Glitches as a creative opportunities**

Like Cloninger, Menkman’s work contests the modernist ideal of a transparent medium and celebrates the ways that media fail to disappear. Computer software is a medium which modulates data, and Menkman’s work plays with the modulation of data so that, in addition to seeing through the medium to what it represents, the viewer is
also aware of the medium itself. Menkman uses computer glitches as opportunities, creating intriguing images. As described by Kallinikos et al., digital objects are inherently open; they can be accessed and modified using other digital objects. Menkman explores this feature of digital objects in *A vernacular of file formats: A Guide to Databend Compression Design* (Menkman, 2010a). Menkman illustrates this document with a series of portraits, some of which are pictured below (Figure 2.14). These images have been created by altering the header information on RAW image files before opening them in Photoshop, a process which Menkman calls databending.

![Figure 2.14: Details from some of Menkman’s portraits which illustrate her publication “A Vernacular of File Formats” (Menkman, 2010a).](image)

While some people experience technical bugs or imperfections as negative, Menkman emphasizes “the positive consequences of these imperfections by showing the new opportunities they facilitate” (Menkman, 2010b, p. 2). Menkman insists that visual artefacts resulting from technical bugs and mistakes are creative opportunities; she appreciates glitches in the same way that a painter might appreciate splashes or dribs of paint.

**Becoming alive to the accident**

![Figure 2.15: “Study for Head of George Dyer” by Francis Bacon, 1967. Oil on canvas.](image)

Like Menkman, Francis Bacon values the capacity for his medium (paint) to continually surprise (Sylvester, 1975), and he evidently values an artist’s capacity to respond to surprises more than their capacity to entirely control a practice situation (Sylvester, 1975).

Bacon praises the work of 17th century painter Rembrandt van Rijn for his “profound sensibility, which was able to hold onto one irrational mark rather than onto another” (Sylvester, 1975, p. 58). Although using traditional painting tools, Bacon values accidents because they disrupt what he can do with ease and help him to avoid habitual and conventional practices; they help him to avoid illustration and sometimes help him achieve something more “poignant” and “more profound that what [he] really wanted” (Sylvester, 1975, p. 17).

Counter to a Technical Rationalist approach, for Bacon, it’s important *not* to proceed in an entirely formulaic or rational manner. For Bacon, becoming a skillful practitioner means that “One possibly gets better at manipulating the marks that have been made by chance, which are the marks that one made quite outside reason … one becomes more alive to what the accident has proposed for one” (Sylvester, 1975, p. 53).

*The “profound sensibility” cherished by Bacon is evident also in the work of Menkman, O’Reilley and Hilleli, because using commercial software in new ways means being open and responsive to the surprising opportunities that it affords.*

In his book *Heidegger, Art and Post-Modernity*, Heidegger scholar Iain Thomson contrasts the challenging-forth of Enframing (which Thomson calls *technological revealing*) with bringing-forth or *poetic revealing* (Thomson, 2011, p. 21). While challenging-forth imposes a predefined framework, bringing-forth allows things to reveal themselves (somewhat) on their own terms. Thomson suggests that poetic revealing is illustrated in the comportment of a skilled woodworker who studies a piece of wood and, developing a feel for its inherent qualities, allows a sculptural form to emerge through an improvisational process. In contrast to the sensitivity of the woodworker, Thomson imagines an industrial factory indiscriminately reducing pieces of wood to woodchips and then recomposing them to form straight, flat units of timber ready for use in the building industry. As standardised sheets, the wood has become standing-reserve, “optimized, enhanced, and ordered for
maximally flexible use” (Thomson, 2011, p. 19). The factory indiscriminately imposes a predefined framework, order or form onto individual pieces of wood. By contrast, the woodworker works with qualities inherent in each particular piece of wood and ends up with an outcome that is context specific and could not have been known in advance. Importantly, the woodworker is aware that there are multiple appropriate ways to respond the wood’s inherent qualities, but the skilled carpenter is also aware that not any response will do. For Thomson, the industrial factory is an example of “the obtuse domination of technological Enframing”, while the woodworker’s comportment illustrates “the active receptivity of poetic dwelling” (Thomson, 2011, p. 21).

There are similarities between Bacon’s profound sensibility and Heidegger’s descriptions of bringing-forth, because both emphasise our ability to respond to glimmers of meaning or felt significance.

Thomson calls this approach to things a comportment of “active receptivity” (Thomson, 2011, p. 21); he also describes it as a “phenomenological comportment” (Thomson, 2011, p. 20).

Context of practice: Conclusion

Technical Rationality is the dominant model implicit in contemporary 3D software, with CG researchers continuing to provide increasingly accurate (i.e. “realistic”) and efficient representational solutions for use by 3D practitioners. But artists such as O’Reilly, Hilleli and Menkman don’t simply accept these solutions at face value. O’Reilly’s work explicitly disrupts predesigned solutions and, rather than seeking media transparency (either in their tools or in finished works), O’Reilly and Hilleli each explore qualities inherent in 3D software. Not limited to 3D software, Menkman’s work responds to qualities inherent in digital media more generally. Exploring digital media as modulated data, she responds to accidental (and unexpected) outcomes, embracing them as creative opportunities.

Rather than necessarily seeking a straightforward, efficient and transparent process, these artists explore opportunities arising from the process itself. Exploring qualities inherent in their medium and responding to unexpected outcomes are creative strategies which allow these artists to take a fresh approach to their work.

These strategies could be regarded as conscious or unconscious attempts to escape Enframing and to foster a comportment of active receptivity; a profound sensibility which is able to hold onto one irrational mark rather than onto another.

The diversity of approaches to digital media and 3D software discussed in this section indicates that, just as important as the particular technological artefacts/devices that we use is our approach to technology. As we will see in the next section, this emphasis on approach, attitude and comportment accords with Heidegger’s thought.
3. Things themselves

A phenomenon can be defined as “what appears” (H. Spiegelberg, 1971, p. 684) and, since the 1970s, much computer graphics research has aimed to simulate the way that phenomena appear to a camera (Bolter & Grusin, 2000, p. 28; Catmull, 2014, p. 14; Manovich, 2001, p. 180).

Phenomenologist philosophers, on the other hand, are interested in how phenomena appear to a living person, and it turns out that this question is far more complex because, unlike a camera, a living being never perceives reality in a detached or “objective” way.

Edmund Husserl is considered the father of phenomenology and it is from him that we get the slogan “back to the ‘things themselves’” (Husserl, 2001, p. 168). This slogan is urging us to get below distorting categories imposed by the intellect in order to notice and describe things as we actually experience them. Husserl’s early work suggests that, with careful attention, we can describe our experience of something without distorting it.

Heidegger and fellow phenomenologist philosopher Maurice Merleau-Ponty, however, insist that acts of description and acts of understanding are always acts of interpretation. They emphasise that before we have explicit awareness of a thing (before we can start describing it), it has already been revealed to us in a particular way; Heidegger emphasises the role of language in this revelation and Merleau-Ponty emphasises the role of the body.

As proposed by Husserl at its inception, phenomenology was to be a systematic and exact science, but later proponents (including Heidegger and Merleau-Ponty) realised that it could never be an exact science and its value is as a “manner or style of thinking” (Merleau-Ponty, 2002, p. viii). As a style of thinking, phenomenology requires us to ask ourselves “what is it like?” and, without imposing an answer, we need to let things speak for themselves. But what are the things themselves and how can we allow them to “speak”? How can we hear what they have to say? Husserl, Heidegger and Merleau-Ponty each imply different answers to this question.

I am interested in the phenomenologist philosophers for what they can offer my project, which aims develop approaches to 3D animation that avoid reductive frameworks.

In this section I look at their work for suggestions about what strategies or methods might encourage an actively receptive comportment.

Objects and things

In “The Question Concerning Technology” Heidegger explicitly states that the essence of modern technology (as Enframing) is “nothing technological” (Heidegger, 1977, p. 20), suggesting that our approach to technology is more important than the particular technological devices that we use. Heidegger wants us to recognise Enframing as a particular mode of revealing, which means recognising it as a particular way of understanding (or revealing) the world.

A mode of revealing is a particular comportment or a sensibility, and Enframing is a reductive mode of revealing characterised by a challenging-forth which reduces things to standing-reserve (Heidegger, 1977, p. 17). I suggested above (introduction.html#d-animation-and-technological-enframing) that there are similarities between standard 3D animation practices and Heidegger’s notion of Enframing. I used, as an example, Default Whippet. This project involved standard 3D animation practices which feel different from activities such as painting or drawing.
While drawing Ginger in the park I was sitting on the grass; she was doing her thing (i.e. sniffing around) and I was observing her as I sketched. But my comportment toward Ginger dramatically changed once I started using 3D software. Prompted by workflow requirements, I physically manipulated her into a “relaxed pose” (Figure 3.1), after which I dismissed her altogether, preferring to refer to easily accessible digital files.

While working in 3D, my focus was on achieving predictable and repeatable results and I dismissed a number of surprising (yet intriguing) images as mistakes. In accordance with the standard 3D character workflow, I exercised control over the dog and over the software. The resulting dog mesh is standing-reserve: standing by ready for use in other 3D projects and ready to be shared or sold on the internet.

The software itself is also reduced to standing-reserve, because I stick to standard procedures and only appreciate outcomes that are explicitly intentional and repeatable; I am not open and responsive to images that I didn’t (at least partially) conceive of in advance.

Finally, as a proficient user, who correctly implements the character animation workflow in a manner interchangeable with other proficient users, it seems that I have also become standing-reserve; valuable as a human resource for the animation industry.

Is it possible to use 3D software and still be open and responsive to Ginger as a particular and peculiar dog? And how can I be open to surprises that the software has to offer? In short:

*How can I use 3D software without reducing things, such as the software, the dog and even myself, to standing-reserve? In other words, how can I avoid reducing things to objects?*

In everyday speech, we often use the words *thing* and *object* interchangeably, but in Heidegger’s work these words have different connotations. In his book *Heidegger Explained*, Harman succinctly explains that “generally speaking, ‘thing’ is a good term for Heidegger and ‘object’ a bad one” (Harman, 2009, p. 8). For Heidegger the word “object” has negative connotations because it suggests something present-at-hand, consciously studied and reduced to a set of attributes. The word “thing”, on the other hand, has positive connotations and suggests a certain respect; a recognition that a thing’s “thingliness” ensures that it will never be exhausted or fully disclosed. For Heidegger, even the simplest thing could never be fully disclosed or fully intelligible, but reducing things to objects is a way of forgetting this fact. Harman could be talking about 3D computer graphics rather than science when he says that for Heidegger, “science reduces the thing to a present-at-hand caricature by replacing it with a set of tangible properties through which it is modelled” (Harman, 2009, p. 8).

My research is concerned with everyday physical things, as well as digital things including 3D software and 3D images and animations. The aim of my research is to explore ways of working with these things without entirely reducing them to objects.

**Being present to things**

In the opening pages of his essay “The Origin of the Work of Art”, Heidegger argues against our commonly held belief that a thing is the bearer of its characteristics (or traits) (Heidegger, 2002, p. 7). He finds that this definition of a thing is lacking because it doesn’t fit with our *experience of things*. He says that:

> Prior to all reflection, to be attentively present in the domain of things tells us that this concept of the thing [as bearer of traits] is inadequate to its thingliness, its self-sustaining and self-containing nature. From time to time one has the feeling that violence has been done to the thingliness of the thing and that thinking had something to do with it. (Martin Heidegger, 2002, p. 7)
A thing’s “self-sustaining and self-containing nature” (Martin Heidegger, 2002, p. 7), its thingliness, ensures that it always has more in reserve and always has the capacity to surprise us.

Heidegger’s description of violence being done to the thingliness of the thing resonates with my experience of making Default Whippet, where a type of violence was done to the thingliness of Ginger, the thingliness of the software and perhaps to the thingliness of the animated work.

Heidegger suggests that when we consciously study a thing by pulling it apart (physically or conceptually) and measuring it, the thing becomes present-at-hand and it is reduced to standing-reserve. He goes on to say that, when it comes to being attentively present to things, feeling or mood can be more perceptive and “more open to being” than calculation or “reason” (Martin Heidegger, 2002, p. 7). Through the use of a modular workflow, studying things as component parts and working with digital images, feeling and mood played little role in the production of Default Whippet.

My concern with fulfilling workflow requirements and achieving accurate results meant that I wasn’t attentively present to things constituting the practice situation; to Ginger, to the software or to the work.

The thingliness of digital objects

According to contemporary author and a computer scientist Jaaron Lanier, a digital object is always designed for a particular purpose. In his book You are not a Gadget, he explains that “there is no such thing as a digital object that isn’t specialized” (Lanier, 2011, p. 133). He insists that if we assume that a digital object entirely stands in for what it represents then we “believe in bits too much” (Lanier, 2011, p. 133). To illustrate this point, he gives the example of a jpeg image (a digital file or digital object) which represents an oil painting. Given this example, we can see what Lanier means because an oil painting obviously has qualities that a jpeg image doesn’t have. However, it is equally true that a jpeg image has qualities that an oil painting doesn’t have, and the question is whether we miss something about the jpeg image if we think of it merely as a representation of an oil painting. In other words:

*do we miss qualities inherent in digital objects if we encounter them only as they were designed to be encountered?*

The danger of believing in bits too much is that we reduce physical objects and processes to digital representations, i.e. we reduce things of inexhaustible complexity to simplistic caricatures. But not believing in bits enough means conceiving of digital objects as bearers of a quantifiable set of traits, and this could be an equally reductive move. Approaching a digital object (such as a 3D animation package, a tool within that package or the 3D animation produced) solely in accordance with the purposes for which it was designed might be another way of doing violence to the thingliness of a thing. Glitch artists such as Menkman appreciate the capacity for digital media to consistently surprise, and innovative 3D animators such as Hilleli and O’Reilley continue to develop practices that are not listed in the help menu and are not anticipated by software designers. These digital artists respond to qualities inherent in digital objects and these are qualities which have not always been put there by design. In the next section I describe a practical enquiry which explores the thingliness of digital objects and asks whether qualities inherent in digital objects are inexhaustible. Does 3D software have the capacity to continually surprise?

Beginning in poverty

Heidegger suggests that when we consciously study things they become present-at-hand (like the broken hammer) and are reduced to standing-reserve. He emphasises that things themselves are essentially withdrawn, and suggests that we learn more about things by using them rather than consciously studying them (Heidegger, 1962, p. 99). But, for Husserl, the things themselves are things explicitly examined and consciously experienced. Husserl’s writings include detailed descriptions of isolated objects such as a white piece of paper (Moran, 2000, p. 153), a mailbox (Harman, 2002, p. 131) or an inkpot on his desk (Husserl & Welton, 1999, pp. 52–53). For Heidegger, Husserl’s phenomenological studies are contrived experiences because normally we interact with things without explicit awareness of them. Husserl is interested in conscious experience but for Heidegger conscious experience is always secondary and arises from an unnoticed, pre-theoretical background.
By carefully describing the perceptual phenomena associated with his inkpot, Husserl becomes a subject (a conscious observer) knowing an object (the inkwell), and Heidegger insists that we don’t normally experience ourselves as subjects standing back and assessing objects. Just as a normal day might find Newell anticipating the taste of tea rather than focused on the shape of his teapot, on a normal day at his writing desk, Husserl would use his inkpot without explicit awareness of it at all. On these days his relationship with the inkpot is one of practical involvement or “equipmentality” which, as Thomson explains, is a relationship where subject and object have not yet been differentiated and things are not yet entities with objective properties (Thomson, 2011, p. 82).

The previous section described Heidegger’s notion of “equipmentality” in terms of a ready-to-hand hammer in use and a present-at-hand broken hammer. It’s important to recognise that Heidegger’s description of “equipmentality” isn’t just about tools; ready-to-hand and present-at-hand (withdrawn and revealed) are two basic modes of being that belong to all entities (Harman, 2002, 2009).

Despite Heidegger’s critique, Husserl’s phenomenological methods are of interest to 3D users because they represent an attempt to study the appearance of something in a manner free from theoretical explanations and common sense assumptions. Husserl calls his phenomenological method the reduction (Husserl & Welton, 1999, p. 67) or epoche (Husserl & Welton, 1999, p. 374), and it requires us to carefully describe what appears without referring to theories or explanations of that appearance. For example, a phenomenological description of hearing a siren should refrain from talking about sound waves vibrating the eardrum. For 3D users this is an interesting exercise because the tools that we use to describe things often seem to explain them as well. For example, there are algorithmic models which seem to explain how trees grow (Lindenmayer & Prusinkiewicz, 2004), how cloth behaves (Kaldor, James, & Marschner, 2008; Yuksel et al., 2012) and how birds fly in a flock (Reynolds, 1999; Reynolds, 1987). And at the heart of 3D software are a variety of complex algorithms which seem to describe how objects reduce in perspective, how light rays are projected onto the retina and how shading and shadows work.

3D software is comprised of computer algorithms which not only simulate the look of things, but also simulate the way that things behave (see e.g. Figures 3.2 and 3.3).

*Given the intimate relation between description and explanation in 3D software, how can a user study the appearance of something in a manner free from explanation and assumption?*

**Forgetting the objects before you**

According to art historian John Gage, it is the Impressionist painters of the 19th century “who first seemed to paint simply what they saw” (Gage, 1999, p. 209). In his book *Color and Culture: Practice and Meaning from Antiquity to Abstraction*, Gage quotes Claude Monet, a founder of the Impressionist painting movement, who gives the following advice to a fellow artist:

> When you go out to paint, try to forget what objects you have before you, a tree, a house, a field or whatever. Merely think here is a little square of blue, here an oblong of pink, here a streak of yellow, and paint it just as it looks to you, the exact colour and shape, until it gives your own naive impression of the scene before you [Emphasis added]. (Gage, 1999, p. 209)
Monet’s words suggest that there are similarities between the Impressionist approach to landscape painting and Husserl’s *epoche*. Husserl’s method aims to describe phenomena in words while Monet’s method describes perceptual phenomena in paint; both are careful studies of appearance, both bracket questions of existence and both aim to forget the meaning behind the phenomena being studied.

![Monet's haystack paintings](image)

**Figure 3.4:** Three of Monet's many haystack paintings. “Stacks of Wheat (End of Day, Autumn)”, 1890-91, oil on canvas; “Haystacks, midday”, 1890, oil on canvas; “Wheatstacks, Snow Effect, Morning”, 1891, oil on canvas.

## Accidents and essences

According to Graham Harman, the significance of Husserl’s work is his discovery of a gulf between things (or objects, as Harman prefers to call them) and their qualities. Husserl’s phenomenological enquiries reveal that the thing I am conscious of (and consciousness is always consciousness of something) persists below its accidental mode of appearance. An example of this is the way that I experience a unified thing (such as a dog, a car or a teapot) which remains constant despite a retinal image consisting of shapes and colours which change in accordance with changing lighting conditions and angle of view. What I see also changes according to shifts in my attention, mindset or mood.

Harman explains that “The trees and blackbirds we encounter are not detailed presentations of specific bundles of qualities before the mind. Instead, intentional objects have a unified essential core surrounded by a swirling surface of accidents” (Harman, 2010, p. 20). In other words, we normally just see a tree or a blackbird, not its shifting silhouette, its specular highlights or the shape of a shadow across its surface. We normally see through the swirling surface of accidents to unified things, but it’s interesting to note that it’s this swirling surface that intrigues many painters (including Monet).

Husserl wants to uncover the structures of consciousness (Moran, 2000, p. 153). In other words, he wants to find out what makes an experienced thing what it is. For this he needs to know which features of a thing are essential and which are accidental. He needs to know which features can be removed or altered without changing the identity of a thing. Therefore, in addition to bracketing the natural attitude, Husserl’s phenomenological method requires that we train ourselves to see essences by stripping away accidental qualities and surface noise. He calls this method the “eidetic reduction” (Husserl & Welton, 1999, p. 326) and it involves a thought experiment that incrementally strips things of their qualities while testing whether the thing still persists. Husserl assumes that this method will allow us to uncover the essential qualities of things. Natalie Depraz describes the eidetic reduction as “the procedure of discrimination between the intrinsic properties of an object (an armchair necessarily has ‘arms’) and its contingent ones (an armchair is not necessarily made of wood)” (Depraz, n.d.). Husserl’s aim is to see essences and to “fix” them conceptually, then linguistically (Moran, 2000, p. 134). To create 3D tools, programmers also learn to see essences; they need to fix them conceptually and then algorithmically.

![Maya Paint Effects' tree](image)

**Figure 3.5:** Maya Paint Effects’ tree. Image from online tutorial “Designing and Animating a Birch Tree in Maya using Paint Effects” (Kumar, 2013). Copyright 2016 Envato Pty Ltd.

For example, in the design of a mesh creation tool such as a Paint Effects tree for Maya (Figure 3.5 and Figure 1.15, above), decisions have to be made about which features of a tree are essential and which are accidental. In other words, the tool’s programmer or designer needs to decide what are the intrinsic properties of a tree (a tree necessarily has a trunk and leaves) and which are its contingent ones (leaves are not necessarily green – they might be yellow or red). Creating a 3D tool (such as Paint Effects tree) involves the definition of eternal or
fixed qualities that are transferrable across states and situations of use and, if we investigate the code used to build such a tool, we find the notion of essences replicated on a more detailed level because 3D animation software embraces a development paradigm called Object Oriented Programming.

**Object Oriented Programming**

Object Oriented Programming (OOP) was introduced in the 1960s and has been the dominant programming paradigm since the late 1980s (O’Regan, 2008, p. 88).

The traditional view of programming sees it as a list of instructions to be performed by a computer. In contrast to this approach, OOP views a computer program as a collection of objects that act on each other (O’Regan, 2008, p. 87). Traditional languages keep code and data separate but, with OOP, code and data are combined within a code “object”. As O’Regan explains, “Each object is capable of receiving messages, processing data, and sending messages to other objects. That is, each object may be viewed as an independent entity or actor with a distinct role of responsibility.” (O’Regan, 2008, p. 87)

One of the tasks facing an Object Oriented programmer is to write code that defines a class. A class is like a recipe or a blueprint for the creation of an object. In OOP, an object is an instance of a class with its own particular mix of characteristics. For example, a class called “Dog” will define the attributes and behaviours for all possible dog objects. As the programmer who defines the Dog class, I decide what features are essential to a dog object and I decide which of these features can be varied by the user. For example, I may decide that breed, colour, size and number of legs are all attributes that can be varied without destroying a dog’s essential nature. A user of my Dog class can then create a dog object and input values for each of these attributes. One dog object might be a small black Kelpie with 4 legs and another might be a large purple Poodle with 3 legs. Reminiscent of Husserl’s eidetic reduction, the notion of essences pervades the construction of 3D computer software. As illustrated in Figure 3.6, another feature of OOP is inheritance, which means that a class can acquire the properties of another class. For example, a Dog class might be derived from (and inherit attributes from) a parent class called Mammal which might, in turn, inherit attributes from Animal.

![Figure 3.6: Diagram showing how, in OOP, a class can inherit attributes from another class. Image sourced from *The Nature of Code* (Shiffman, 2012).](image)

Husserl died in 1938, long before the introduction of OOP and the development of 3D animation software, but I wonder how he would regard practices related to these technologies and whether he would consider these practices consistent with his phenomenological method. As it resembles the eidetic reduction, Husserl may find the process of creating 3D tools or defining classes to be useful activities but, as predesigned digital objects, what would he think of using these tools? In *Cartesian Meditations*, Husserl writes that the phenomenologist must “begin in absolute poverty, with an absolute lack of knowledge” (Husserl, 1982, p. 2). Given this assertion, we can assume that Husserl would not approve of using a Paint Effects tree because too many decisions about what constitutes our experience of a tree have already been made.

> Given that there is so much knowledge embedded in Object Oriented computer software, it’s probably impossible for a 3D animator to begin in absolute poverty. But what would happen if we tried?

**Wonder in the face of the world**

Husserl asks us to bracket the existence of the world and forget the meaning of things so that we can focus on describing appearance. Merleau-Ponty, like Heidegger, insists that this is an impossible task because we could never entirely remove ourselves from the meaningful world. Merleau-Ponty thinks that the value of Husserl’s phenomenological reduction is that it teaches us “the impossibility of a complete reduction” (Merleau-Ponty, 1996, p. xv). He adds that the “best formulation of the reduction is probably given by Eugen Fink, Husserl’s assistant, when he spoke of ‘wonder’ in the face of the world” (Merleau-Ponty, 1996, p. xv).
We cannot remove ourselves from the world but we can slacken “the intentional threads which attach us to the world and thus bring them to our notice” (Merleau-Ponty, 1996, p. xv). In other words, we can never entirely forget the meaning of things but we can “break with familiarity” in order to notice “the unmotivated upsurge of the world” (Merleau-Ponty, 1996, p. xv). It seems to me that this is what Monet is doing when he sits in front of a haystack and tries to forget what it is that he’s painting.

Like many painters, Monet has trained himself to see accidental surface qualities instead of meaningful things. In a sense Monet turns himself into a camera and paints light hitting the retina but, while painting outdoors in front of a haystack, how detached can Monet really be? It takes work to stay detached and Monet is always pulled back in. Noticing a rustle of hay while he works he surely sees a home for rats, not streaks of colour. He can’t entirely detach himself from the meaningful world, but Monet can slacken intentional threads and get a sideways look at ordinary experience.

Monet has taught himself to see the swirling surface of accidents, but he is also interested in the moment that pieces of paint on a canvas become a haystack. Likewise, Newell (see above (context-of-practice.html#early-computer-graphics-research)) might have been interested in the moment when Bezier curves become a teapot, but for those of us using this off-the-shelf mesh 40 years later, that moment is long forgotten.

Merleau-Ponty feels a particular affinity with the work of Post-Impressionist painter Paul Cézanne, whose project, as he sees it, is the same as his own. In his 1945 essay “Cézanne’s Doubt”, Merleau-Ponty explains that Cézanne’s work explores “primordial” perception (Merleau-Ponty, 1993a, p. 64). This means that Cézanne doesn’t succumb to the perceptual habit of seeing an already constituted or known and categorised world (as classical painters did according to Merleau-Ponty) and neither does he paint pure sensation (as Impressionists such as Monet supposedly did). Instead, Cézanne pursues an “exact study of appearances” (Merleau-Ponty, 1993a, p. 61), but this isn’t as simple as it sounds because appearance doesn’t correspond with light hitting the retina. According to Merleau-Ponty, Cézanne abandons himself “to chaos of sensation” (Merleau-Ponty, 1993a, p. 63). But he doesn’t paint pure sensation: instead he paints the world as it acquires meaning on the level of body and of the intellect.

Objects in the act of appearing

In accordance with Husserl’s suggestion that we should begin in poverty, Cézanne (like Franck) isn’t content to use established painting conventions, such as laws of composition or pictorial perspective, which were popular among many of his contemporaries: instead he struggles to describe perceptual experience which is dynamic, ambiguous and elusive. According to Merleau-Ponty, the apparent futility of Cézanne’s struggle prompted one of his contemporaries to remark that he pursues reality while denying himself the means by which to achieve it (Merleau-Ponty, 1993a, p. 63). By “remaining faithful to the phenomena”, Cézanne eventually develops “lived perspective” which is a looser version of the strict mathematical perspective used by his contemporaries (Merleau-Ponty, 1993a, p. 64). Taken in the context of the whole, Cézanne’s “perspectival distortions are no longer visible in their own right, but rather contribute, as they do in natural vision, to the impression of an emerging order, an object in the act of appearing, organizing itself before our eyes” [emphasis added] (Merleau-Ponty, 1993a, p. 65).
It’s interesting to note that, although Cézanne’s paintings are still images, they feel more dynamic than many 3D animations, which seem somehow static and stable by comparison. Perhaps this is because, in many 3D animations, the objects have already appeared, i.e. things have been objectified.

Following Cézanne, how could a 3D animator describe objects in the act of appearing?

![Image](35x664 to 166x737)

According to Merleau-Ponty, techniques employed by the Impressionists end up dissolving objects. He prefers Cézanne’s technique of modulating colour because it returns the solidity to objects (Merleau-Ponty, 1993a, p. 62). He also likes the way that Cézanne describes things using multiple outlines because “To trace just a single outline sacrifices depth – that is, the dimension in which the thing is presented not as spread out before us but as an inexhaustible reality full of reserves” (Merleau-Ponty, 1993a, p. 65). As 3D users, we work with a virtual third dimension and we are obsessed with depth which, for us, means the distance from a virtual camera. In 3D animation, distance is represented using automatic perspective projection, which can be accentuated by applying “depth of field” effects (Beane, 2012, p. 125; Derakhshani, 2013, p. 569; see Figure 3.9). But the illusion of spatial distance is not the type of depth that Merleau-Ponty is referring to: what he is referring to might be better described as allusion; it’s more like a hint or suggestion of something that is not entirely present.

The world of perception

Husserl’s methods enquire into things that we are explicitly aware of, but Merleau-Ponty (like Heidegger) emphasises that before we can have explicit awareness of a thing, it has already been revealed to us in a particular way.

For Merleau-Ponty the level of our primary revelation of the world is perception, and much of his work examines the phenomenology of perception. Rather than a scientific or biological explanation of our perceptual system (which might talk about light rays bouncing off objects and forming an image on the retina), Merleau-Ponty is interested in our perceptual experience as embodied beings.

Through an exhaustive investigation, Merleau-Ponty concludes that perception is not the construction (or projection) of a world by our intellect, nor is it the passive uptake of data about a pre-existing external world. He attributes each of these extreme accounts of perception to “intellectualist” and “empiricists” (Merleau-Ponty, 2002, p. 30)(T. Carman, 2005, p. 50). For Merleau-Ponty, neither of these positions can account for the richness and complexity of lived perceptual experience; his work takes a kind of middle road between these two extremes.

Merleau-Ponty insists that we exist both as conscious subjects in a world of meaning and as finite, embodied beings. For Merleau-Ponty, perception is always “value-added”; it is expressive, it stylises, filters and attributes meaning.

According to Merleau-Ponty (1963, p. 167), one of the many phenomena which show how perception is always already meaningful includes the way that we have trouble treating a face like a thing.

Through an examination of perceptual experience, Merleau-Ponty notices that what we see is largely based on the physical activities that we are able to perform. For example, he describes how distance and movement on a vertical plane (i.e. up and down) is different from movement on a horizontal plane (i.e. side to side). This is because, as embodied human beings, we have a front and a back and a top and a bottom. For perception, space is not neutral; it’s not like an empty container where each point or location is equivalent to any other. Instead, space “consists of different regions and has certain privileged directions; these are closely related to our distinctive bodily features and our situation as beings thrown into the world” (Merleau-Ponty, 2004, p. 67). 3D
animation software is primarily based on the Cartesian coordinate system and it represents virtual space in mathematical terms. As a user of this software, there are a number of practices (e.g. building objects out of context and assembling a scene from component parts) that make it easy to assume that space is neutral.

*Working within mathematically described space is one of several ways that standard 3D workflows allow the role of the body in perception to be hidden from view.*

As an embodied being, my perception is based on my biology (e.g. eyes on each side of my head would reveal a different world) and it is also based on my culture and personal experience. Like many other humans (with bodies like mine), it’s difficult for me to treat a face like a thing and, similarly it’s difficult for me to see a dog in the same way that I see an inanimate object. When it comes to my own dog, Ginger, it’s difficult for me to see her like any other dog (or even like any other whippet). In *Default Whippet* I approached Ginger as though she were any other dog and (to a certain extent) like she was any other object. In this project my relationship with Ginger (including our physical proximity and our emotional bond) doesn’t come into play. For Merleau-Ponty this omission means overlooking the phenomenology of perception; and Heidegger, as we’ve seen earlier, suggests that it is a way of doing violence to a thing.

As 3D animators, how can we allow our (physical and emotional) relationships with things to influence our practice?

**The arbitration of conflict**

The dominant rhetoric surrounding the development and use of photorealistic rendering algorithms and NPR techniques reflects an assumption that there are some ways of representing the world that are “realistic” (Kozbelt, 2006) or objectively accurate, while there are others which are stylised, “expressive” or artistic (Power, 2009) – but Merleau-Ponty rejects this distinction.

For example, we might assume that images with mathematically perfect perspective provide an objectively accurate representation of the way the world is, and that deviations from this precision are stylistic interpretations. But, according to Merleau-Ponty, perspective is not a “technique of imitation”, it is the “invention of a world” (Merleau-Ponty, 1993b, p. 86).

In his article on aesthetics, “Indirect Language and the Voices of Silence”, Merleau-Ponty writes:

> It is clear that the classical perspective is only one of the ways that humanity has invented for projecting the perceived world before itself, and not the copy of that world. The classical perspective is an optional interpretation of spontaneous vision (Merleau-Ponty, 1993b, p. 86).

For Bolter and Grusin (2000, p. 25), the relevance of pictorial perspective is that it helps to dissolve the surface of a picture and present viewers with the scene beyond. For Merleau-Ponty, the relevance of pictorial perspective is that it is a means of arbitrating conflict (Merleau-Ponty, 1993b, p. 86). Comparing “free perception” or “spontaneous vision” (Merleau-Ponty, 1993b, p. 86) to classical perspective he says:

> Before [in spontaneous vision], I had the experience of a world of teeming, exclusive things which could be taken in only by means of a temporal cycle in which each gain was at the same time a loss. Now the inexhaustible being chrysalises into an ordered perspective within which backgrounds resign themselves to being only backgrounds (inaccessible and vague as proper), and objects in the foreground abandon something of their aggressiveness, order their inner lines according to the common law of the spectacle, and already prepare themselves to be backgrounds as soon as is necessary (Merleau-Ponty, 1993b, p. 87).

Throughout this essay, Merleau-Ponty provides evocative descriptions of spontaneous vision, many of which resonate with my experience of activities such as sketching Ginger in the park. He describes how, in normal vision (or free perception), “things [compete] for my glance; and anchored in one of them, I [feel] in it the solicitation of the others which made them coexist with the first – the demands of a horizon and its claim to exist” (Merleau-Ponty, 1993b, p. 87). In other words, visually anchored in a thing, I am continually compelled shift my attention and, visually anchored in the next thing, the first is still somehow present.
But when drawing in perspective, “what I transfer to paper is not this coexistence of perceived things as rivals in my field of vision. I find [instead] the means of arbitrating their conflict” (Merleau-Ponty, 1993b, p. 86). Adhering to the conventions of perspective, “The whole scene is in the mode of the completed or of eternity... Things no longer call upon me to answer and I am no longer compromised by them” [emphasis added] (Merleau-Ponty, 1993b, p. 87). For Merleau-Ponty, perspective is a way of imposing order on things and fixing them in place and, in this sense, it is like Heidegger’s descriptions of Enframing and reducing things to objects.

In 3D animation perspective projection is automatic and it is just one of many ways in which the software arbitrates conflict. Merleau-Ponty notices that in spontaneous vision, things call upon us to answer, which means that things move us physically and emotionally. Merleau-Ponty also insists that in free perception things are ambiguous and invite a variety of interpretations; in free perception things are vaguely unsettling. But for a 3D user, objects and scenes are usually unambiguous and don’t call upon us to answer.

*Given the numerous ways that 3D software quietly arbitrates conflict, how can a 3D user explore spontaneous vision?*

Franck describes how he is physically moved by ordinary things, always seeing them as extraordinary, while Cézanne’s work is motivated by a “feeling of strangeness” (Merleau-Ponty, 1993a, p. 68). Both of these artists work from life and are continually intrigued by things that are usually taken for granted. Working with predesigned digital objects (including automatic perspective projection, algorithmic lighting models and mesh creation tools such as a Paint Effects tree), the 3D user seldom interrogates their own perceptual experience and easily takes the perceived world for granted, often blindly accepting representational conventions.

Merleau-Ponty urges us to bring the world of perception “back to life”, and warns that it is often “hidden from us beneath all the sediment of knowledge and social living” (Merleau-Ponty, 2004, p. 93). Similarly, the world of perception is hidden from 3D users beneath all the sediment of knowledge packaged as 3D tools and standard workflows.

In my attempt to shake up this sediment of knowledge and expand the language of 3D software, Merleau-Ponty’s thought, in conjunction with Monet, Cézanne and Franck, suggests the value of returning to the world of perception.

*How can a 3D animator disrupt habitual ways of seeing and experience their own role in the constitution of things?*

This is one of the questions explored in the next section of this document.

**Things Themselves: Conclusion**

At the conclusion of this section I am left with a number of questions, including the following. As a 3D animator, how can I move from challenging-forth to bringing-forth? How can I foster a comportment of active receptivity? In other words, how can I work with things without turning them into objects?

Following Husserl’s suggestions, what would it mean for a 3D user to begin in poverty? And, rather than simply using predesigned tools (where essences have already been defined), can a 3D animator gain something from seeing (and “fixing”) essences themselves?

Merleau-Ponty urges us to return to the world of perception but what sort of 3D practices could explore the role of the body in perceptual experience? As 3D animators, how can we incorporate our physical and emotional relation to things?

The answers to some of these questions are suggested by the work of other artists, including Bacon, Franck, Monet and Cézanne, as well as the work of digital artists such as O’Reilly, Hilleli and Menkman.

In the next section, I describe a practice-based enquiry which is informed by the considerations of phenomenology, and the various approaches adopted by the above artists.
4. Experimental animations

Previous sections of this document have described how orthodox approaches to 3D software often focus on control, and involve a comportment which resembles Heidegger’s description of Enframing and challenging-forth. This section describes a practice-led enquiry aimed at developing approaches to 3D software which encourage a comportment of active receptivity, or bringing-forth.

Motivated by the search for particular qualities of practice (i.e. by a particular feeling or experience of making) this research has involved the creation of a large number of short 3D animations. As the research progressed, I began to recognise a number of creative strategies that were used repeatedly. These strategies are interrelated and overlapping but I refer to them by the following names: Playing with Software, Playing with History, Animation as Relation, Colour as Light, Geometry as Shadow, Working from Sketches and Working from Life. Using these strategies as an organisational device, this section describes a number of experimental animations, creative strategies and custom tools, with a focus on those that reflect significant moments in my enquiry.

Research methods

Reflection on prior practice

This research draws on my previous experience as a 3D animator, and as a visual artist using traditional media. Since graduating with a fine arts degree in 1996, I have worked with a variety of creative media including pencil, charcoal, paint and clay. In 2001 I began using 3D animation software.

Over the past 15 years I have been employed as a 3D animator in the advertising industry, have delivered 3D animation courses at several tertiary institutions and have worked on personally motivated 3D animation projects. In my role as a 3D animator I have engaged with the global 3D community through online forums, courses and tutorials and I have created work that approaches photorealism (e.g. Figure 4.1), as well as work that strives for a sketchy or painterly aesthetic (e.g. Figure 4.2).

While I have shown that it is possible for a 3D animation to look like a moving drawing or painting, I have also noticed that digital and analogue media tend to encourage different artistic concerns. Through reflection on my prior practice, as well as my work with Default Whippet, I have identified a number of common 3D animation practices which derive from and encourage a comportment of control:

- Using the software to get things done in an ordered and efficient manner. Focusing on achieving predictable and repeatable results through the correct use of tools and by adhering to standard procedures.
Turning away from everyday physical things in the local environment and using screen-based digital images to inspire and guide the work.

Describing everyday physical things in accordance with standard 3D tools, presets and workflows (examples include assessing a coffee cup in terms of specularity and reflectivity, or manipulating a dog to emulate a default pose).

Working with stable properties, completed models and unambiguous images wherever possible.

Creating images and animations which simulate the look of prior media (such as pencil or paint) or seeking to emulate photography. Both of these approaches obscure the fact of the digital medium in a final animation.

Obscuring the process of production (including the user’s intuitive decisions) in a final animation. For example, there are no working iterations of the dog model visible in the finished Default Whippet animation.

Against the backdrop of common 3D practices such as those listed above, my research uses a practice-led enquiry to explore a variety of alternative practices in order to develop approaches to 3D software which encourage a comportment of active receptivity. In the discussion that follows I describe a number of these alternative practices with a focus on those which were found to be most successful.

Taking notes during production

My production work has involved the creation of a large number of short 3D animations. While making these works I kept a project diary consisting of written and visual records which were useful during production and also when I later reflected on what (if anything) had been interesting, relevant or surprising to me. These records include images of works in progress, technical notes and snippets of code, as well as observations about my general mood and level of engagement.

At regular intervals throughout the production each animation, I noted whether the process was engaging, boring, challenging or absorbing. I also noted whether I encountered surprises and, if so, whether they altered the direction of a project. I wrote about my plans for a specific project and, as I worked, I jotted down ideas and opportunities as they arose. Reading back over these notes, it’s clear that my plans continually changed. Selected pages from my project diary, which exists as a series of online documents, have been downloaded in PDF format and appear in Appendix A of this document.

Technical notes

Description of hardware and software

All of the animations created for this research use one of three commercial 3D software packages: Autodesk’s Maya, Side Effects’ Houdini, and Pixelogic’s Zbrush. Maya is the primary software used for this research and the package with which I feel most comfortable (having used it for almost 15 years). Houdini is used for a few of the animations and, as its tools and workflows are different from those in Maya, it provides a point of comparison. ZBrush represents a third example of contemporary 3D software; its architecture is different from both Maya and Houdini. As discussed above, ZBrush is commonly considered an “artist friendly” package because its interface design encourages users to think of polygon modelling as “digital sculpting” (Pixologic, 2015). In the past I have used ZBrush to create detailed polygon models, in this research ZBrush was used to create photographic textures for Default Whippet.
Scripting

Many of the experimental animations described in this section use existing tools in an unorthodox way, or they use custom-made tools created as part of this research. Some of these custom tools take the form of simple scripts (executed from Maya’s Script Editor or via a shelf button) which are used on one or two projects, while others take the form of graphical user interfaces (GUIs or UIs) and are used for a whole series of works.

Some of the scripts simply speed up repetitive tasks, while others enable ways of interacting with the software which would not otherwise be feasible. To undertake this research, it was necessary for me to acquire rudimentary programming skills and I chose to learn the object oriented programming (OOP) language, Python. A relative simple language, Python can be used to customise several 3D software packages including Maya and Houdini. Maya’s native scripting language is MEL and Houdini’s is VEX; some scripting in these languages was also undertaken as part of this research.

Custom tools discussed in this section are available in Appendix B of this document as Python code (which has been converted to PDF format for readability). Rather than contributing to a system of standard software solutions, these tools should be regarded as works in progress.

Description of practice-led enquiry

For Default Whippet (Figure 1.2 above (introduction.html#ginger-and-default-whippet) ), I was after as much control over the dog's form and movement as possible; but in my experimental animations I wanted to find ways of varying my level of control over the work. This is analogous to the idea, common among painters, that the paint should be given some freedom to do what it wants; to splash, to dribble and make marks that are not entirely intentional. A painter can vary their level of control of a work by altering the consistency of the paint, throwing paint at the canvas or by altering their grip on the brush. And a painter can respond to accidental marks by leaving them, adjusting them or painting over them. In search of equivalent approaches to 3D software, my animations explore the following questions:

- How can a user vary their control over the software in order to elicit glitches or surprise outcomes? How can they respond to these surprise outcomes and how can these responses be incorporated into a final animation?

Playing with Software

Many of the animation projects described below aimed to elicit suggestions from the software to which I could respond. In my search for suggested form and suggested movement, there are several strategies that I used repeatedly. The first of these strategies I refer to as Playing with Software.

Playing with Software is different from using software because it is less concerned with what a tool has been designed to do and is more concerned with what the tool actually does.

Cow Reduce
An example of *Playing with Software* is *Cow Reduce*, an early animation which I created by animating Maya’s *Reduce* tool. This tool is used primarily for the purpose of creating a low-resolution version of a high-resolution model. This is especially useful in situations where file sizes need to be kept to a minimum, such as in the production of computer games. The Reduce tool is designed to give the user immediate visual feedback while they work but is not normally a tool that is animated.

*Wobbling Primitives*

Some of my software experiments involved playing with simple polygon primitives and one day I noted in my project diary, “I am just changing the creation parameters of a pyramid primitive ... as an animation it gets a great little wobble as divisions are added” (Appendix A, Project Diary: General, 17/4/13). Intrigued by the “great little wobble” that you can see in the movie at Figure 4.5b, I recorded the steps I had taken in my project diary:

- make a polygon pyramid primitive (number of sides = 5), scale it on Y axis, PolySplitRing on each side (split from bottom to almost to the top), move new edges outward (the further you move them the more extreme the wobble), animate the PolyPyramid “subdivisions height” attribute (in this case it goes from 1 to 100). (Appendix B, Project Diary: General, 17/04/13)

These steps are illustrated in the viewport snapshots (Figure 4.5a).

![Figure 4.5: a) Three viewport snapshots showing steps taken to create Wobbling Primitives. b) Wobbling Primitives 3D animation.](image)

The animated forms and the great little wobble described above result from characteristics of the software’s architecture; specifically they are the result of *construction history*. In the context of 3D animation, "history" refers to data stored during the creation of a 3D mesh. When you create a mesh the software stores certain information about how it was made and (in Maya, as well as some other 3D programs) this stored information is known as construction history or simply as *history*.

An Autodesk manual from 2009 defines history as “A buffer in which the construction steps of an object are stored for later editing” (Derakhshani, 2011, p. 584). As indicated by this definition, in orthodox 3D practice altering the construction history on an object can be a useful way of editing. However, as the list of stored construction steps grows, small changes to a model’s history can yield surprising results. This is because the list of construction steps is entangled, i.e. the result of one step depends on the results of previous steps. In order to avoid unpredictable deformations, we normally delete construction history while we work but many of the experiments discussed here explore surprising deformation and movement that results from this historical dependence. Used to elicit suggested form and suggested movement, I refer to this strategy as *Playing with History*.

*Playing with history*

*Hand*

Like the wobbling pyramid (Figure 4.5), the hand models (Figure 4.6) were also created by *Playing with History*. In this case I deliberately made the model on the right and then changed its history to cycle through a variety of forms such as those you can see to the left of this original model. My favourite is the form on the far left because, although evocative of a hand, it is not something that I could have designed in advance or even imagined. In other words, this form says something about hands but it’s not how I would think to describe a hand.
Another example of Playing with History is Blinding Tree, which started as a simple polygon cube. After extruding faces to create branches, I changed properties of the original cube which resulted in a series of unpredictable mesh deformations. Like the hand and wobbling primitive examples above, I enjoyed cycling through the tree’s glitched forms in the Maya viewport. Like Wobbling Primitives, for Blinding Tree I keyframed attributes on the cube so that they changed throughout the timeline. In this way the cycling is animated so that what I saw in the viewport is similar to the rendered animation. The movie in Figure 4.7 shows how I also animated the tree's shader so that light passes over the tree object. This reminds me of trees and passing cars that I see each evening on my ride home.

Traditional animation principles: Conformity and disruption

In 3D character animation we normally strive for as much control as possible over a character’s movement and its deformation. Figure 4.8a shows how in Default Whippet I carefully positioned the dog’s internal skeleton to gain control over her form, and Figures 4.8b and c show how I meticulously laid keyframes to control her movement.

Default Whippet uses a standard approach to 3D character animation; an approach which is nicely summarised in a 1987 paper by John Lasseter. A highly acclaimed animator and screenwriter, Lasseter is currently chief creative officer at Pixar and Disney studios.

In his 1987 paper “Principles of Traditional Animation Applied to Computer Animation” (Lasseter, 1987), Lasseter describes how principles used to create traditional 2D hand-drawn animation can be applied to 3D computer animation. He stresses that, for animators, knowledge of these principles is essential to the production of quality work. Lasseter also suggests that these principles should inform 3D animation software design and says “an understanding [of traditional animation principles] is important to the designers of the systems used by these [3D] animators” (Lasseter, 1987, p. 35). The animation principles that Lasseter describes were designed primarily for character animation and their main goal is to give a character “personality” (Lasseter, 1987). Traditional animation principles include “Squash and Stretch”, “Anticipation” and “Exaggeration” (to name a few) and they are familiar to many 3D animators, including myself. Having absorbed these principles by
watching and creating animations over many years, they form part of the vast background which informs my research. However, rather than adhering to these principles, my experimental animations are trying to escape from or to disrupt them.

*Rather than finding principles or methods that help animators achieve an already familiar style of movement, my research seeks strategies which encourage close attention to styles of movement that emerge.*

For Lasseter, Pixar and many traditional animators, “it is important to make the personality [of a character] distinct, and familiar to the audience” (Lasseter, 1987). These animators want to communicate unambiguous stories and images. By contrast I am interested in imagery which is evocative but that might be difficult to define or to describe.

According to Lasseter “all actions and movements of a character are the result of its thought processes” (Lasseter, 1987). This is an interesting statement because it suggests an emphasis on deliberate, consciously motivated activity. The implications of Lasseter’s statement are that Pixar characters are largely motivated by conscious ideas or thoughts; but what about movements that aren’t consciously motivated; what about feelings or sensations that aren’t attached to thought; and what about thoughts and ideas that arrive through activity? Lasseter emphasises the realm of conscious and deliberate actions and this emphasis applies to Pixar characters as well as to Pixar animators. He says, “In order to get a thought process into an animation, it is critical to have the personality of the character in mind at the outset” (Lasseter, 1987). In the context of my research, I am wary of Lasseter’s advice because I sense that pre-designing the personality of a character or the style of an animation might result in missed opportunities. Working toward a predefined goal might distract me from paying careful attention to emergent characters.

My research aim is to develop strategies that focus on emergent properties of an image or animation. I think of these as “bottom up” strategies compared to traditional approaches, such as Lasseter’s, which are more “top-down”.

**Spline Whippet**

*Figure 4.9: Viewport snapshot from Spline Whippet.*

With *Default Whippet*, I wanted to control the dog’s form and movement as much as possible; but in *Spline Whippet* (Figure 4.9) I was seeking subtle surprises. As the name suggests, the construction of *Spline Whippet* involved the use of curves (or splines) to guide the extrusion of faces on a mesh. This is not an unusual way to create a character model but normally, after completing the model, we would delete its construction history and bind the mesh directly to joints. In *Spline Whippet* the construction history was not deleted and the guide curves, rather than the mesh, were bound to the joints. As a result of this workflow there are subtle movements and deformations that I didn’t explicitly design. In *Spline Whippet* I didn’t entirely relinquish my control over the dog’s form and movement but I reduced it considerably. If my approach to *Hand* and *Blinding Tree* was like throwing paint at the canvas, *Spline Whippet* was more like painting with a brush attached to a long and flexible stick. As well as suggested form, I found that *Playing with History* can elicit suggested movement.

**Library Man**

*Figure 4.10: The pencil sketch which I referred to when modelling Library Man (Figure 4.11, below).*

Library Man is another example of how a character’s personality or disposition can emerge through a process of discovery. After using a pencil sketch (Figure 4.10) to create a figure model, in this project I experimented with different code fragments to drive the rotation of joints. In the course of this project I asked myself simple
questions such as “how do I make the amount that the head turns non-constant?” (Appendix A, Project Diary: Library Man, 24/01/13). And, as indicated by the following excerpt from my project diary, I found a variety answers:

```c
found this to change direction;
if ( frame % 60 == 0) $xDir = rand( -45,45 );
pCube2.rotateY = $xDir;”
Found a way of creating random loop!!!!;
if (frame == 1) seed(1);
translateY = rand(5). (Appendix A, Project Diary: Library Man, 24/01/13)
```

The character of *Library Man* emerged through a process of continual discovery. In the movie above (Figure 4.11 he looks agitated or nervous but, throughout production, *Library Man* exhibited a variety of subtly different moods or behaviours.

### Sine Whippet

Figure 4.12 shows early animation experiments which attempt to bring characters to life using maths functions such as *noise*, *linstep* and *sine*. In these examples a character or style of movement emerges through a mathematical process which is still playful but is more obviously constrained by self-imposed restrictions.

![Figure 4.12: a) Detail from project diary showing working sketches for Spline Whippet and similar experiments; b) Viewport snapshot showing my attempt to animate Default Whippet using maths functions; c) Sine Whippet animation.](image)

Rather than using a function to drive the motion on each part of a character separately, in these experiments I made links between parts (i.e. objects) so that the motion on one object defines the motion of others. For example, in *Sine Whippet* (Figure 4.12c) it is the movement on the large cube in the dog’s chest that drives the motion of other cubes in her body, neck and head. Using a time offset function, movement of the chest is distributed throughout the dog’s body.

Creating movement by defining the relationship between one object and another was a strategy used in *Sine Whippet* and also in *Library Man*. As my appreciation of this animation strategy increased I began to call it *Animation as Relation* and I would go on to explore it in a variety of ways.

### Animation as relation

Experimental animations discussed so far in this document use standard interface tools in an unorthodox manner, but as my research progressed it became apparent that the ability to make my own custom tools would allow me to explore a wider variety of approaches.

After 15 years of using 3D software with no programming knowledge, for this research I needed to learn to program. According to Douglas Rushkoff (2010), we shouldn’t think of programming as the boring stuff; it doesn’t have to be a rote activity or involve the implementation of a preconceived design. Rushkoff is one of many programmers who insists that programming is creative and it can be improvisational (Maeda, 2004; Rushkoff, 2010). In accordance with this sentiment, the tools described below were developed in an improvisational fashion. Sometimes I was motivated by a perceived need (“I need to find a way of doing X”),
other times I was motivated by a question (“What would happen if I did Y?”), and often I was intrigued and inspired by a surprise outcome (“Wow, look at that! I accidentally did Z”). In the same way that the experimental animations emerged through an improvisational process, so too did the custom tools.

**Auto Expression**

One of the first custom tools that I created was *Auto Expression*. This is a custom user interface (UI) that allows the animator to quickly link different object properties using mathematical *expressions*. There are countless types of properties (or attributes) that can be linked using this custom tool including those that define an object’s colour, size, shape, position or orientation. The screen capture video at Figure 4.13 shows the tool being used to define relationships between elongated cubes.

![Figure 4.13: Demonstration video showing my Auto Expression UI in use.](image)

In the video above, the cube to the right of screen has been defined as the *driver* object. Rotation values of this cube have been animated using keyframes. In the first half of the video the cube on the far left is the single *driven* object; its location on the Y axis (i.e. its vertical position) has been linked to the driver cube’s rotation value. After defining this relationship, you can see how the driven cube (left of screen) moves up and down when the driver cube (right of screen) rotates. The second half of the video shows what happens when a single object (and this example uses the same driver cube on the right of screen) drives multiple objects. Toward the end of this example you can see the wave effect that results from changing the attribute which I call “child delay”. This attribute offsets the movement of driven objects in time. Using *Auto Expression* involves defining relationships and then playing through the timeline to view the resulting movement on driven objects. This movement feels like a suggestion from the software to the user responds by tweaking relationships between (driver and driven) objects.

**Woman with a Bag**

*Woman with a Bag* (Figure 4.14c) uses the *Auto Expression* UI to link cubes so that the height of one fence post drives the height of others.

![Figure 4.14: a) The pencil sketch which inspired Woman with a Bag; b) A working playblast; c) Woman with a Bag animation.](image)

*Woman with a Bag* was also the testing site for a second custom tool, this one I call *Auto Keyframe* because it creates many key frames with the click of a button.

**Auto Keyframe**

*Auto Keyframe* is a tool for creating animation which cycles, loops or repeats and, just like *Auto Expression*, it works by defining relationships.

![Figure 4.15: Demonstration video showing my Auto Keyframe UI in use.](image)
To use *Auto Keyframe* the animator first selects an object which they want to keyframe (again, this is the driven object). They then select another object which has already been keyframed (this object is the driver). After specifying how many keyframes define a loop, the user fills out a spreadsheet in order to define the relationship between existing keys (on the driver) and yet to be created keys (on the driven). After completing a spreadsheet for each driven object, the user clicks a button to create all the keyframes which have been defined. The animation is then viewed and, in response to what they see, the user can then move the new keyframes to tweak the movement of objects. Alternatively, they can tweak keys on the driver object or they can adjust the values on the spreadsheet and then press the red button to automatically recreate all keys. This process of watching and tweaking is likely to be repeated many times.

The *Auto Keyframe* UI is flexible, i.e. the number of tabs and columns expands and contracts according to the number of keyframes in a given cycle. Data defining the relationship between a driven object and its driver is stored as text files on the computer. One option for a user is to copy, create or manually edit these files in a text program then, from within Maya, text files can be loaded to create the interface with user-defined default values.

Of all my custom UIs, this one took the longest to make and it is perhaps the most ambitious. It is also the custom tool that I have used the least. *Auto Keyframe* was challenging and fun to build but (at least at first) it was less fun to use. When used on *Woman with a Bag* (to create the woman’s walkcycle), *Auto Keyframe* felt cumbersome (i.e. unnecessarily complex). However, there was to be a later experiment using *Auto Keyframe* which was more successful (see Figure 4.16 and 4.57).

*Peds Prance*

![Image](image.png)

Figure 4.16: Playblast (i.e. series of viewport snapshots) of Peds Prance animation.

*Peds Prance* (Figure 4.16) involved several characters (not just one as with *Woman with a Bag*) and *Auto Keyframe* worked well in this context because, having defined object relationships once, I could copy and alter these definitions for use with other objects. *Auto Keyframe* allowed me to quickly lay down a number of keys; pressing play I could then watch a character walk, sway or run. In response to what I saw, I then tweaked the keyframes and sometimes the spreadsheets.

In the past I have always created a character’s walk cycle from scratch, usually with a particular outcome in mind and with assumptions about the movement that would best convey a certain emotion or would best suit a particular character model. Apart from the obvious advantage of saving time, automatically generating keys and then responding to what you see feels different from starting with a clear objective and with a “blank slate” (i.e. with no movement). The difference between these approaches is subtle and difficult to describe but it’s true to say that with *Peds Prance* there were times when I felt that I was helping characters become what they wanted to be.

*Flocking Whippet*

A common way of working with movement suggested by 3D animation software is to use computer simulation. This means using a computational model which has been designed to simulate the behaviour of a system. 3D software simulations usually involve the use of particles. A particle is a point in space which, by default, doesn’t render but can be visualised in a number of ways. We can add colour to particles or we can replace them with geometry (as I did for *Flocking Whippet*, Figure 4.17 and 4.18). 3D animators commonly use particle
systems to create things such as “dust, fire, rain, snow, flocking birds, swarming bees, or magic pixie dust” (Beane, 2012, p. 214). For phenomena such as dust and fire, particles are likely to be animated using virtual forces (e.g. virtual wind or turbulence) but for flocking birds or swarming bees, particles might be treated as autonomous characters or autonomous agents (C. W Reynolds, 1999) and this is the approach taken in Flocking Whippet.

![Flocking Whippet animation with 500 whippets.](image)

In Figure 4.17 you can see 500 whippets running together. The location and orientation of each whippet corresponds to that of a particle. For each frame in this animation the position of the particles is calculated by the software according to algorithms (or rules) which I have written using VEX, Houdini’s native coding language. In a sense, the movement of each whippet has not been directly specified by me, the animator, because, although I have written computer code specifying rules to describe the system, I have not specified the location of each individual particle.

Flocking Whippet was completed while participating in an online Houdini course hosted by computer graphics artist Shawn Lipowski, and it uses an approach to character animation based on the model proposed by Craig Reynolds in his 1987 paper, "Flocks, Herds, and Schools: A Distributed Behavioral Model" (Craig W. Reynolds, 1987). In this paper, Reynolds outlines how the behaviour of individual particles within a system can be defined by a few simple rules. In Flocking Whippet, the behaviour of particles is defined by rules for alignment, separation and cohesion. These algorithms ensure that the orientation of a particle is similar to that of its nearest flockmates; that it doesn’t get too close to its flockmates, and that it doesn’t get too far away. Each particle (or autonomous character) behaves according to these simple rules but complex flocking patterns emerge because the position of one character is determined by the position of others.

![The same Flocking Whippet animation rendered through another camera.](image)

While making Flocking Whippet I never tired of tweaking algorithms and playing through the animation, enjoying the fact that the whippets seemed to have a mind of their own and ran in a new formation every time. As described by Reynolds in his paper, when working with simulated characters, it can feel like you are working with something that is alive. Reynolds notes that “One of the charming aspects of the work reported here is not knowing how a simulation is going to proceed from the specified behaviors and initial conditions; there are many unexpected, pleasant surprises” (Craig W. Reynolds, 1987, p. 27). He also notes that there are times when our lack of control over simulated characters can be frustrating (Craig W. Reynolds, 1987, p. 27). The fact that character movement is not directly controlled by the user prompts Reynolds to suggest that “the person who creates animation with character simulation might not strictly be an animator” (Craig W. Reynolds, 1987, p. 27).

Given his insistence that an animator should precisely define their goal and achieve it by adhering to the 12 principles, Lasseter might agree with this sentiment. Published in the same year (1987), the papers by Lasseter and Reynolds describe different approaches to computer animation. Most contemporary 3D packages are designed to accommodate both these styles of approach, i.e. a “hands on” approach where character movement is set using key frames, and a more “indirect” approach where the user manipulates attributes of a system and the software is left to calculate the details.

Flocking Whippet started off as a flock of birds and ended up as a flock of whippets running frantically in a white void. I didn’t set out to make these characters and their environment but they evolved through a process characterised by discovery and response. I started by exploring the movement of particles and later decided to confine particle movement to a horizontal plane and add whippets. Whether using traditional methods (like
those outlined by Lasseter) or using computer simulation, in 3D character animation we usually start with a character model (built in a static pose) and then make decisions about how that character will move. *Flocking Whippet* differed slightly from this approach because I started with movement and added characters to suit. At the time, the idea of characters emerging through movement intrigued me and the movies below (in Figure 4.19) show how I briefly explored it further.

**Force First**

Starting with a particle simulation, *Force First* began in the same way as *Flocking Whippet*. However, in this experiment, instead of adding whippets, I added simple cubes to the particles and linked some of their properties (e.g. colour and shape) to their movement.

![Figure 4.19: Force First playblast (showing particle movement without geometry attached) and rendered animation.](image)

I’m still intrigued by what kind of characters could emerge through this approach, and would like to explore it further. A painter can begin by making abstract marks on a canvas. Prompted by what the marks suggest, the painter can then progressively add marks until recognisable forms emerge. The approach taken in *ForceFirst* is analogous and suggests that movement, rather than painted marks, can act as suggestions.

When he was writing, in 1987, Reynolds was describing an approach to animation which was revolutionary, but today simulation tools have become an indispensable part of most 3D animation packages. Knowing how to use these tools is part of being a 3D animator. But knowing how to use simulation tools is not the same as knowing how they work. *Flocking Whippet* and *Force First* involved writing simulation algorithms rather than simply using existing tools, as I have often done in the past. This experience shifted my view of computer simulation; I now appreciate these tools as ingenious ways of describing things but not as providing an accurate or definitive explanation.

As my research progressed, I learnt more about computer simulation and computer programming. As well as enabling me to create my own tools, acquiring programming experience shifted my general perception of 3D software. As my skills improved, I began to see that software is comprised of packages of code which can be explored and which potentially can be repackaged. I realised that before acquiring these skills I had experienced orthodox 3D tools and procedures as somehow “natural”, settled or fixed. To an extent, I had even assumed they were the final word in accurate representation.

> According to Rushkoff, we don't necessarily need to learn to program but we do need to learn that programming exists (Rushkoff, 2010, p. 8). It was by writing computer code that I gained awareness of its existence.

**Modelling as animation**

A 3D character is usually created at the centre of the world and in a default pose. And like other 3D objects, it is usually modelled in isolation, removed from any context or environment. “Modelling as Animation” is the name I have given to an unorthodox approach to 3D character animation enabled by a custom UI that I called *Modelling as Animation* (see Figure 4.20).

![Figure 4.20: a) Viewport snapshot taken while coding the Modelling as Animation custom UI; b) Viewport snapshot showing the UI in use.](image)
The Modelling as Animation workflow uses only observational sketches as reference, with no pictures or movies downloaded from the internet. Importantly, this workflow doesn’t require the model to be created in a default pose. Instead the model is created in an observed pose and, as the internal skeleton moves from one pose to the next, the deformed model is amended to fit the new pose. One result of this approach is that the model’s topology can be kept very simple and still capture key pose characteristics. Another result is that the model stretches and deforms in unusual ways.

The Modelling as Animation UI allows an animator to work quickly; one click of the red button completes a collection of tasks that might otherwise take several minutes.

With observational sketches as reference, the user starts by creating a simple polygon model. They then create joints (an internal skeleton) and attach (or bind) the model to the joints, which are then animated to deform the model. Up until this point, the only difference between Modelling as Animation and a typical character workflow is that it uses an observed pose, not a “relaxed”, generic or default pose. As I was to discover, this small change has significant ramifications.

Figure 4.21 Diagram showing steps involved in the Modelling as Animation workflow.

Because the model is simple, moving from one pose to the next requires model alterations. For example, at the start of Conference Figs (viewport snapshots shown above in Figure 4.21 and animation shown in Figure 4.22, below) the man’s hand and his head are one continuous geometric form but, as the man moves his hand from his head, that one form has to be separated into two. Scrolling through the timeline, the character deforms, and the user responds to these deformations by making changes to the model: adding detail, removing detail, joining models, or splitting them in two. As they work, the user periodically clicks the red button to duplicate the model, set visibility keyframes, and bind the model to the joints. I have described the workflow here as a linear set of steps but, after creating the initial joints, these steps are repeated many times, and in any order.

Conference Figure

Conference Figure was the first animation made using the Modelling as Animation UI. It took about two hours to make using the tools as described. What you see in Figure 4.22 is a number of different models, appearing and disappearing in quick succession. "Glitchy" moments (when the model deforms in unusual ways) are remnants of the unforeseen outcomes or accidents that have been deliberately kept.

Figure 4.22 Conference Figure animation.

In 3D animation, there are a number of modelling conventions which are aimed at minimising unpredictable deformations; examples include regularly deleting construction history and using three or four-sided faces. Modelling as Animation abandons these conventions and welcomes unexpected forms as suggestions to respond to; to keep, alter, or discard.

Like straight-ahead animation techniques such as paint-on-glass, charcoal or plasticine animation, this workflow encourages improvisation. For example, without knowing what this man’s face would look like, Figure 4.22 shows the enjoyable process of adding detail and watching the character emerge. As well as working straight ahead, with this workflow it’s easy to move backwards in the timeline. An animation can be amended at any time by adding or deleting models, and making timing or movement changes.

MasA whippet
In order to compare *Modelling as Animation* with a standard approach to 3D character animation, I created *MasA Whippet*, which uses as its basis the animated skeleton of *Default Whippet*.

![MasA Whippet animation](image)

*Figure 4.23: MasA Whippet animation.*

With *Default Whippet*, the production process was a means to an end. I was after a particular outcome but exactly how I achieved that outcome didn’t seem important – as long as the process was efficient. The way that I interacted with the software while making *Default Whippet* (e.g. whether I worked slow or fast or was bored or excited) is not evident in the final animation. Figure 1.6 ([above](introduction.html#a-typical-3d-workflow)) shows the dog model in various stages of completion, but these iterations of the model do not appear in the finished work. This means that the hundreds of intuitive decisions that I made while modelling, as well as the speed or style in which I worked, are not explicitly evident. By contrast, *Modelling as Animation* incorporates some of these decisions.

With *Modelling as Animation*, the process is visible in the outcome. This, along with the continual discovery of surprising forms, combined to make it an engaging experience. Without the need to achieve a single perfect model, I found that there was no temptation to obsess over one particular moment in the model’s evolution. Instead I was compelled to work fast, making many decisions quickly and duplicating the model at regular intervals.

*With MasA Whippet, much more than Default Whippet, I sensed that the process itself, not just the outcome, was important.*

**Working from sketches**

Most of the animations described in this research are motivated by everyday things in the world around me. Many of these animations use observational pencil sketches as reference and others involve working directly from life.

![Sketchbook page showing some sketches of Ginger](image)

*Figure 4.24: Sketchbook page showing some sketches of Ginger.*

These practices (using observational pencil sketches as reference and working in the physical presence of things) are common among drawers and painters but they are less common among 3D animators. Comments made in the 2012 book “3D Animation Essentials” indicate why this might be the case. In this book the author gives the following advice to budding 3D animators:

> Let’s say you need to model a tiger. Taking your desktop computer to the zoo and setting it up in front of the tiger exhibit is not really practical. Of course, you could take your sketchbook with you to the zoo and sketch a tiger – and this is a good way to get reference – but this would be time consuming. You could also take a camera to the zoo and take pictures, which would save you some time, but you’d still have to factor in the travel. As an alternative, the Internet is a great place to find images that you can use for references. It’s fast and you don’t have to leave your studio. (Beane, 2012, p. 85)

The author’s words are pragmatic and sensible and they reflect views dominant within the 3D animation community.
Even without having read this book, most 3D animators (including myself prior to this research), intuitively adhere to this advice. But is there something different about experiencing a tiger face to face instead of learning about tigers through internet images or books? If a 3D animator stayed away from internet images and photographic reference, how would their work be different?

Figure 4.24 shows the page from my sketch book which inspired Spline Whippet, discussed above. As we can see from these small scale dog studies, pen or pencil sketches are often indicative and unfinished. This is particularly true when sketching moving things (such as a person or a dog). These sketches are unfinished because I could only put down a few lines describing the dog’s head before she moved to look in another direction. The sketchy quality that we see in these images can be contrasted with 3D software's propensity to depict stable and completed forms.

**Phone Figure**

Like many of the animations described above, Phone Figure was inspired by a pencil sketch. For other projects (such as Spline Whippet and MaxA Whippet) I referred to a full page of drawings, but Phone Figure is based on just one (shown in Figure 4.25 a). With this sketch (which is the size of a postage stamp) as my only reference, I created a 3D character model. Rather than using one final model, Phone Figure (like the Modelling as Animation projects described above) cycles through various model iterations.

With Modelling as Animation, the user presses a button to save iterations of a mesh – but for Phone Figure, I developed a script which does this automatically, at prescribed intervals. This amendment means that duplication of the mesh now happened in the background while I worked and I no longer had to make decisions about when to duplicate the mesh. This made it even easier to achieve a state of “flow” (Csikszentmihalyi, 1990), i.e. to get lost in the process of modelling. With mesh duplication happening in the background it now seemed like I was working on a single model – however it felt different from standard polygon modelling because I knew that my working process was being recorded.

![Figure 4.25: a) Sketch that I referred to while modelling Phone Figure; b) Phone Figure animation.](image)

Inspired by an ambiguous sketch, I took Phone Figure in a number of subtly different directions (has he got a coat on? Is he wearing glasses?) and, in the end, I was left with a multitude of models which were like crumbs left to mark a journey. The movie (Figure 4.25 b) cycles through some of these model iterations.

Throughout this research I have found that Working from Sketches is a good way to avoid illustrating an image or idea that already exists, and instead to let the work itself take over. This is because sketches are “sketchy”; they are incomplete and don’t have the same authority as photographs or anatomy diagrams. Sketches can act as triggers or points of departure and, once the 3D project is underway, attention easily turns from a sketch toward an appreciation of the work on its own terms. In other words, the 3D work (in its current iteration) easily becomes the point of focus and calls for responses that aren’t necessarily planned in advance.

**Building Phone Figure**

In the foreground of the movie below (Figure 4.26) is Phone Figure, and in the background is a model of a building inspired by an apartment block situated across the road from the studio where I sometimes work. It is not an architectural masterpiece, but this building has often grabbed my attention on a sunny day because of the way that light plays across its facade. One afternoon I decided to work from the front seat of my car and, juggling a laptop computer and a graphics tablet, I looked out the windscreen at the building.
For **Building** I used the same basic approach as **Phone Figure**; a script that automatically duplicates models while I work. **Phone Figure**, like other projects described above, was made in the comfort of my studio – but for **Building** my working conditions were far from ideal. For example, the laptop screen size was limited and the sun setting behind me made it difficult to see the viewport display. **Building** was difficult and uncomfortable but it was also absorbing; I found that working from life feels different to working in the studio using sketches or other reference. With projects described above I was mainly responding to digital things (the screen display, the software architecture and the animated work), but **Building** involved an ongoing response to physical things in front of me. After my first life-modelling session I noted that “I delighted in various shapes and shades that I discovered as my eye moved across the forms, and I wanted to record these shapes and shades quickly” (Appendix A, Project Diary: Plein air Still life, 14/05/14). Despite the inherent discomfort and inefficiency, I decided to further explore **Working from Life** and made a suite of tools to help. I call this custom toolset **Plein air Still life** and describe it in detail below.

**Working from Life**

![Tools used when working from life](image)

By “working from life” I mean working in the physical presence of the things which I am studying. This is a strategy I have practiced many times in the past with painting, and which I continue to practice with drawing.

For Franck, Monet, Cézanne, and many other artists using pencil or paint, working from life is common practice. As mentioned above, it is not a strategy that I have previously used (or seen used) with 3D animation software.

**Plein air Still life**

One of the purposes of the **Plein air Still life** UI is to make a number of Maya’s different tools easily accessible in one location, minimising the need to navigate the software’s many menus. Like other tools discussed above, **Plein air Still life** automates several processes so that a single button performs multiple actions, enabling workflows that would otherwise be untenable. Like **Modelling as Animation**, a major feature of **Plein air Still life** is that it automatically saves iterations of a mesh. The video, diagrams, and description below further explain the tool’s features.

![Video showing Plein air Still Life tools in use](image)

**Plein air Still life** tools can be conceptually divided into two sets. The first set (which I refer to as “Production Tools” see Figure 4.29) is most useful during a life modelling session. These tools include things such as a colour palette, which makes it quick and easy to create a shading network and apply it to selected faces, as well as controls to specify how often a mesh is duplicated.
The second set of tools (which I refer to as “Post-production Tools”; see Figure 4.30) allows the animator to easily collate a multitude of models into a single animation by setting keyframes on the visibility of each model.

In Plein air Still life experiments I think of the life modelling session (which uses production tools) as a way of collecting data or “raw material”, which the animator then works with to create an animation. Using Plein air Still life post-production tools, the raw material can be arranged in a variety of ways – which means that the same life-modelling session can result in a variety of different outcomes.

Plant

Plant (Figure 4.31) is the result of my first life-modelling session using Plein air Still life tools. Like Building, Plant started as a polygon cube and, while studying the plant in front of me, I altered the cube by adding more and more detail, applying colour to faces of the mesh as I worked.

As well as simply storing iterations of a mesh, Plein air Still life also incorporates the option to blend or morph between mesh iterations. It achieves this by using Maya’s blend shapes (known in other 3D programs as morph targets). Instead of simply showing a number of models in sequence (as was the case with Modelling as Animation, Phone Figure and Building), in Plant the computer is interpolating between plant models. By adding the ability to interpolate between models, I hoped to create animations which were more than simple timelapse modelling videos. What I didn’t predict was the glitches that this addition to the workflow would introduce.

In the movie at Figure 4.31 there are moments when the plant seems to quiver, other moments when it turns itself inside out and sometimes it almost disappears completely. These glitches of varying intensity all result from the use of blend shapes to morph between model iterations. When model amendments are made by repositioning existing vertices, the software interpolates easily between one model and the next. When amendments involve the addition of new vertices, these newly defined points are left behind while the others slide into position. When amendments involve deleting existing vertices then the software needs to reassign numbers to vertexes on the model, and major glitches occur. With practice, I developed a feel for how particular topology changes would elicit particular styles of glitch, but there were always surprises.
Making Plant was entirely absorbing and, in many ways, it felt similar to observational pencil sketching, with the addition of a virtual third dimension. As well as enjoying the process, I was intrigued by the outcome. I especially enjoyed the moments of quivering or twitching that occur throughout the animation. These glitching twitches are the result of my intuitive modelling decisions in combination with intricacies of the software’s architecture.

Plein air Still life projects involved paying attention to everyday things in the world around me. With these tools I could build things as I see them. This felt different from approaches I had used in the past, which focussed more on how an object is “known” or assumed to be.

For example, it might seem obvious that a plant is comprised of leaves and branches or stems, and creating a 3D model of a plant normally involves assessing it as a collection of these (or similar) component parts. This is true whether using a Paint Effects plant or using other standard approach to plant modelling. In Plant I wanted to avoid orthodox approaches in order to explore the thing in front of me with fresh eyes. To use Franck’s words, I wanted to explore a particular plant without considering it in terms of “plants in general”. As I worked on Plant, the mesh was often in a chaotic state and, reflecting on an afternoon of modelling, I wrote in my project diary:

resisting the urge to make sense ... i.e. resisting the urge to “wrap it up”, to simplify, to make the model look obviously “plant like”; by this I mean that I resisted extruding a stem from the base and then leaves from the stem/trunk. I tried to sit with it being chaotic. (Appendix A, Project Diary: Plein air Still life, 27/05/14)

While working on Plant, I had to continually resist the urge to “tidy up” my model and see it as a collection of leaves and stems. To borrow Merleau-Ponty’s words, I had to exercise a “tolerance for the incomplete” (Merleau-Ponty, 1993b, p. 88).

Castlemaine and Falls Creek

Inspired by this new approach, I went on to use Plein air Still life tools in a variety of locations, including my car, my house, cafes and public buildings as well as in parks and in the countryside. I also used them at different times of the day and I used them to study a variety of different things (e.g. plants, cars, buildings, people and animals).

For Castlemaine (Figure 4.33) I took a folding chair, a card table and my laptop computer into the countryside. While modelling Castlemaine I was surrounded by natural/organic/chaotic forms and I was somewhat overwhelmed by the complexity of my surroundings. It’s common to model a 3D landscape by creating different types of objects using different tools (e.g. one set of tools might be used for creating grass while another set is used for modelling rocks). As well as using different toolsets, each object is likely to be created separately before being moved into position. You can see in the Castlemaine excerpt (Figure 4.33) how I described many different things, including plants, rocks and trees using a single mesh.

For Falls Creek (Figure 4.34) uses a similar approach but in this project I modelled architectural forms within the landscape. Although there were shrubs and small trees within sight, I was compelled to focus on a small hut and ski lift pylon because, when polygon modelling, it’s easier to create hard edged geometric forms than subtle, ambiguous or organic ones.
Like most of my life modelling sessions, my work on *Falls Creek* was interrupted when my laptop battery ran out. At this point I returned to my hotel room and I continued to model what I saw. This is why *Falls Creek* transitions from an exterior to an interior scene.

**Boots**

Like many of my experimental animations, *Boots* avoids breaking a scene into component parts and it also avoids duplication. The four boots in Figure 4.35 are treated as one mesh even though (as two pairs) it would be more convenient to model just two boots, duplicate them, mirror them, and then move them into position. One result of modelling things in context is that they are not aligned to the grid. This makes manipulation tools, such as move, rotate and scale, difficult to control and means that model amendments are always imperfect.

![Boots Animation](image)

As you can see in the movies above, many of my *Plein air Still life* projects avoid complex lighting algorithms and they also avoid smooth shading. Instead of defining the value of shading parameters (such as diffuse, reflectivity and specularity) and getting the software to calculate tonal and colour modulation, in these projects I have applied shades of colour directly to objects. I refer to this strategy as *Colour as Light* and it could be described as a back-to-basics approach to 3D software (discussed above context-of-practice.html#back-to-basics-and-the-low-poly-aesthetic). Like modelling things in context, *Colour as Light*, by default, produces inaccurate results.

**Colour as Light and Geometry as Shadows**

Working with *Plein air Still life* encouraged me to explore dynamic and contextual features of perceptual experience. Focused on the physical world around me, I noticed how things are always moving, and lighting conditions are always changing. I also noticed how shifts in attention and changes in context changed what I saw. Using *Colour as Light*, I found that adding a colour to a mesh changes the look of existing colours on that mesh by changing their context. For example, adding a bright shade of orange makes existing colours look bluer. I also noticed that, as the hours passed and I became more attuned to the things in front of me (the shoes, the plant, the dog or the building), I could always discern more detail and find new colours.

I created *Building* (Figure 4.26, above) as the sun set behind me and I noted in my project diary that:

> as the light changed I wanted to change the topology of the model (e.g. move verts of shadow faces upward) ... These topology changes were reflected or captured with the duplicated mesh, however the colour changes where not reflected. (Appendix A, Project Diary: Plein air Still life, 14/05/14)

What I had noticed when making *Building* is that adjustments I made to shaders throughout production had no impact on the finished work.

**Chair**

![Chair animation](image)

The way that a painter refines colours as they work is captured in layers of paint. In my search for something similar, I added an option to *Plein air Still life* tools which allows the user to save “colour tweaks” (i.e. changes made to shaders during production). With this addition I could use *Plein air Still life* post-production tools to
cycle through saved colours. In *Chair* (Figure 4.36) colours stored during a life modelling session are cycled through in a variety of ways.

**Shoes**

Figure 4.37: *Shoes* animation.

Cycling through saved colours is also evident in *Shoes*, Figure 4.37, particularly when shades of white and blue moving across the model. Obviously storing RGB values and cycling through them in a finished animation bears little resemblance to the richness of layered oil paint, but it was still an interesting addition to the *Plein air Still life* toolset.

**Auto Camera**

Figure 4.38: Viewport snapshot showing the many cameras created when using my Auto Camera script.

While I enjoyed the process of modelling from life and found the results of my life modelling sessions intriguing, I often wondered how I could turn animated models (which exist as three-dimensional data) into two-dimensional movies. My tendency was to position a virtual camera at a point relative to the mesh which loosely corresponded to my real world position relative to the things I was studying. Alternatively, I positioned the camera so that it framed parts the mesh that appealed to me. In most projects, the position of the camera moves between these two alternatives.

In order to remedy this arbitrary approach to camera animation, I decided to write a script which automatically saves the working camera (i.e. the viewpoint that I am working from) at regular intervals throughout the modelling session. The idea was that these cameras could then be culled and collated (based on certain criteria) to create an automatic animation. I thought that “taking the decisions away or at least having a starting point [from which] to work” (Appendix A, Project Diary: *Plein air Still Life*, 13/06/14) might reveal a new approach to camera animation. It took a lot of time to get the script working and ultimately I didn’t find the results interesting so I went back to animating the camera by hand (i.e. using keyframes).

Figure 4.39: Still from *Shoes* animation rendered from a different point of view.

Figure 4.39 shows same project, *Shoes*, from above. You can see how different the shoes look when rendered from a different point of view. Arbitrary camera animation is an issue that remains unresolved in these experimental animations and it would perhaps be more interesting for a viewer to be able to move around and interact with the models, like I can when I’m working. This is something to be explored in future research.

**Books**

In relation to *Plein air Still life* I continued to ask myself, “what makes this different from yr [sic] average timelapse modelling movie?” (Appendix A, Project Diary: *Plein air Still Life*, 30/05/14). I found that one answer to this question was:
the fact that in practicing this technique I actually work (i.e. model and texture) differently ... just like when doing a charcoal animation you might work differently than you would when just doing a charcoal drawing. I’m not necessarily taking the EASY PATH/ the EASY OPTION in terms of making a model. (Appendix A, Project Diary: Plein air Still Life, 30/05/14)

I have explained how this refusal to take the easy option applies to Plant, above. Another example of how Plein air Still life tools changed the way I model is evident in Books, below. In this animation I tried breaking the mesh into parts so that, after working on one part and then another, stored mesh iterations are divergent. Once blend shapes have been added, this divergence results in a dynamic animation. I wondered if this could replicate the way that our eyes move across a form or around a scene.

Figure 4.40: Books animation.

Franck calls his process “Drawing-Seeing” because, for him, drawing and seeing are one activity, not two (Franck, 1973). Similarly, in many of my experimental projects, modelling and seeing became one activity. Rather than first observing the books in front of me and then modelling (or describing) them, it’s more accurate to say that Books involved observing (seeing and understanding) the pile of books through my use of the software. At the time of making this animation the pile of books had been on my table for weeks: making Books meant seeing them in a new way; exploring the ordinary as extraordinary.

Shadow play

Figure 4.41: Viewport snapshot taken while working on Shadow Whippet, discussed below.

Many of my experimental animations use Colour as Light in conjunction with another back-to-basics approach which I call Geometry as Shadows. This strategy involves describing shadows using geometry instead of using virtual lights. Using Geometry as Shadows means that shadows can be manipulated by moving vertices on a mesh which feels more direct than adjusting the parameters of a virtual light. Colour as Light results in colour and shading that is not mathematically perfect and, likewise, Geometry as Shadows results in discrepancies between models and the shadows that they cast.

Using Geometry as Shadows is often as simple as applying a dark (and sometimes semi-transparent) colour to a flat plane (this has been done in Shoes and Boots, above). But sometimes a shadow needs to move and deform with the main model, and for this I created a custom set of tools called Shadow Play. These tools are accessed via four shelf buttons which are used to create geometry that behaves either like shadows cast on the ground or like shadow rays. The movie at Figure 4.42 demonstrates how these tools can be used.

Figure 4.42: Shadow Play demonstration video.

In conjunction with Colour as Light, Geometry as Shadows provides an alternative to the use of complex lighting algorithms and it allows the user to focus on shadows as much as (or instead of) the objects that cast them.
Shadow Whippet

Figure 4.43: Shadow Whippet1 animation.

Created using Shadow Play, the Shadow Whippet animations in Figures 4.43 and 4.44 indicate how new tools can result in unexpected creative opportunities. Without Shadow Play tools I could never have envisaged the animation at Figure 4.44. In this movie the dog model has been hidden, leaving only the shadow geometry.

Figure 4.44: Shadow Whippet2 animation.

Green Jumper

Compared to 3D software, paint and pencil are direct and immediate mediums. The image (or object) that a painter sees is the same image/object that will be before a viewer the moment the artist decides that the work is finished. This is not the case for a 3D user, because a 3D scene (as a digital file) exists as data and software gives us the ongoing capacity to modulate that data in various ways. Throughout the production of a 3D animation, a user has multiple ways to visualise their work; they can change how the viewport displays objects at any time.

For example, throughout production we can view a mesh in smooth shade or wireframe, view rig controls or see the simulation of forces displayed as arrows. The visualisation style that the user chooses will depend on the task at hand, and each style will differ in various ways from a final render. One of the satisfying things about Colour as Light and Geometry as Shadows is that what you see in the viewport while you work is very similar to a rendered image. By comparing the location photo with a still from the Green Jumper animation (shown side by side in Figure 4.45), you can see this similarity.

Figure 4.45: Photograph showing my setup and location when making Green Jumper and still from Green Jumper animation.

Francis Bacon states that, “moving – even unconsciously moving – the brush one way rather than the other will completely alter the implications of the image” (Sylvester, 1975, p. 121), and these words indicate that Bacon pays careful attention to the way that changes in one area of a painting alter the painting as a whole.

Using a digital medium, it’s more difficult for a 3D user to appreciate how and when localised changes alter the implications of their work. This is not such an issue when you are working in a modular fashion toward a predefined goal. But if you are interested in improvisation and emergent content, you need to appreciate a working iteration in order to respond to it appropriately. A digital medium is inherently flexible but, despite this, I have found that bringing the viewport image closer to a final render is one way of encouraging an appreciation of the work in progress (i.e. an appreciation of the work’s current iteration).

Cup and Specular Whippet

Cup and Specular Whippet are two works which use auxiliary computer graphics in a final work. I use the term auxiliary to refer to imagery which we normally think of as visual feedback because it has been designed to help the animator to complete a particular process; one of many processes/steps involved in making a
main/finished work. These are graphics that the software user interacts with but that are not normally seen by a viewer. An example of such imagery is the Default Whippet texture map mentioned in the introduction to this document (Figure 1.11).

![Figure 4.46: Cup animation.](image)

*Cup* (Figure 4.46) shows the animated texture used to colour the cup model alongside the evolving cup mesh.

![Figure 4.47: Specular Whippet animation.](image)

*Specular Whippet* (Figure 4.47) consists of a rendered specular pass which is normally only one component used to create a realistic image.

![Figure 4.48: A compilation of Whippet in the Sun animations.](image)

*Whippet in the Sun*

Along with using Blend Shapes and storing colour tweaks, the addition of transparent trails is another *Plein air Still life* amendment which I made in order to distinguish my work from timelapse modelling videos.

Figure 4.48 shows a compilation of *Whippet in the Sun* animations with fading trails. Figure 4.49 show stills from *Whippet in the Sun* in which the RGB channels fade at slightly different rates, i.e. the fading of these channels is not in unison.

![Figure 4.49: Whippet in the Sun stills.](image)

I found that the addition of fading trails produces some interesting rendered images but it’s an effect that’s difficult to work with because the transparency of objects is often not accurately visible in the viewport.

*Shelves*

Even with the addition of transparent trails, my animations still showed a simple form becoming more complex over time. In order to move away from this visual logic, I decided to add a function that automatically deforms and reduces the model while I worked. With the addition of this feature, the deformation and reduction of a mesh is based on the position of a bounding box. By moving the box, as well as setting the frequency and degree of deformation, subtly different styles of deformation can be achieved. Although the effect is very subtle, you may notice that I have used this automatic deformation function in *Shelves* (Figure 4.50).
Eventually I discovered that modelling moving objects is also an excellent way to avoid the predictable progression from a simple model to a more completed form.

**Cafe Figures**

As well as *Whippet in the Sun* (discussed above), *Cafe Figures* is another example of working with moving objects. The animation in Figure 4.51 combines the results of two modelling sessions which took place in a cafe. Seated inside at a cafe table, I observed and modelled passing vehicles and pedestrians as they waited at the busy intersection outside.

![Figure 4.51: Cafe Figures animation.](image)

I enjoyed working in a café so much that I repeated this activity several times. After the first session I wrote:

*Had a lot of fun. was sorry when my battery ran out. was very absorbed. It was hard to “get anything down”. At first I thought it was a pointless, impossible task. First attempts were conventional Leggo men; I liked it when the man sat down and I moved the verts into position without trying to make sense ... i.e. move a vert to the position of a foot and let the edges be dragged where they will ... try to capture an aspect of the form while letting other aspects “go to the dogs”. (Appendix A, Project Diary: Plein air Still Life, 30/06/14)*

After several working sessions I noted that:

*As I use these plein air production tools I can begin to see in a different way; to approach the subject matter differently/ to see/attend to/ notice different aspects. (Appendix A, Project Diary: Plein air Still Life, 30/06/14)*

![Figure 4.52: Cafe Figures Colour Test animation.](image)

![Figure 4.53: Cafe Figures MoBlur animation.](image)

After working on location (in the café) with the *Plein air Still life* Production Tools I used the Post Production Tools to iterate between the models and cycle through colour tweaks in different ways. I also experimented with other texturing and render options. The movies in Figure 4.52 and Figure 4.53 show some of these experiments.
There were many occasions throughout this research when I found that a still frame from an animated sequence was intriguing, and often these were images, forms or movement that I didn’t design in advance. A recurring theme throughout my diary is that “THE FORMS I DIDNT BUILD ARE THE MOST INTERESTING” (Appendix A, Project Diary: Plein air Still Life, 30/06/14). Of Cafe Figures I wrote “I like/enjoy/am intrigued by the colours and shapes in these images even though they are not what I would ‘choose’” (Appendix A, Project Diary: Plein air Still Life, 30/06/14).

Figure 4.54: Stills from Cafe Figures animation showing some of the glitched figures that appealed to me.

Similarly, of the last frame of Books (Figure 4.55) I wrote, “For some reason I really like this image (again, it’s an image that is born out of the process; not thought up in advance)” (Appendix B, Project Diary, 30/06/14). Of the image in Figure 4.55 right (another frame from Books), I wrote, “I like the way that this image contains ‘observed colours’. It’s not an image I designed but it’s also not arbitrary” (Appendix A, Project Diary: Plein air Still Life, 30/06/14).

Figure 4.55: Stills from Books animation (described above).

**Plein air Still life version 2**

Toward the end of the research I made major revisions to the Plein air Still life Post Production tools. Like Modelling as Animation, the first version of Plein air Still life iterated through models by setting visibility keyframes but this method of showing and hiding models made it difficult to vary the speed of model iteration throughout the timeline. Rather than using keyframes, the revised version of Plein air Still life iterates by connecting objects and attributes using a selection of Maya Utility Nodes.

Figure 4.56: Video demonstrating Plein air Still life version 2.

Cycling through the models now gives real-time feedback and the speed of iteration can be easily varied. The extent to which these changes alter the “feel” of the medium is significant. If we accept that the life modelling session collects data that is raw material for use in post-production, then we could say that when using Plein air Still life V1 this data (or raw material) is not very pliable; it feels a bit like working with sheet metal. Using Plein air Still life V2, by comparison, feels more like working with a more pliable medium such as clay or plasticine.

**Night Scene**

Figure 4.57: Night Scene animation excerpt.
Using the revised version of *Plein air Still life*, the movie in Figure 4.57 is the result of an evening spent at the RMIT library, looking out the window and modelling what I saw on the street several stories below. After about 2.5 hours (the extent of my laptop battery life) I ended up with vehicles, buildings and half-a-dozen figures in various stages of completion. The next day I continued working on some of the models using *Plein air Still life* tools to store iterations of a mesh as I worked. As described above, I also used *Auto Keyframe* for this project and found that because I was animating multiple characters it worked well in this context.

**Night Building and Whippet in Bed**

At the outset of this research I had trouble breaking away from conventional approaches to 3D software because my habits of use were deeply entrenched. *Working from life* was useful because it sometimes prompted me to intuitively depart from habitual practices and conventions. I found that what I saw sometimes influenced the way that I used the tools, prompting minor adjustments to practices and techniques.

![Night Building animation](image)

*Figure 4.58: Night Building animation.*

For example, sitting in my car on a cold night in a deserted street I modelled the house in front of me (Figure 4.58). Extruding and colouring faces to describe the architecture was relaxing and familiar, but in response to messy areas of vegetation I started pulling faces through each other. This is an example of a life modelling sessions in which physical things called for tools to be used in an unorthodox way.

![Whippet in Bed animation](image)

*Figure 4.59: Whippet in Bed animation.*

Departure from habitual practices also occurred because I had to work fast. For example, scrambling to describe the moving dog when making *Whippet in Bed* (Figure 4.59), I collapsed a number of vertices into one and positioned it to coincide with the dog’s nose. In subsequent projects such as *Whippet in the Sun*, above Figure 4.48, I used this technique (i.e. collapsing vertices and then adding detail) many times.

I’ve *Plein air Still life* tools to study and model a variety of things in a variety of contexts, and I’m still finding new ways of using these tools because with each working session I approach them in a subtly different way. I am often compelled to explore new features and have found that new features sometimes suggest new contexts for use. For example, it was the addition of the automatic bounding box deformation described above which prompted me to model my messy bookshelves (Figure 4.50). There are also instances where the particularities of a working context have suggested new features: for instance, after experiencing a change in real-world lighting while making *Building* (Figure [building|fignum]), I added colour tweak controls.
Coloured Whippet

Coloured Whippet is my favourite work from the Plein air Still life series. While making this work the automation controls were set to save the model very frequently, i.e. every couple of seconds. The Coloured Whippet animation in Figure 4.60 uses a large number of mesh iterations with no blend shapes interpolating between them.

Figure 4.60: Coloured Whippet animation.

Working from life can be uncomfortable, it involves working fast and the outcome is always uncertain. This is true when working in an unusual location and it is especially true when studying things that are moving.

Within a simple repertoire of actions, Working from Life allows physical things in the world around me to call for a subtly different style of response.
5. Outcomes and findings

Through an iterative process of enquiry, this research has developed approaches to 3D animation software which disrupt habits and conventions of practice by fostering a comportment of active receptivity.

Outcomes from this research include a number of short 3D animations as well as seven creative strategies (Playing with Software, Playing with History, Animation as Relation, Working from Sketches, Working from Life, Colour as Light and Geometry as Shadow) and five custom tools (Auto Expression, Auto Keyframe, Modelling as Animation, Shadow Play and Plein air Still life). Appendix C of this document presents research outcomes and findings as a list.

A conversational approach

Surprises, glitches and unexpected outcomes inevitably occur during any 3D animation project, but these are usually encountered by the user as accidents or mistakes to be avoided. I was intrigued by surprising images that occurred while making Default Whippet but I found it hard to incorporate them into my work. In part this is because the modular nature of a many standard 3D workflows means that improvisation is possible only within limits.

My comportment also tended to prevent me from being receptive to the unexpected; because, despite appreciating surprises when they occurred, with Default Whippet my main focus remained on implementing correct procedures and achieving repeatable and predictable results.

The experimental animations, by contrast, were less concerned with achieving specific outcomes and more concerned with achieving specific qualities of practice.

With those animations, rather focusing on my ability to control a situation (which felt like a way of reducing it), I wanted to feel that I was responding appropriately to a situation that is too complex to control and too complex to entirely comprehend or understand.

By shifting the emphasis from outcome to process and from control to response I was able to foster a comportment of active receptivity. This is Heidegger’s bringing-forth, and it can be contrasted with the challenging-forth associated with technological Enframing.

I refer to my approach in these works as “conversational” because it involves listening carefully to the call of things and responding in a way that feels right or appropriate.

All of the experimental projects presented in this research involve responding to the medium (i.e. responding to suggestions from the software) and some of the projects also involve responding to physical things (e.g. dogs, books, people). Whether responding primarily to the software or to physical things, a conversational approach also involves responding to the work as it unfolds. This means responding to emergent content, and making changes that seem to help the work become what it wants to be.

At a certain moment, or moments, during production, a work exhibits its own authority/identity. At these moments judgements of the work are not (entirely) based on its resemblance to reference images or to physical things. I have found that it’s easier to notice these moments when avoiding internet images and Working from Sketches or Working from Life. It also helps to appreciate the viewport display, and not just see it as an indication of a final rendered outcome.

The thingliness of digital objects

A conversational approach to 3D animation attunes the user to the agency of the software and reveals that this agency extends beyond ideas implicit in the software’s design. A conversational approach recognises that 3D software is capable of more and different than it was designed for and that it always has the capacity to surprise.
Lanier (2011, p. 133) suggests that digital objects (including computer software and its constituent parts) are always designed for a specific purpose. This may be the case, but approaching these objects with an actively receptive comportment means encountering them as digital things, in the Heideggerian sense. As a general strategy, Playing with Software helps us encounter digital objects as things because it shifts attention away from what a tool has been designed to do and encourages us to focus on what a tool actually does. We miss qualities inherent in 3D software if we encounter digital objects only as they were designed to be encountered. Whereas, on the other hand:

*Playing with Software reveals digital objects as things, i.e. conceptually inexhaustible with the capacity to continually surprise.*

We can be attentively present in the domain of things, digital and otherwise, by disrupting our own expectations and our own habitual practices. In this research, I have *Played with History* to elicit suggested forms, and I explored *Animation as Relation* to elicit suggested movement. Rather than seeking an efficient production process, I have continually invited surprising outcomes in order to disrupt my own working habits and also to disrupt my own easy interpretation of a work.

Unlike Pixar’s unambiguous imagery, designed to communicate a clear storyline, I have pursued image, form and movement that feels significant or meaningful, but is not so easy to describe in words. Rather than seeking media transparency, or using technology as a means to an end I found that the fact of the digital medium can inform animation content and be (more) apparent to a viewer of a finished work.

It is implied that being “limited only by your imagination” (Pixologic, 2015) is a good thing, but:

*I have found that, by exploring the vitality of the accident, animations and images emerge that are sometimes more interesting, intriguing or poignant than I could ever design, imagine or dream up in advance.*

**Beginning in poverty and seeing essences**

According to David O’Reilley “The problem [with 3D software] is that there is simply too much power and very little control, essentially you get *too much for free*” (O’Reilley, 2010, p. 1). All tools have their associated conventions but I share O’Reilley’s concern that 3D users get too much for free. Cézanne had to learn how to implement pictorial perspective, it didn’t come embedded in his paint brush, and we can assume that his studies of traditional techniques helped him recognise perspective as a convention. Ultimately Cézanne was able to play with this convention and develop his own, looser version of perspective; the version that Merleau-Ponty calls “lived perspective” (Merleau-Ponty, 1993a, p. 64).

For early computer graphics researchers such as Newell, Sutherland and Phong, finding ways to digitally describe objects meant looking at things in the world around them in a novel way. Although their approach to these things (e.g. a teapot or a car) could be described as “objective”, because they attempted to disregard context and meaning, these researchers examined their own perceptual experience and, in abstracting simple models from complex experience, they were likely aware of the limits of their own models. But for those of us who use tools based on their algorithms, 40 years after they were designed, it is easy to forget the limitations of these models. For contemporary 3D users who don’t need any knowledge of programming, there is a danger that we may mistake algorithmic models for an explanation of the way the world is, or as the definitive way of accurately describing it. As 3D software becomes more sophisticated and easier to use, this danger grows rather than diminishes.

This research finds that the acquisition of programming skills and technical knowledge is liberating for a 3D user in a variety of ways. Several programming languages have been used throughout this research, but most programming has been done using Python. In learning Python, an object oriented language, I have begun to appreciate coding as a practice of defining essences. With an understanding of some of the practices involved in the creation of 3D software, I can now appreciate that seeing (and defining) essences is different from working with essences defined by someone else. As well as becoming familiar with practices involved in creating 3D software, programming skills and technical knowledge can reveal conventions and assumptions inherent in standard uses of 3D software; they can foster an appreciation of these conventions, as well as a healthy disregard for them. An example of this is the way that acquiring technical knowledge helped me to escape sculptural metaphors which are common in 3D animation packages.
Designed to be intuitive and to hide its algorithmic workings from a user, ZBrush (Figure 5.1) is an excellent example of software built on a sculptural metaphor. A ZBrush user is not encouraged to know anything about the technical specifications of a virtual mesh; they are instead encouraged to approach 3D modelling as though they were working with clay (Pixologic, 2015). Many users find this liberating and associate this non-technical approach to computer software with artistic expression and creativity. In contrast to these users, I have found that liberation is achieved by gaining awareness of what goes on behind the user interface.

A 3D mesh exists as a dataset which specifies certain points in virtual space, and the polygons that connect them. In my research I gained an appreciation of this fact through activities such as creating a polygon mesh by writing code instead of moving icons in the viewport. An example of such an activity is shown in Figure 5.2. Although a mesh dataset includes a virtual third dimension, unless it is 3D-printed it will be used to produce a 2D image, not a 3D object. This is true even for stereoscopic images, which use a slightly different 2D image for each eye.

Sculptural metaphors imply that a mesh should be stable and coherent or even that it should look like it could stand up to gravitational forces. But 3D modelling is not sculpture and there is no reason for a mesh to look like a completed form; there is no need for it to be a stable throughout the timeline or even to “make sense”. Sometimes a polygon dataset is ambiguous, the software gets confused, resulting in glitches. Letting go of sculptural metaphors, we can see that confusing the software is not necessarily a bad thing.

There is a lot of knowledge embedded in computer software, but it is fruitful for a 3D animator to try and begin in poverty. One way of achieving this is to take a back-to-basics approach to the software, which means avoiding modelling presets as well as complex lighting and shading algorithms. To this end my research has used Colour as Light and Geometry as Shadow - strategies sometimes enabled by custom tools such as Plein air Still life and Shadow Play. These strategies are a way of minimising the number of decisions (about the essential attributes of a thing) that have been made on my behalf and they allow me to explore the accidental surface appearance of things. Like Monet, these strategies allow me to see everyday things as abstract areas of colour and to (almost) forget what it is that I’m looking at. By bringing the viewport display closer (in appearance) to a final render, these strategies also encourage me to appreciate the screen image, i.e. to pay attention to it and to not assume it is merely indicative of a final rendered image. They encourage me to notice when small local changes alter the image as a whole, and to be attentively present at the moment when pixels on a screen become a recognisable object.

The world of perception

Continued advances toward the achievement of photorealism in 3D computer graphics suggest that the way something appears to a camera can be convincingly simulated using computer algorithms, but if we examine our own perceptual experience of things we find that it is too ambiguous and elastic to be captured by formal mathematical models, no matter how complex. Although (or perhaps because) it is “going against the grain”, an examination of perceptual experience is a potentially rich site of enquiry for 3D animators.
For CG researchers aiming to simulate reality, things exist in the world “out there”, separate from our experience of them. Teapots, cars, billiard balls, scarves, trees, snow, even skin: CG seeks general models for convincingly (and objectively) describing these things and uses photography as its benchmark. But when we examine our own experience of things, we find that things are never entirely separate from who and what we are.

One obvious reason for this is that what constitutes a thing (as experienced) depends on the types of distinctions that we, as biological entities, are able to make. As philosopher and biologist Francisco Varela explains, “there are many ways the world is – indeed even many different worlds of experience – depending on the structure of the being involved and the kinds of distinctions it is able to make” (Varela, Thompson & Rosch, 1993, p. 9).

In accordance with this insight, I have noticed that, although they overlap, Ginger’s world is different from my own. Watching her sniff the air, it’s obvious that her world is populated by things that don’t exist in my world, and the reverse is also true. As well as being based on my biology, the things that populate my world are the result of my cultural background and personal history; they are “a result of a mixture of sensory experiences, emotional responses, memories, prejudices and the like” (Willis, 2001, p. 1). As individuals, cultural groups, or whole species, we can't simply project our own meanings onto things (we can’t just decide to see all dogs as unicorns for example). But equally, we are not simply material entities which respond to things in a mechanical, causal fashion. We exist between these two extremes. Just as making can be conceived as a type of conversation, perception also has a conversational structure: i.e. perceptual experience can be conceived as a conversation between ourselves and other things.

Our relation to things is not simple – it is entangled from the start. But standard approaches to 3D animation make it easy to overlook this fact. Along with its emphasis on human agency (promoting the idea that, for the human user, 3D software is a means to an end), obscuring the role of the body as well as personal experience in perception are ways that 3D promotes a simplistic model of our relations with things.

As well as providing the impetus to disrupt standard practices, returning to the world of perception helps us to notice the complexity of our relation with things, including the mundane things that constitute our local environment.

Mapping process to outcome

When working with paint, pencil or charcoal, the artist’s intuitive movements leave marks which sometimes remain visible in a finished work as multiple overlapping brushstrokes, graphite lines or smudges of charcoal. In a digital medium, there is no longer an indexical link between the artist’s actions and the traces that they leave. For the viewer of a 3D image or animation, unconscious or intuitive decisions, movements and marks made by a user are often difficult discern. This research has developed several strategies and custom tools that capture something of a user’s intuitive actions. The most successful of these are Modelling as Animation and Plein air-Still life, which record iterations of a model while I work.

Using strategies and tools developed in my research, a finished 3D animation can embody something of a user’s intuitive decisions made throughout production.
6. Discussion

This section begins with a discussion of style. I argue that all things exhibit a style and it is useful to think of 3D animation as a stylistic exchange between the style of an animator and the style of other things.

Part of this exchange involves the style of computer hardware and software. Due to its inherent granularity, computing incorporates the style of a variety of things which operate at different scales and on different levels. At a level above electronic circuits and digital objects is the software’s architecture and interface which have been designed to afford a particular user experience. Above this, is the style of product marketing and practices promoted by help menus and user forums. A detailed analysis of how each of these things/elements interact is beyond the scope of this research, however, the inherent flexibility of digital objects and the dominance of a default 3D style suggests that marketing and user experience design (UX) often dominate a conversation. My research has shown that custom tools and unorthodox practices can help users escape this domination and interact with a digital medium which feels ( stylistically) different.

The style of a 3D animation can be changed with the click of a button which leads us to believe that style and content are separate. But this is not entirely true because a work communicates on the level of style, not only through what it represents. This is especially the case with unfinished, sketchy works. Comparing my experimental animations with the default visual style ( exemplified by Pixar), I find that mine are more sketchy and recognisable things are harder to pin down. This ambiguity highlights how viewers bring their own style to a work and how meanings emerge between the two. Like the exchange between animator and thing, there is an exchange between the style of a viewer and the style of a work.

Phenomenological enquiry is the starting point for my experimental animations but we can see that the finished works embody a distinctly non-human gesture reminiscent of digital objects and object oriented programming (OOP). This insight leads us to consider OOO, which conceptually situates humans on the same level as other things (including physical, digital, real and imagined things). OOO focuses as much on the relationship between (inanimate) things as the relationship between human and world.

Merleau-Ponty’s work encourages us to regard the making and viewing of 3D animations as a stylistic exchange and similarly, OOO allows us to think in terms of collectives and entanglements. Rather than seeking deliberate, detailed representations, my research explores how animations can emerge from entangled relations.

Style

For Merleau-Ponty (and for Heidegger), a successful artwork communicates primarily on the level of style, not through what it represents. In other words, an image or an animation embodies a particular attitude or comportment toward the world, communicating “at a level more fundamental than the sense making judgements of the mind” (Gilmore, 2005, p. 302). It is this “pre-objective” level of communication to which Merleau-Ponty is referring when he says of a Lascaux cave painting that, rather than seeing it, I see according to it, or with it (Merleau-Ponty, 1964, p. 164).

Bacon is expressing a desire to communicate on this level when he says he wants to bypass the brain because he is after something “more poignant than illustration” (Sylvester, 1975, p. 17). According to Bacon, paint that is illustrative “tells you the story in a long diatribe through the brain,” while paint that avoids illustration “comes across directly onto the nervous system” (Sylvester, 1975, p. 18). Noting that it’s difficult to exactly define the difference between paint that is illustrative and paint that is not, he says that “It’s something to do with instinct” (Sylvester, 1975, p. 18) – and here Bacon is referring to both his instincts as a painter as well as the instincts of a viewer. The process of making a non-illustrative painting involves the instincts of the painter and the finished work communicates on an instinctive level with the viewer.
In 3D animation there is a default visual style exemplified by the work of Pixar Animation Studios. Pixar films exude a kind of dogmatism or self assurance which resonates with the “classical world-view” described by Merleau-Ponty (Merleau-Ponty, 2004, pp. 31, 106). This world-view believes in the possibility of a rational understanding of things (and the world) that holds true for all places and all time. In Pixar animations, individual things (e.g. a collar, an eyeball, a tongue or an individual hair on a dogs coat) are rendered clearly and distinctly and (although they move within the frame) are stable throughout the timeline. Pixar films elicit an emotional response, but not because the status of things and the connections between them is troubled. If we cry watching a Pixar film it’s because we are responding to a carefully designed narrative which unfolds within a fully calculable and stable world.

Merleau-Ponty explains that the dogmatism of the classical world view is implicit in classical works of art whose meanings are unequivocal (Merleau-Ponty, 2004, p. 31). He contrasts these works with Cezanne’s paintings which give things their “unsurpassable plentitude which for us is the definition of the real” (Merleau-Ponty, 1993a, p. 65). Even more than classical paintings, Pixar productions are seamless because they are rendered with mathematical precision. Created by teams of talented individuals, in these films every pixel has been accounted for and has a definitive role in telling the story. This visual style depicts a world in which categories of the intellect have already been applied; things are settled and (on a stylistic level at least) the viewer feels “somehow relaxed” (Merleau-Ponty, 1993a, p. 66). I appreciate these films for their high production values but watching them makes me feel claustrophobic.

Through a series of experimental animations I have moved away from the default 3D style (exemplified by Pixar) toward something more sketchy and incomplete. In these animations precise meanings and forms are harder to pin down because (recognisable) things emerge and subside. Things move and change throughout the timeline so that stable orders and easy definitions are consistently disrupted.

Comparing “Dug the Dog” (Figure 6.1) with Whippet in the Sun (Figure 6.2) we can see that my 3D model is incomplete and inaccurate; it is a mere sketch; an inadequate response to an inexhaustibly rich dog/thing. Though a cartoon dog, Dug is comprised of components that ordinarily define dogs. His tail, ears, nose, tongue, and each hair on his body are all rendered with detail and precision. Whippet in the Sun is more ambiguous and leaves more room for interpretation, highlighting how meanings emerge between a viewer and a work. More than a highly finished work, Merleau-Ponty suggests that a sketchy or incomplete work invites viewers “to take up the gesture which created it” (Merleau-Ponty, 1993b, p. 88). Compared to painters of the classical era, who use standard representational techniques and achieve a high level of finish in their work, modern painters such as Cezanne and Klee display a startling “tolerance for the incomplete” (Merleau-Ponty, 1993b, p. 88). Critics of the day apparently complained that modern artists presented sketches as finished works. Merleau-Ponty explains that modern paintings have an unfinished quality, not because the artists are lazy or inept, but because (like Bacon many years later) they want to communicate with viewers on the level of style and instinct.

Like these modern paintings, my experimental animations are also sketchy and incomplete. We could say that my work invites a viewer to take up the gesture which created it. But what is this gesture? Where does it come from and whose gesture is it?
Obviously the gesture of the artist or animator plays a big part in the creation of a work. Typically, it is a painter’s physical gestures (movements of the arm and hand) that leave marks on a canvas. If a painting is sketchy or incomplete, brushstrokes remain visible and viewers get a sense of the actions that they embody. A painter’s physical movements leave a trace for viewers to gear into or inhabit.

Just like the way that they walk or sign their name, the way that a person moves a brush or a stylus (or mouse) is part of their personal style. The link between personal style and physical gestures makes it easy to understand how a painter’s style is embodied in the physical artefact (the painting) which they create. A painting communicates (at least partly) at the level of the brushstroke and a painter’s brushstrokes easily embody her style because her physical movements leave a physical trace.

As we have seen in this research, the relationship between physical actions, visible outcomes, and personal style is not so simple for 3D animators. For us, the mapping between actions and outcomes is something to play with and explore rather than something to be taken for granted. It is therefore important for 3D animators to remember that a person’s style doesn’t only manifest as physical movements.

A person’s style is evident in the way that they move, but it is also evident the way that they think and the way that they see. For Merleau-Ponty, style describes the way a person gears into the world; it describes the particular way that the world shows up for them. Perhaps more than my physical gestures, experimental animations presented in this document embody my style of thinking and seeing.

Statements by software companies (Pixologic, 2015) encourage us to think that what makes an animation unique is the user’s ability to exercise an idiosyncratic imagination. The insinuation is that 3D software allows users to make their (inner) mental images visible to others. But if we accept this idea then we are in danger of forgetting that learning to use 3D software means learning to see and think in a particular way.

Rather than focusing on style as physical gesture, or style as idiosyncratic imagination, my experimental animations have shown that exploring style as response is a fruitful line of enquiry. A user’s style is evident in their ongoing, intuitive response to tensions perceived in the world and in their work. Capturing a user’s intuitive decisions (and using them as raw material) is one way to create animations that embody a user’s particular style. This research has involved creating custom tools and I would like to suggest that a user’s style can be embodied in custom tools as much as in animated works.

As a way of understanding the role of these custom tools, we can make an analogy between computer coding and mixing paint. Changing the feel of the medium and the trace that it leaves, mixing particular consistencies of paint is one of many activities involved in making a painting. Similarly, coding custom 3D tools is a way of changing the feel of the software, changing the data that the software stores, and changing how that data is used (e.g. how the data is modulated to form an image displayed on a screen). Although I started with a general idea, query or need, each of my custom tools was created through an improvisational process and I sometimes found that new tools suggested new practices or practices suggested new tools.

Human beings exhibit a style and so do other things; For Merleau-Ponty style is what makes a thing what it is. Linda Singer explains that “style is that which secures the harmonious flow of adumbrations which grounds the movement from the thing-seen-from-a-point-of-view to the thing seen” (Singer, 1996, p. 160). Loosely speaking, style is Merleau-Ponty’s answer to Husserl’s search for essences (i.e. the style of a thing is what persists below accidental surface appearance). By saying that style is what we respond to in a thing, Merleau-Ponty (2002, p. 523) is emphasising that things don’t normally appear to us as a collection of qualities or traits; in everyday experience, a thing’s style moves us on a bodily level.

In 3D animation, there are many practices through which the style of particular things is overlooked. For example, as 3D animators we normally try to set aside our relationship with things (e.g. their physical proximity or personal significance) and we usually describe things in terms of attributes which have been defined by someone else. Whether describing a car, a billiard ball, a plant or a dog, we tend to break things into component parts and we work with digital images of things rather than with things that are physically present. As described in section 4 of this document, my attempts to respond to the style of particular things have resulted in numerous alternative practices.

Given that things in the world, and the world as a whole, already exhibit a style, Jonathon Gilmore describes creative practice as a type of debate where the style of things enters into an exchange with the style of the artist (Gilmore, 2005, p. 306).
The dominance of a default 3D style (epitomised by Pixar) indicates that standard approaches to 3D software can obscure, reduce or level the style of the user and of other things. Everything (whether human, animal, animate or inanimate) exhibits a style and style is the level at which communication occurs. This deep understanding of style encourages us to explore 3D animation as a stylistic exchange and escape the grip of the default 3D style.

Like all things, computer software and hardware exhibit a style which calls for a particular response. For example, to see the screen, tap the keyboard and hold a mouse, we intuitively hold our bodies in a certain way. Many 3D practices have emerged as a response to software and hardware design. One way that I tried to push the boundaries of these practices was by working outside but when I did so I needed to sit down on a chair or on the ground.

Merleau-Ponty emphasises the role of the body in human experience and his thought is useful for understanding how users respond to computers (and to other physical things) and how viewers might respond to a work. Merleau-Ponty is primarily interested in how humans (and animals) perceive, but computers are complex machines with a lot going on below the surface (i.e. beyond the realm of our perceptual capacities). Animations presented in this document invite a viewer to take up the gesture which created the work, and it is apparent that this gesture is not only mine. These works embody the gestures of many non-human participants.

For Singer, Merleau-Ponty’s thought explains how: “The style of a musical instrument consists in those particulars to which the musician must adjust when he plays” (Singer, 1996, p. 160). A violinist moves their bow and it presses against the violin’s strings. The musician responds to sounds as well as the weight and texture of the wood and, meanwhile, the strings bend under pressure from the bow. The musician interacts with their instrument and the instrument’s components interact with each other. For phenomenology, these two types of interaction are completely different but for Object Oriented Ontology they are very similar. My works are about the way that I experience things and object oriented ontology insists that they are also about the way that things “experience”, comprehend, or translate each other.

Non-human gesture

Object oriented ontology is a term coined by Graham Harman and it has its roots in Heidegger’s thought. In his 2002 book, Tool-Being: Heidegger and the Metaphysics of objects, Harman expands upon Heidegger’s tool analysis to describe an ontology of objects (Harman, 2002). He initially calls this school of thought object oriented philosophy and later refers to it as object oriented ontology (or OOO for short).

Some interpretations of Heidegger (e.g. (Dreyfus, 1990, 1992)) read his tool analysis as a vote for praxis over theory but Harman insists that Heidegger’s essential point is that neither theory nor practice exhaust objects. Whether I consciously study a cup or use it to drink coffee, the cup object itself always has plenty of surprises up its sleeve. No amount of conscious reflection or theorising about the cup can entirely capture it, and neither can any amount of intuitive interaction. No matter how far it fades into the background of everyday use, the cup could surprise me at any moment by breaking or burning my lip. Harman insists that objects are deeper than all their possible relations.

Heidegger’s insights reveal that objects withdraw from human consciousness and Harman extends this to say that objects also withdraw from each other. This is a move that Heidegger would not have liked.

Ian Bogost, a critic and creator of computational media, explores concepts from OOO in much of his work. In his book Alien Phenomenology; or what it’s like to be a thing, Bogost wonders “What’s it like to be a computer, or a microprocessor, or a ribbon or a cable?” (Bogost, 2012, p. 9). Bogost is particularly interested in the “experience” of inanimate things and his descriptions of how computer hardware components interact remind me of some of the non-human gestures at play in my work. My animations are about the way that I experience things and, according to Bogost, they are also about the way that things “experience” each other. Bogost’s “platform studies” investigates hardware and software as actors and insists that a digital work has to be understood in the context of the computing system with which it was made (Bogost, 2012, p. 100).

Bogost’s is an alien phenomenology because he wants to understand a thing “on its own terms” (Bogost, 2012, p. 10) rather than describing a thing from his (human) point of view. Bogost wants to know what it’s like to be a thing and he thinks that computers can give us a special insight into this question.
Bogost creates video games that make arguments and express ideas using computational processes (Bogost, 2010, p. 6). These are games which represent worldly logics with computational logics. Computers allow us to explain processes with other processes and Bogost suggests that this is very useful for an alien phenomenologist. The implication is that by exploring how things work and by crafting (computational) models of things we gain insight into what it is like to be a thing. Reading Bogost’s work alerts us to unprecedented opportunities afforded by digital media but it also brings up a number of questions.

Does a focus on worldly logics mean that Bogost seeks to be impartial; is impartiality possible and is it desirable? Is Bogost alert to surprises while he works? Do viewers get a sense of the withdrawn when playing/viewing his games, or do Bogost’s computational models end up being dogmatic? Further exploration of these questions is beyond the scope of this research but could be pursued in future work.

Rather than “object”, Bogost prefers the word “unit” because it doesn’t have material connotations. Using this word also helps us imagine how objects are comprised of other objects and phenomena are the “emergent effects of the autonomous actions of interrelating parts of a system” (Bogost, 2012, p. 25). He explains that a unit is like a machine, it does something. What a unit does, or what it can do, makes it what it is. Like Harman, Bogost insists on the autonomy and withdrawn nature of objects, stating that “units remain fundamentally in the dark about one another’s infinite centres” (Bogost, 2012, p. 30).

Bogost’s description of the way that units interact reminds us of the particles in Flocking Whippet (Figure 4.17 and Figure 4.18). Isolated and unique, a unit is an enclosed system which becomes part of another system. Bogost explains that emergence and complexity is an after effect of hidden logics (Bogost, 2010, p. 10). Each particle (or unit) in Flocking Whippet operates according to its own internal logic and the (complex) flocking phenomena is an emergent property resulting from each unit’s simple coded rules.

Bogost’s description of units reminds us of digital objects and his work illustrates similarities between object oriented ontology and Object Oriented Programming.

Humans have a very special place in Heidegger’s thought but, for OOO, human beings are just one type of object. Harman insists that physical things, digital things and fantastical things, although not equally real, are all equally objects. Harman’s quest for a philosophical system “which is able to speak of all objects and the causal relations with which they become involved” (Harman, 2010, p. 5) results in what Levi Bryant calls a “flat ontology” (Bryant, 2011, p. 246) or a “democracy of objects” (Bryant, 2011).

Husserl, Heidegger and Merleau-Ponty focus on the (special) relationship between humans and things, but OOO treats this as just one type of relation. Admittedly it is a complex example, but for OOO human perception is no different in kind that thing-thing causality. In other words, the way that I perceive an object (or a thing) is no different in kind from how one object “perceives” (or interacts with) another. All objects partially interact while always holding something in reserve. I can never fully describe, fully use, or fully understand the cup in front of me and likewise the cup can never fully grasp (or exhaust) the table.

OOO blurs the boundaries between humans and other things and encourages us to think in terms of collectives and entanglements. This move disturbs the assumption that, as empowered humans we can/do/should freely manipulate things for our own purposes. By placing humans on the same level as other things, OOO troubles human exceptionalism.

**Blurring boundaries**

Human exceptionalism is a term that describes how we continually come up with stories about what makes humans superior to, and different from, everything else. In a short paper called “Against Human Exceptionalism” (Pickering, 2008), Andrew Pickering describes how these ideas have changed over the centuries. At various times we have assumed that it is the soul, reason, consciousness, language, knowledge, or representation that makes humans different from other things. Pickering suggests that discourses of exceptionalism impoverish us and its time that we moved away from this preoccupation.

3D software is designed to feel empowering and, for proficient users, it’s easy to assume that the software is at our command. Instead of focusing on humans as actors in the world, Pickering prefers to talk about “dances of agency” (Pickering, 2008, p. 3). This term describes a reciprocal and transformative back and forth between human and things. Pickering explains “that dances of agency have their own inner dynamics and an emergent
quality—in dances of agency we find out about and react to the unexpected” (Pickering, 2008, p. 4). As creative practitioners it’s easy to assume we are the ones that act upon (or manipulate) things but, reflecting on work done in this research, Pickering’s description of a distributed agency seems more accurate.

All things engage in dances of agency, i.e. all things engage in “performative and adaptive interactions with their environments” (Pickering, 2008, p. 4). Pickering’s term therefore avoids the obsession with human specialness.

Another theorist who believes that notions of human exceptionalism limit us is philosopher and biologist Donna Haraway. She insists that being human is always “becoming-with”; this includes becoming with other people and becoming with technologies. In recent work she is particularly interested in how we become with other species (Haraway, 2008). For Haraway, human exceptionalism “is the premise that humanity alone is not a spatial and temporal web of interspecies dependencies” (Haraway, 2008, p. 11). Trying to create “limited only by your imagination” (Pixologic, 2015) would be a deluded ambition for Haraway because we could never become based on our own autonomy. Haraway’s insights highlight that, along with computer software and hardware, Ginger has a part to play in who I am and in my work.

Human exceptionalism relies on distinct boundaries (e.g. between animate and inanimate, human and non-human) and, thanks in part to digital technology, many of these boundaries are today being blurred. Despite this blurriness, human exceptionalist assumptions are encouraged by the design and marketing of 3D software. Statements by software companies emphasise the agency of human users, implying that our dreams can become reality and our idiosyncratic imagination can be (transparently) rendered visible to others (Autodesk, 2012; Pixologic, 2015). As free-thinking human actors, we are encouraged to assume that we are the ones in control. These statements imply that we float above the material world and can use 3D software to express something that exists in our minds. Escaping the obsession with human exceptionalism requires that we reject this narrative of empowerment.

**Tolerance for the incomplete**

By keeping human beings at the centre of all relations, phenomenology (arguably) doesn’t trouble anthropocentrism or human exceptionalism. Whatever its limits, I have found phenomenological enquiry to be useful. The shift in focus, from how phenomena appear to how they appear to a living person, has been enough to disrupt my habits of practice and has helped me to question the authority of algorithms, i.e. to notice what is left out of representational models.

Phenomenology insists that we are always embedded in a meaningful world. Our knowledge of things, of the world or of ourselves can only ever be partial because we can never achieve an objective viewpoint from which to gain a full understanding. We can perhaps progress toward this objective but we can never finally get there. Through a series of short animation experiments, I have found that returning to things themselves promotes a certain humility which is evident in the work on a stylistic level.

The default style associated with 3D animation reveals a world that is dogmatic and sure of itself. Merleau-Ponty suggests that, by exercising a tolerance for the incomplete, we can add “a new dimension to this world too sure of itself by making contingency vibrate within it” (Merleau-Ponty, 1993b, p. 88). In this research I have paid close, earing attention to all manner of things and have worked with contingency. In contrast to the default 3D style, my sketchy, incomplete works acknowledge that our understanding of digital objects will always be incomplete and so will our algorithmic modelling of the world.
7. Conclusion

Implicit in the design of 3D software are assumptions about knowledge, creativity and what it means to be human. 3D software has been designed to feel natural and to feel empowering. It is promoted as a medium which enables us to transparently achieve our design goals or to render our imaginative ideas visible to others. For proficient 3D users it’s easy to assume that (as human actors) we are the ones in control. With the click of a button, we construct all manner of virtual objects using complex computer algorithms which seem to accurately describe and explain. We work on powerful networked computers; turn away from local physical things and sometimes feel that we float above the material world like a disembodied mind.

For 3D animators it’s tempting to spend long hours building detailed fantasy creatures and virtual worlds. While these creations might seem highly “imaginative”, they are actually just extraordinary things rendered in an ordinary way. Bathed in glow of a computer screen, it’s easy to overlook the extraordinary in the ordinary; it’s easy to overlook the strangeness of the (everyday) human world. Qualities inherent in digital media can trouble entrenched humanist orders but to explore this potential we need to pay careful attention, question our working habits and get past the glossy surface of interface design.

Today computer software (and OOP) structures many aspects of our lives, most obviously our means of production and our modes of communication. As computer algorithms become an increasingly prevalent and powerful force in our lives, it becomes increasingly important (and perhaps increasingly difficult) to notice their effects. We need to continually question the design and implementation of computer software because algorithms are never neutral. Mathematical models are not simply tools for explaining, describing, or understanding things; they are (potentially reductive) ways of revealing the world.

The challenge facing 3D animation practitioners is how to stay in the messy zone where situations and entities are encountered in their complexity. In order to respond to this challenge we need to overcome the narrative of empowerment, question the authority of algorithms and resist the temptation to think of ourselves as a disembodied mind. We need to continually question the design of computer software and we can do this by delving below the interface, creating our own algorithmic models and noticing what falls through the cracks.

This research does not contribute software solutions. Rather, it serves as reminder that in algorithmic models and blanket solutions of any kind there will always be gaps. This research contributes to animation theory and practice by showing how a conversational approach to 3D software can help users avoid the reductive potential inherent in digital tools.

In the age of digital technology many believe that phenomenology is no longer relevant because it is focused on a human-centric approach to the world. A detailed critique of this view is beyond the scope of this research but my short animation experiments show that phenomenological enquiry can help us to see familiar things in a new way. An actively receptive comportment encourages us to see the ordinary as extraordinary and to see the human world as strange. Based on my research outcomes and influenced by phenomenology I promote a careful, caring attention to things which we normally overlook.

By challenging the idea that the aim of a creative practitioner should be to illustrate a preconceived idea, this research questions prevailing notions of software as merely “a means to an end” (Jones, 2014). Ideas, imagination and deliberate intentions are important to creative practice, but we shouldn’t forget that the tools we use and the skills we acquire impact upon what we see and how we think. Like computer algorithms, animation practices are never a (neutral) means to end and are always ways of seeing, ways of knowing and ways of revealing the world. Why be limited by imagination when so much can be revealed in the process of making? Recognising the generative potential of material practices can help us avoid moving in an increasingly choreographed world.
Bibliography


List of Figures

- Figure 1.1: Stills from “Drive” (Moore, 2001).
- Figure 1.2: Default Whippet 3D animation.
- Figure 1.3: Detail of page from my sketchbook.
- Figure 1.4: Sketchbook page showing some of the sketches I made of Ginger in the park.
- Figure 1.5: Drawing of a horse by Frederick Frank. Reprinted from “The Zen of Seeing” (Franck, 1973, pp. 56–57); Copyright 1973 by Frederick Frank.
- Figure 1.6: This diagram is comprised of stills from Default Whippet and illustrates some of the steps involved in a standard 3D character animation workflow.
- Figure 1.7: The final Default Whippet model. Note that an internet connection is needed to access the 3D data associated with this Figure. Click on the arrow to load the model and then use the mouse to view it from any angle.
- Figure 1.8: The image above left shows composite photographic images of Ginger laid out as a model sheet. On the right is a viewport snapshot showing how these images were used as a guide to model Default Whippet.
- Figure 1.9: Sample of the many internet images and movies which I referred to when creating Default Whippet.
- Figure 1.10: Screenshot from Default Whippet 3D animation showing skinning error.
- Figure 1.11: Screenshot from Default Whippet 3D animation showing digitally painted texture map.
- Figure 1.12: Francis Bacon “Three Studies of Lucian Freud” (detail). Oil on canvas.
- Figure 1.13: Illustration indicating how software modulates digital data to create a screen representation. This illustration uses images from xsens.com and orionstarmedia.com.
- Figure 1.14: This illustration shows the same (Default Whippet) 3D mesh rendered in four different ways; with Blinn shader, Toon shader and a Paint Effects charcoal linework.
- Figure 1.15: Images showing how to choose a birch tree from a number of presets and how to tweak parameters. These image from online tutorial “Designing and Animating a Birch Tree in Maya using Paint Effects” (Kumar, 2013) Copyright 2016 Envato Pty Ltd.
- Figure 1.16: Screenshots showing how to create ocean waves using Maya's Ocean Shader and how to create smoke using Maya's Fuid Effects.
- Figure 1.17: Screenshots from Default Whippet.
- Figure 1.18: Images of Sutherland’s students measuring the coordinates of a VW car, and renders of the resulting car mesh. Image sourced from computerhistory.org.
- Figure 1.19: A car advertisement which uses a computer generated car (Southern, 2012).
- Figure 1.20: This illustration shows a 3D render (i.e. a computer generated image) of an IKEA chair, a more complex 3D scene, and an IKEA 3D artist at work. Images sourced from cgsociety.org.
- Figure 2.1: Catmull’s plaster cast of his own hand; a wireframe render of his digital model; and a smooth shaded render of the same model. Images sourced from Catmull’s 2014 book (Catmull, 2014, pp. 14, 15) and from maacindia.com.
- Figure 2.2: Detail of Newell’s Utah Teapot sketch (computerhistory.org, n.d.); wireframe model of Newell’s Utah teapot created by University of Utah researchers (computerhistory.org, n.d.); and a render of Newell’s teapot.
- Figure 2.3: Visual illustration of the Phong equation. (Wikipedia.org, n.d.-a).
- Figure 2.4: Illustration showing the way that that a Phong reflection model can be used to shade the surface of a 3D mesh. This image combines screenshots from Turbosquid (“TurboSquid,” n.d.) with a screenshot from Maya’s user interface (UI).
- Figure 2.5: O’Reilley’s Shitmaker tool.
- Figure 2.6: Still from Please Say Something, by David O’Reilly (2009).
- Figure 2.7: Still from Between Bears by Eran Hilleli (Hilleli, 2010).
- Figure 2.8: Detail of still from Pivot (Kevin Megens, Floris Vos, Arno de Grijs, 2009); Detail of still from Dropp by Hajime Nagatsuka (Nagatsuka, 2011); Illustration by Mordi Levi; Detail of screenshot from Catdammit! by Polish game designers Fir & Flams (2014).
- Figure 2.9: Detail of still from Matatoro (Mauro Carraro Jérémy Pasquet, 2010), a 3D animation with a hand-made aesthetic.
- Figure 2.10: Still from The Third and the Seventh by Alex Roman (2009).
Figure 2.11: a) “Making of a Bruce Lee 3D portrait”, Youtube video by Alexander Beim.; Published Dec 18, 2012 ; 478,976 views; b) “Tywin Lannister Sculpt Timelapse”, Youtube video by by Adam Fisher.; Published Aug 28, 2013 ; 295,195 views.; c) “Character Modeling Demo Reel”, Youtube video by Patrick Kilcher.; Published Feb 3, 2014. 70,625 views.

Figure 2.12: Two screenshots from The Third & The Seventh – Compositing Breakdown by Alex Roman (Roman, 2010); Followed by an image showing Roman’s book, From Bits to Lens (sourced from evermotion.org).

Figure 2.13: Screenshot showing the ZBrush interface.

Figure 2.14: Details from some of Menkman’s portraits which illustrate her publication “A Vernacular of File Formats” (Menkman, 2010a).

Figure 2.15: "Study for Head of George Dyer” by Francis Bacon, 1967. Oil on canvas.

Figure 3.1: Photograph showing my hand holding Ginger in position while taking “relaxed pose” photographs.

Figure 3.2: Stills from an online video which supplements the paper "Simulating Knitted Cloth at Yarn Level" (Kaldor, James, & Marschner, 2008).

Figure 3.3: Still from an online video describing how Disney created snow simulations for the movie Frozen (Buck & Lee, 2013).

Figure 3.4: Three of Monet's many haystack paintings. “Stacks of Wheat (End of Day, Autumn)”, 1890-91, oil on canvas; “Haystacks, midday”, 1890, oil on canvas; “Wheatstacks, Snow Effect, Morning”, 1891, oil on canvas.

Figure 3.5: Maya Paint Effects’ tree. Image from online tutorial “Designing and Animating a Birch Tree in Maya using Paint Effects” (Kumar, 2013). Copyright 2016 Envato Pty Ltd.

Figure 3.6: Diagram showing how, in OOP, a class can inherit attributes from another class. Image sourced from The Nature of Code (Shiffman, 2012).

Figure 3.7: “The Card Players” by Paul Cezanne, 1894–1895. Oil on canvas.

Figure 3.8: An advertising image for Pixar showing characters from Toy Story (Lasseter, 1995). Image sourced from http://www.disneypictures.net/r-buzz-lightyear-202-buzz-lightyear-woody-3888.htm.

Figure 3.9: Illustration of “depth of field” from the book 3D Animation Essentials by Andy Beane (2012, p. 125). Copyright “After Hours Animation”.

Figure 4.1: Still from The Deep (a 5 minute animation completed in 2009) and an architectural rendering of a bathroom (completed in 2010). These are examples of previous works which approach photorealism.

Figure 4.2: Still from Plant to Plant (a 5 minute animation completed in 2008) and development still from Saltram (an animated TVC completed in 2004). These are examples of previous works which approach a hand made aesthetic.

Figure 4.3: Screenshot of ZBrush being used to texture Default Whippet.

Figure 4.4: Cow Reduce animation.

Figure 4.5: a) Three viewport snapshots showing steps taken to create Wobbling Primitives. b) Wobbling Primitives 3D animation.

Figure 4.6: A composite image showing stills from Hand.

Figure 4.7: Blinding Tree animation.

Figure 4.8: a) Viewport snapshot showing how I carefully positioned joints for Default Whippet; b and c) Viewport snapshots showing how I animated Default Whippet using keyframes.

Figure 4.9: Viewport snapshot from Spline Whippet.

Figure 4.10: The pencil sketch which I referred to when modelling Library Man (Figure [libraryMan |fignum]below).

Figure 4.11: Library Man animation.

Figure 4.12: a) Detail from project diary showing working sketches for Spline Whippet and similar experiments; b) Viewport snapshot showing my attempt to animate Default Whippet using maths functions; c) Sine Whippet animation.

Figure 4.13: Demonstration video showing my Auto Expression UI in use.

Figure 4.14: The pencil sketch which inspired Woman with a Bag; Playblast and the Woman with a Bag animation.

Figure 4.15: Demonstration video showing my Auto Keyframe UI in use.

Figure 4.16: Playblast (i.e. series of viewport snapshots) of Peds Prance animation.

Figure 4.17: Flocking Whippet animation with 500 whippets.

Figure 4.18: The same Flocking Whippet animation rendered through another camera.

Figure 4.19: Force First playblast (showing particle movement without geometry attached) and rendered animation.

Figure 4.20: Viewport image showing the Modelling as Animation custom UI.
- Figure 4.21: Diagram showing a Modelling as Animation workflow.
- Figure 4.22: Conference Figure animation.
- Figure 4.23: MasA Whippet animation.
- Figure 4.24: Sketchbook page showing some sketches of Ginger.
- Figure 4.25: a) Phone Figure sketch; b) Phone Figure animation.
- Figure 4.26: Building Phone Figure animation.
- Figure 4.27: Photo showing the tools I used when Working from Life.
- Figure 4.28: Demonstration video showing Plein air Still Life tools in use.
- Figure 4.29: Diagram indicating several features of the Plein air Still Life Production Tools.
- Figure 4.30: Diagram indicating features of the Plein air Still Life Post-production Tools.
- Figure 4.31: Plant animation.
- Figure 4.32: Stills from Plant with glitched form on the left.
- Figure 4.33: Castlemaine animation excerpt.
- Figure 4.34: Playblast of Falls Creek animation.
- Figure 4.35: Boots Animation.
- Figure 4.36: Chair animation.
- Figure 4.37: Shoes animation.
- Figure 4.38: Viewport snapshot showing the many cameras created when using my Auto Camera script.
- Figure 4.39: Still from Shoes animation rendered from a different point of view.
- Figure 4.40: Books animation.
- Figure 4.41: Viewport snapshot taken while working on Shadow Whippet, discussed below.
- Figure 4.42: Shadow Play demonstration video.
- Figure 4.43: Shadow Whippet1 animation.
- Figure 4.44: Shadow Whippet2 animation.
- Figure 4.45: Photograph showing my setup and location when making Green Jumper and still from Green Jumper animation.
- Figure 4.46: Cup animation.
- Figure 4.47: Specular Whippet animation.
- Figure 4.48: A compilation of Whippet in the Sun animations.
- Figure 4.49: Whippet in the Sun stills.
- Figure 4.50: Shelves animation.
- Figure 4.51: Cafe Figures animation.
- Figure 4.52: Cafe Figures Colour Test animation.
- Figure 4.53: Cafe Figures MoBlur animation.
- Figure 4.54: Stills from Cafe Figures animation showing some of the glitched figures that appealed to me.
- Figure 4.55: Stills from Books animation (described above).
- Figure 4.56: Video demonstrating Plein air Still life version 2.
- Figure 4.57: Night Scene animation excerpt.
- Figure 4.58: Night Building animation.
- Figure 4.59: Whippet in Bed animation.
- Figure 4.60: Coloured Whippet animation.
- Figure 5.1: Screenshot illustrating how a user might “sculpt” in Zbrush.
- Figure 5.2: Viewport snapshot indicating how to create a polygon mesh by writing code instead of moving icons in the viewport. Activities such as this helped me to appreciate that a 3D mesh exists as a dataset.
- Figure 6.1: A Pixar character from.
- Figure 6.2: Still from Whippet in the Sun.
Definition of key terms

3D Studio Max (or 3Ds Max) is a 3D computer graphics program developed by Autodesk. The creation of 3D computer graphics can be broken into 3 basic areas; modelling, animation and rendering. My research spans all three of these areas and in this document I use the terms '3D computer graphics software' and '3D computer animation software' interchangeably. Most often I use terms such as '3D software', '3D programs' or '3D tools'. 3D computer graphics software uses a three-dimensional representation of geometric data. 3D computer graphics software can be used for 3D printing (i.e. to create physical objects) but it is mainly used to create 2D computer graphics. The fact that it computes 2D graphics based on a virtual 3rd dimension is what distinguishes 3D from 2D computer graphics software. The specific 3D programs used in this research include Maya (by Autodesk), Houdini (by Side Effects Software), and ZBrush (by Pixelogic).

Modelling is the process of creating a virtual form, i.e. a 3 dimensional mathematical representation of an object.

Texturing is the process of adding colour and texture to virtual models.

Rigging is the process of creating a digital skeleton in order to deform a virtual model.

Photoshop is an image editing program developed by Adobe.

A mesh (or polygon mesh) is a collection of vertices (or points) and faces (or polygons) that describes a 3D model. Throughout this document I use the terms '3D model', 'virtual object' and 'mesh' interchangeably.

A texture map is a 2D digital image used to colour a 3D model.

Skinning is the process of binding vertices of a mesh (or model) to its internal skeleton so that they move in accordance with the rotation of joints.

In 3D animation 'the centre of the world' refers to the origin of the Cartesian grid, i.e. the co-ordinate (0,0,0).

ZBrush is a 3D sculpting tool developed by Pixelogic.

Pixar, a subsidiary of the Walt Disney Company, is an American 3D animation studio.

Google Maps is an internet-based mapping service developed by Google.

Second Life is an online virtual world developed by Linden Lab and launched in 2003.

Paint Effects is a component of Maya - a 3D animation program developed by Autodesk. Using Paint Effects to create trees for a 3D animation involves choosing from a library of available designs and changing parameters such as the number of branches, leaves etc.

In Maya we can use Fluid Effects tools to create smoke and fire.

In Maya we can use Fluid Effects tools to create smoke and fire.

We can use Fluid Effects tools or Maya's specially designed Ocean Shader to create ocean waves.

A shader is a set of algorithms that tells a computer program how to render the surface properties of a 3D model.

3D software users can choose from a variety of virtual lights which have been designed to replicate the effects of real world lights. These typically include spotlights, point lights, directional lights and area lights.

A lighting model (also known as reflection model or illumination model) is a model of interaction between light sources and objects (Guha, 2015, p. 425).

Subsurface scattering is a shading and rendering technique which simulates the look of translucent objects such as skin, milk or wax.

Global illumination is a lighting and rendering technique designed to simulate indirect lighting effects such as light bouncing and colour bleeding.

Maya’s Script Editor lets you type and execute code and it also allows you to see code that has been executed when you use standard interface tools.

Using Maya, it's easy to create a custom shelf button which executes a block of code written by the user.

This tool automatically reduces the number of polygons in a mesh. The user specifies a percentage and the tool automatically reduces the overall polygon count.

In computer graphics a spline is another name for a curve.

In Maya an expression is a mathematical function or small short block of code that is written by the user. Using expressions we can control keyable attribute over time.

Utility nodes (such as Blend, Condition, Multiply Divide and Sampler Info) are often used when setting up shading networks and also when rigging. Plein air Still Life v2 uses these nodes to make connections between object attributes.
Appendices

Contained in the following pages, appendices to this document include;

Appendix A – sample pages from my online project diary
Appendix B – Python code
Appendix C – Research outcomes presented in the form of a list
My conclusion on completion of this experiment is that the glitches annoy me… they are “glitches” not indications of “emergent content”. They haven't yet suggested to me meaning/ a way of proceeding… they are rather deviations from a correct run cycle.…

HOW TO REMEDY THIS?
- maybe start with models and animations (ie movements) that are less standard/well-known/ cliched. Ie start with dog forms (for example) and dog movements that are not so well recognised; such as impressionistic rendition of a dog and maybe a movement such as fighting…

  eg two dog forms could interact like in the table tennis film, “Dropp”
  .I am wondering if it would look better by combining early whippet experiments that I did with surfaces built from curves…(see image left from “Whippet Sketch” experiment???

QUESTIONS for Shaun;

INHERITING FROM QWIDGET or QMAINWINDOW
Q. when I change my code to inherit from QMainWindow instead of QWidget my layout gets stuffed up. Why is that? and how do I fix it?
A. ---- inheriting from QMainWindow VS QWidget;
appearently with QMainWindow, you need to add a CentralWidget before adding GUI elements.
Q. on what occasions do you need to import sip, and/or QtCore? I dont seem to need these for my scripts.

................................

**PACKAGES AND MODULES**

Q. how to organise files/scripts?

Once the scripts get longer is it advantageous to organise them as separate documents so that one script file calls other modules?

A. difference between a module and a package;

A **module** is a single file (eg. import myModule)

A **package** is a collection of modules in directories that give a package hierarchy

eg. `from my_package.timing.danger.internets import function_of_love`

Documentation for modules

Introduction to packages [http://www.network-theory.co.uk/docs/pytut/Packages.html](http://www.network-theory.co.uk/docs/pytut/Packages.html)

Importing a package Python searches the **sys.path**…

SYS.PATH ⇒ is Initialized from the environment variable PYTHONPATH, plus an installation-dependent default
...heres a link where the guy talks about what to do if Python is not finding modules;

To add a path to the sys.path ⇒

```python
import sys
sys.path.append("/path/to/Modules")
print sys.path
```

I did this and added; 'C:\GINA\RESEARCH_laptop\PROJECT\SCRIPTS\Modules'

Now in my **SCRIPTS** folder I have 'dev' and 'Modules'. dev will be all the development scripts and Modules will be all the completed scripts. as a script is superceded I will replace it and put the old one with date of supercention in the name.

PROBLEM importing --- I have a simple script that imports a file Modules\testProject\test.py'...
when I press F2 and send the script to Maya I get the error; 'no module named testProject'
when I press ctrl+b to execute in Sublime I get the error; "No module named maya.cmds"
is it a matter of pointing sublime to maya.cmds ? and pointing maya to my module in scripts path (ie. Maya Python path)?
A. -- from Autodesk docs;

**Adding items to your Python path**

To add items to your path in Python, do one of the following:

1. Set PYTHONPATH in your Maya.env file, or in your environment before you run Maya
2. Append to sys.path in your userSetup.py or other script once Maya is running.

Here is an example of appending sys.path

```python
import sys sys.path.append('~/Users/jd/we/maya/Scripts')
```

So all I need to do is to run the same code that I ran in Sublime in Maya (I suspect this is the same as adding the path manually to the environment variables). I guess sys.path is relational to the software I run the sys.path.append from...

NB... Gina working with these imported files seems to be confusing because it seems to reference older files which makes it hard to test the changes I make. I keep having to rename the file I’m importing to truly test the new functionality. very confusing.

I think it is working as I expect, I just have to be careful with the way that it imports modules and stores them in...
memory.

__init__.py file can be empty or can contain initialisation code

IF __MAIN__()
maybe a related question… what does this mean?


19_03_14

Amendments and Additions;
be good to have a button “select driver” and “select target”

Limitations;
if I build one node off another (so a driven becomes a driver) then I need to manually set the keys one at a time because …. or gina try adding the rotZ to the cog...?? Actually it seems to work fine!! Seems I just needed the COG rotZ value. I guess Maya must have a record of the key number perhaps? I’m not sure….but it seems that the keys are laid in the correct order (ie they build upon each other).
--- actually I think that was a fluke…. the results are erratic when I daisyChain the driver nodes. will try now to link all to one node…. might make that node an “external” one… ie. a piece of geometry or a group….
The way that its set up the driver is something with values and keyframes… What does this means??
that it cant be a noise texture or something….
that driver could be a ramp
driver could be an existing animation (ie something conceived of separately or a “found object”.
Im thinking how relevant this enquiry is to “Dropp” animation.

Evaluation;
I am enjoying connecting all to one central (and external) node. However the issue of adding extra keys and using time offset (as a fractional number less than 1) is confusing I guess because when I add another key it changes the “get next key” function and all the keys get shuffled around.
-- I just realised that I could add keys to the original driver node… or I could have a type of two tiered driver system...

Thoughts;
wondering how to transport the keys onto another rig (ie other nodes).
- open text files and change one node name for another…. I could make a little helper function to open files and change the text…
  - “autoAnimWhippet_008.ma” layers controls…. is it going to be more interesting to start with complexity????

18_03_14
WAS going well but seems to be a problem setting keys from default files….

Python Strings-
This link http://stackoverflow.com/questions/1228299/change-one-character-in-a-string-in-python
says work with strings as lists until you have no choice but to turn them into strings

glitches to fix ⇒ done
"apply default values"; gina I needed to type .txt at the end…

things to addr ⇒ done
duplicate tab
fill in fileName and folderName for save defaults...
save all values

TESTING...
“GINGER” file has no expressions on neck...
testing going well… fun…. but I can’t seem to get the right timing for the COG.ty....

THOUGHTS…..
I’m still not sure how to save iterations of the animation. Some ideas;…
- save the file iteratively as I work on it and then combine them somehow...

.......  
17_03_14
AutoAnimDev_041.py ⇒ is looking good :))
To do today;
- use tangent in and out
- change Ref keys to correspond with keys
- make tangent in and out combo boxes
how to set the comboBox as it was saved?? do I use setCurrentIndex()? A. seems I have to use findText() first. this returns an index, then I use setCurrentIndex().
fill in the defaults file name and folder name for tabs from files
- add the anim expressions tab page to main window (should this be in a different file?)
-- make new tab the current one?
It uses setCurrentWidget().... [not sure if I need currentIndex()] 

-- could inherit from QMainWindow and add help menus
--?? I could try putting the utils in a separate file…
- and putting AutoAnimPage in a separate file....
--?? check out Maya script jobs
?? error checking

QUESTIONS
I’m just wondering what happens when you set one node against another…. (ie not all referring back to one central node)... I’m thinking what happens might be glitches….

TESTING
glitches to fix ⇒
--- “apply default values”; gina I needed to type .‘txt’ at the end…
things to add / consider ⇒
--- duplicate tab (do so without target)

.......  
14_03_14
TO FIX
Ref Keys to correspond with key numbers
for “save defaults”; only add “\" if there is a folder entered

HOW to have the text in a line edit so that it is just placeholder and disappears as soon as user clicks in the box??
A. ? setPlaceholderText
CREATE TABS FROM FILES has to be a separate class or function because it goes in the main window.
SAVE and APPLY DEFAULTS is on each tab.
can I make a separate class that saves default files and reads default files?
How would it “plug in” to my UI? how would I connect it to buttons?
For the moment it seems much easier to do it all in one file....
maybe later I will try to separate out the utility functions into another file (including Qt utils)

Today I am wondering how I can organise my scripts so that they call each other.
Perhaps I will test this as I implement the next phase of my AutoAnim script which is to write and read files
(default value files).

to do;
recap functionality of the script;
what can be inputs so that the code is reusable (eg; filePath)
do simple test calling a script, making an instance of a class and calling a function on that class.

Seems that my animation tools could be designed as a package…

It would be amazing if I could have the code reusable so that I could use PyQt or maya.cmds to build the UI... but at this stage I think this is too much to ask.

A TENTATIVE STRUCTURE;
AnimationTools/ top level package
__init__.py initialise the AnimationTools package
AutoKeyframe/ subpackages for automatic keyframe tools
__init__.py
saveLoadDefaults.py
AnimExpressions/ subpackages for animation expression tools
__init__.py
AnimExpressionsTab.py
AnimToolsUI/
__init__.py
AnimToolsMainUI.py

ModellingTools/ top level package
__init__.py initialise the ModellingTools package

TexturingTools/ top level package
__init__.py initialise the TexturingTools package

RiggingTools/ top level package
__init__.py initialise the RiggingTools package

Utilities/ top level package
__init__.py initialise the Utilities package
userInterface.py (user interface templates; dynamic tab setup etc)

13_03_14
Trying to decipher what my keyframe code does…
wondering about tools and strategies for designing software…
thinking about going back to square one…
createNewDriverKeyList seems to be working. Now test `calculateKeyTime`;
and `calculateNextKeyTime`  

```
setTargetKeys()
```

need to incorporate;
- valueOffset
- inTangent
- outTangent

“dynamicUItest_031.py” - seems to be working …

```
NOW;
```

- apply default values
- save default values
- create tabs from files

GINA it would be very cool if I could do these using classes or even separate files (packages? modules?). Basically if I could do them in a way that makes it easy to keep track of them and also to reuse them.

- Fix interface issues;
  - where do I use “parent = self”

Make interface collapse when group boxes are unchecked

Driver and Target cycle length QHBoxlayout jumps around (up and down).

Maybe add help docs??

A. it seems that I cant add a menu bar. Perhaps its because I am inheriting from QWidget and not QMainWindow.

- eg the file path where default values are stored
- the fact that time offset is entered as a fraction (greater than 0 and less than 1)
- Tab Names -- if a node and channel are selected the new tab (or the new tab name for current tab) will be named as follows, “node.channel”. If nothing is selected the tab name will be based on user input. If there is no user input and nothing selected the tab will be named “tabn” (where n = the number of tabs).

Maybe add error messages??

- eg when timeOffset is greater than 0 or less than 1

DEFAULTS --
- path I use to store defaults;
- projectDirectory/data/autoAnim…

default text files must be collected into a folder in order to use them to create tabs

```
12_03_14
```

COMBINE `keyframeTest_009` with `DynamicUItest_028`.

HOW does `keyframeTest_009` work?

At the moment I have written it as a class. Should it be written as functions rather than a class? How do I incorporate it as a class?

hard coded `driver` and `target` (node and channel),

and also `driverKeyLoopLen` and `targetKeyLoopLen`…

`makeKeyDataList` defines a list of the multiplier and key attributes etc. Replace this with accessors that retrieve
user input (6 values per key).
the list will contain as many lists as there are keys on the driver ...
each list (ie each index) in the keydataList has 6 members, ie [Ref keys, TimeOffset, ValueMult, ValueOffset, inTangent, outTangent],

getRefKey (new name for getDriveIndexNum) .this function gets the driver key (ie it gets the driver keys position in the driver loop).
It returns a list of driver keys. Ie for each key in the targetkeyLoop it returns a list of which driver key to refer to

g getTimeOffset and getValueMult are both similar.
Should I combine these functions so that they return all 6? or make one standard function and you enter the index?

createNewDriverKeyList =
gets a list of all the keys on the driver node
splits the list into lists of length of the driverKeyLoopLen
calculates number of keys to lay (on the target node)
returns a list of the driver keys, e.g. it might be something like [0,1,2,0,1,2, etc]
...
try to combine using AutoKey() class....
I’m thinking that it may be better not to use a class because I need to recalculate the key data list each time...

DICTIONARY
I’m thinking of creating a dictionary instead of a list...actually I don’t think there’s a lot of saving in that.... I think I will continue with lists for the moment ....
I could have done as follows ;
ZIP two lists together to create a dictionary..........
keyList = list of keys
dataLists = list of data lists
D = dict(zip(keyList, dataLists))

...calculateKeyTime (timeOffset, keyIndex)

....
11_03_14
FILE BROWSER
QFileDialog.getExistingDirectory
..........
      file = str(QFileDialog.getExistingDirectory(self, "Select Directory"))
......
Gina; How to change the ViewMode? at the moment it takes up too much room. I just want a simple file browser...actually I just want an icon that launches a file browser to help choose a file path.

GOAL; to finish this script before next Monday (when I might be working at XYZ).
then to package as .exe so that user doesn’t need pyQt (or explore using “PySide”?)

changing my group boxes to use my “createGroupBox” function. Problem I’m having is that the bottom one (tab
control) collapses the window but the higher ones don't. I will push on and come back to this...

........

09_03_14
READ THIS BOOK

HOW TO EXPAND GROUP BOX?
found this;
I have a checkable
QGroupBox, whose sole child is a QFrame with no borders. All the
widgets I put into the QFrame.
Then I connect the **toggled(bool)** signal of the group box to the frame's **setShown(bool)** slot. That way, the
QFrame is hidden when the QGroupBox is un-checked. And you can do it
all from the Qt Designer: less hand-written code, yay!

HOW TO MAKE QFRAME a child of QGroupBox?
“Composite Widgets
When a widget is used as a container to group a number of child widgets, it is known as a composite widget. These can be
created by constructing a widget with the required visual properties - a **QFrame**, for example - and adding child widgets to it,
usually managed by a layout. The above diagram shows such a composite widget that was created using **Qt Designer**.
Composite widgets can also be created by subclassing a standard widget, such as QWidget or **QFrame**, and adding the
necessary layout and child widgets in the constructor of the subclass. Many of the **examples provided with Qt** use this approach,
and it is also covered in the **Qt Tutorials**.

LAYOUT MANAGEMENT
I can get the frame to show and hide the widgets but the group box doesn't expand and shrink.

Is this link useful?  http://qt-project.org/doc/qt-4.8/layout.html

QSizePolicy  Layout attribute describing horizontal and vertical resizing
y  policy

ADDING SQUASH and STRETCH?
addStretch(1)  ??

??........Remove additional spacing

In order to remove the additional spacing while using QBoxLayout we can set
the margins as

layout->setContentsMargins(QMargins(0,0,0,0))

........when I make the window smaller (using setGeometry) the group box shrinks so I'm wondering if I set group
box size??
How to add widgets to a frame?

```python
frame.setLayout(hBoxLayout)
frame.setWindowTitle("QFrame Add Layout and Widgets")
frame.show()
```

Seems you add widgets to a layout the set the layout attached to the frame. also seems that you have to show the frame

HOW TO CONNECT A SIGNAL FROM A GROUP BOX???
-- the two signals are - toggled and checked.
I use toggled

IT WORKS to directly connect the GB.toggled to the Frame.setShown. However I am not immediately sure how to shrink the size of the GB when frame is not shown.

SIZE POLICY

```python
QWidget.setSizePolicy(QGui.QSizePolicy.Minimum, QGui.QSizePolicy.Minimum)
```
or

```python
QWidget::resize(rect().width(),100);
```

LOOPING OVER WIDGETS

how to determine if its a widget or a layout?

```python
if type(item) == QGui.QWidgetItem
```

...this seems to return the type ie QLineEdit etc. can also use isinstance as follows;

```python
if isinstance(item, QLineEdit):
```

```
08_03_14
HOW TO CHANGE COLOUR OF A BUTTON?
testbutton.setStyleSheet('QPushButton {color: blue}')
```

```
07_03_14
--------just ordered this book Practical Maya Programming with Python
order: Order: PAC-14-1971486-1380339
https://www.packtpub.com/account
my password is-- research
```

HOW TO COLLAPSE A GROUP BOX??
To use a group Box;
Make widgets
make layout
addwidgets to layout
make group box and use groupBox.setLayout(layoutname)

….having trouble making it collapsable; is it a matter of parenting?

```python
QWidget.setlayout() to install the QVBoxLayout object onto the widget. At that point, the widgets in the layout are reparented to have window as their parent.
```
You can align the items to the top. Also you can play with making your
>> expanded widgets size policy set to minimumExpanding
       size hint
       paintEvent

all working well. Now I just want to bring my tabs test and my dynamic UI together. I might actually make it with all
the same functionality as my maya.cmds UI.

----- rename tab
-- maybe try collapsable group boxes; could have one for default value ands, one for key table.
- maybe file/folder browser for default value files
- eventually I would like to use scriptJobs to save iterations of the animation

INHERIT FROM QWIDGET??
Q. I will start by working out why my window doesn't show if I inherit from QWidget instead of QDialog…
A. I now have it simple and inheriting from QWidget.

06_03_14
from yesterday; “dynamicUItest_008.py” seems to be working…now I just have to work out how to add tabs and how to access each text
box.

so I am now trying to;
access the layouts in a list
access the widgets in a layout
   this might use children(), or count()

maybe I need to use a “groupBox” widget? it seems that this is an “organiser widget”
so instead of each column being a Layout, it could be a group box. Would this help me to access the widgets in
the group box?

from http://stackoverflow.com/questions/1781173/pyqt-removechild-addchild-qgroupbox
“I don't think calling removeWidget is necessary. Try just calling widget.deleteLater on whatever you want to delete. Then when you want
to add it back, recreate it and use layout.insertWidget to put it in its proper place. Does that work?”

DELETING THE GROUP BOX
--- changing my create keys columnLayout to creating a group box breaks the way I was deleting the layout ---
guess I need to work out how to delete the widget ---
Gina it looks like I can use deleteLater on a groupBox………..
widget.group_box.deleteLater()
widget.group_box = None

GETTING THE KEY VALUES
looks like this lists all the widgets in the group box (NB. the first one is the QVBoxLayout); GroupBox.children()
… wondering about this documentation; does it mean I could use “findChild” and enter the name of the child
(which would be the name of the keyAttribute) ---
QObject findChild (self, type type, QString name = QString())
**dynamicUItest_011.py** is working well.
Now I want to work out how to;

----------delete groupBox widgets from keys layout (when columns need to be deleted). **dynamicUItest_014.py** does this

use tabs so that each tab returns the correct data (as per my last example using maya.cmds)
Create an “update all” button

---

**USING TAB WIDGET**

from: [http://qt-project.org/doc/qt-4.8/qtabwidget.html#details](http://qt-project.org/doc/qt-4.8/qtabwidget.html#details)

The normal way to use QTabWidget is to do the following:

1. Create a QTabWidget.
2. Create a QWidget for each of the pages in the tab dialog, but do not specify parent widgets for them.
3. Insert child widgets into the page widget, using layouts to position them as normal.
4. Call addTab() or insertTab() to put the page widgets into the tab widget, giving each tab a suitable label with an optional keyboard shortcut.

A tab and its associated page can be removed with removeTab().

---

Below is the simplest tab test;

```python
class MyTabWindow(qg.QWidget):
    def __init__(self):
        qg.QWidget.__init__(self)
        # set geometry
        self.setGeometry(250, 150, 400, 300)
        # set window title
        self.setWindowTitle('Tab Test')

        tabWidget = qg.QTabWidget()
        tab1 = qg.QTabWidget()
        tab2 = qg.QTabWidget()

        tabWidget.addTab(tab1, 'tab1')
        tabWidget.addTab(tab2, 'tab2')

        # layout manager
        vBox = qg.QVBoxLayout()
        vBox.addWidget(tabWidget)
        self.setLayout(vBox)

window = MyTabWindow()
window.show()
```

**tabTest_03.py** seems good

---

05_03_14
from yesterday;

I SEEM TO HAVE found a workaround for the QLineEdit SIGNAL which is to use “returnPressed” as in the following format,
self.inText.returnPressed.connect(self.pushedButton)
this reference was helpful; http://pyqt.sourceforge.net/Docs/PyQt4/qlineedit.html

to continue...i have changed from using textChanged to returnPressed and it seems to work. I’m now thinking that I start with two keys in loop (ie. 2 key rows) and then add key rows….or do I just rebuild?

HOW to add command with arguments in PyQt?
a maya button command is something like this;
command=(partial(self.populateTextField, self.textField_Driver)
def populateTextField(self, textField, *args):

HOW to add an item to the layout and maybe delete an item from the layout? useful on deleting -- http://stackoverflow.com/questions/4528347/clear-all-widgets-in-a-layout-in-pyqt
You can use the close() method of widget:
for i in range(layout.count()): layout.itemAt(i).widget().close()

Or I could use a grid layout and use the addWidget() method described here;
http://zetcode.com/gui/pyqt4/layoutmanagement/

"layout.count()" looks useful… and "layout.takeAt(index)"
GINA be aware that removing a widget from a layout does not delete it, it just isn't visible cause its not laid out… You might need to use widget.setParent(None)

this looks useful; http://stackoverflow.com/questions/9374063/pyqt4-remove-widgets-and-layout-as-well
seems to give two solutions; this WORKS;

def clearLayout(self, layout):
    while layout.count():
        item = layout.takeAt(0)
        widget = item.widget()
        if widget is not None:
            widget.deleteLater()
        else:
            self.clearLayout(item.layout())

YAY!!!; “dynamicUItest_008.py” seems to be working
now I just have to work out how to add tabs and how to access each text box

........

doing some study on the API from AU Masterclasses. Link provided by Gerardo.
http://area.autodesk.com/masterclasses/masterclass/class2_q3_2012_kristine_middlemiss
My notes from re this video lecture are here;
I got this far...but its not happening dynamically... will go back to the tute below… maybe just get something to update when No of keys is changed………

"Every widget emits signals when it's updated in some way. We need to connect to those signals so that we know when the user has changed the name or shape type. Let's use the connect method to do just that.

The arguments passed to connect are the following (in order):

- The widget emitting the signal.
- The signal being emitted. Each widget is capable of emitting many different signals so we need to specify the one that gets emitted when the condition we're interested in occurs. For instance, the signal for nameLE is "textChanged", which as the name suggests is emitted whenever the text gets changed.
- The function/method that gets called every time the signal is emitted. In this case, we're providing our updateDescription method.

Got a version to update one lineEdit when the other is edited ---- "dynamicUItest_temp.py" (in week08 folder) --now add widgets for each number “dynamicUItest_temp_02.py”

I will now try to implement this in “dynamicUItest_003.py”

-------------

PROBLEM == QLineEdit SIGNAL crashes Maya 2013-------

I did a quick Google search and found this;

For windows users, PyQt 4.9.1 built against Qt4.7.1 and SIP4.13.2 for Maya2013 x64, can be downloaded here.

Links to the respective sources are included below.

Update: Have updated an executable to make the installation easier. You may download it here.

Qt 4.7.1 source
SIP source
PyQt source

from here;

http://codecg.com/2012/04/15/pyqt4-for-maya2013-on-windows-7-x64/

To clarify what I have installed;

Python 2.6.4
Maya 2013
etc

I SEEM TO HAVE found a workaround for the QLineEdit SIGNAL which is to use “returnPressed" as in the following format,

self.inText.returnPressed.connect(self.pushedButton)

this reference was helpful;

http://pyqt.sourceforge.net/Docs/PyQt4/qlineedit.html

LATER I need to separate my code into __build__ functions etc as Shaun has it.
seems I am pursuing auto anim tools on a few levels; rebuilding the window (tabs etc) with **PyQt** - this might apparently give me more options for interactively or procedurally generating the UI (eg. I might be able to get the number of keys in the loop to generate a certain number of input boxes.

    working out how to iterate over all the keys
    working out how to get the tabs working properly with Maya cmds - I seem to have pretty much worked this out. Although I could use the Maya API to get the long name of the autoAnimTab instance…

    - optional is to try using **XML** for reading and writing presets - this would probably involve writing (or using) the read XML class that Shaun wrote in the intermediate course.

**GENERAL POINTS…**

I have been having trouble getting Sublime Text setup so that it will auto complete. I think I have auto complete now working for Python code. I also have Qt able to execute directly from sublime. Also have sendToMaya working and cntr1+B executes python code. I think I might leave it at that because I keep breaking things when I try for more functionality. But ideally I would like;

    autocomplete for maya cmds (strangely I had this for a short while then it stopped working)
    autocomplete for PyQt

**TO DO…**

make PyQt UI in designer
    save file for Shaun so he can recreate it
    try recreating it by writing Python code - this will entail writing “widgets” so that it can procedurally generate boxes etc.

Figure Sketches Development Notes commenced 28.04.14

....

GENERAL THOUGHTS ON THIS PROJECT;
Ideas for script
how to make geo within which event occurs such as move geo or merge verts?
- use bounding box
- use falloff a la “get closest point” script

Ideas for footage made of a string of sketches
A string of POVs… each camera angle indicates a character ...eg. view from a bike. view of a person on a bike…
As each sketch (or vignette) is based on my sketches I guess it really does indicate a POV (ie my POV).

DIARY OF PROGRESS; ....
28_04_14

POSSIBLE USEFUL TOOLS
- parent (move position) and unparent (presevere position). THis allows me to position and align a poly primitive to a joint.
- tool to connect attrs such as polySplitRing5 > Split Type to a controller…. maybe to sliders that scroll through changing various attrs?
- perhaps the ability to delete some history and add other history to my Anim Potential

TEXTURING THOUGHTS / NOTES
texture as I model…
-- there’s a big difference between surface shader and Lambert… maybe mix the two...

STEPS-
DOING and DONE
- using Rapid Rig Advanced. NO toon legs and arms (hoping it doesnt matter too much)...
- block in form with pieces of geometry (polygon cubes eg). do this quickly and roughly…
- Create “unintentional forms...by playing with history and “creation parameters” of polygon primitives
eg polySplitRing5 > Split Type = Mult… then change “Divisions”.
Perhaps as I work I could rig these attrs up to a controller?

TO TRY
- I could try making curves based on the sketch and create sufaces based on these curves.
The topology of such surfaces is intrinsically more interesting it seems…
- I could be recording the process (eg. duplicting the models every X seconds or X mouse clicks).
-- could the equivalent of the pencil strokes (ie that texture) come from (be contributed by) the mesh edges?
- think about which parts move! ie. maybe its his hands that are fiddling
OBSERVATIONS

- changing parameters at the beginning of the tree makes a big change to the form of the figure (eg changing the number of divisions on a fundamental polyCube). I guess if there were initially more divisions on the cube to start with the change in form wouldn't be so radical....

- my compulsion is to join all the polys together. should I indulge this compulsion or resist??

- M as A recorded my modelling process (albeit by pressing a button). This one uses the modelling process by accessing history and animating parameters of that history.

- interesting to experiment with combinations of history... eg animate the translate Z (local axis) of an extrude node and also change the number of divisions on the original cube. The face that moves changes and the axis of movement also changes.

-- maybe it needs to be random on each channel so that the hand movement is never the same...

29_04_14

I was thinking last night that it might be good to keep the models separate (not acquies to my desire to “combine” them. My reasoning was that I could manually add more detail rather than smooth the model; I thought that by snapping verts together (rather than merging) and then animating history the models would act as one.... I have since realised that the snapped verts are unlikely to stay together...

There are many options when it comes to animating the PolySplitRing node (eg. insert edge loop).

I wonder if you can animate the profile curve shape?

DELETING HISTORY SELECTIVELY

I don't think this is possible..
I can use this cmd to delete all history `mc.DeleteHistory()`; but can't find any flags for this command.

30_04_14

SCRIPT DEV
Try not to get absorbed in making tools for their own sake… come back to making anims… it doesn't matter if I use existing tools..
- a pro of using a script might be to make it easy to deviate from an existing attr value (such as `polyExtrude.localRotateX = -25.2108`), so the script would establish a NORM or a BASE attr value from which to deviate.
-- be excellent to have a script to create either expression or node setup that allows for current value (eg in `polySplit.edge[0]` and multiplies that by the control)
-- could I use the `anim expressions` to change edges etc?

Quotes from my sketchbook;
“Start with observation and work with suggestions from the medium” At the moment I feel that this statement sums up my project impetus.
“A leg can be shapely but not entirely deliberate[ly described]”
“Make in Maya what you see- this might be a shadow [rather than the object that casts it]” On this point, I found the problem is how to end geo… what to do with the boundaries? Maybe this is where viewport BG comes in. Maybe start on mid grey. Or try using two colours like I do when painting (eg ultramarine blue and burnt sienna). Think about these forms much more as 2D colour compositions..

IDEAS
-- someone on a bike... off a bike... pushing a bike
  when pushing the bike it is something that obscures vision of the pusher’s lower body
  when riding, the bike choreographs the rider’s body. could the wheels be indicated by changing the texture of the background?

GEO and TEXTURE
I'm thinking that these experiments could be developed through experiments with form and texture together.

THOUGHTS
WORKING WITH HISTORY- one aim of this project is to see whether its possible to become attuned to working with deformer/ object history.. to model (as well as texture and animate) in a way that allows input from teh software (but not “clicheed input- need to expand on what I mean by this).
ARM project - could make the same arm many times in different ways... using different tools. presumably each different way of making would present different animation opportunities... different specific arm characteristics might emerge.
- as I model I hook up attrs to a controller and slide through the cntrl to see results of combinations of attr changes. (different to scrolling through one attr change at a time).

** This could also be a place to use my auto keyframe script -- this script is designed for looped animation but it could also be used for many types of driven animation I think --- its now based on the index of driver loop: could it be amended to be based on value of driver key? or would this just be assuming a driver loop of 0 ?
-- polyExtrudeFace.keepfacesTogether == interesting for hands and fingers
-- polyExtrudeFace.localTranslate X,Y,Z == interesting for finger movement -- how to set this up so that the axes move in relation to one another?

--- can I rename history nodes? eg can I name polySplitRing to “sleeve”

QUESTIONS
Merg verts / distance --- how to alter the verts involved? ie I suppose the results would be like the SOuP plugins.
GINA - maybe use something similar to “closest point” script. this might entail a custom node

01_05_14

DISCOVERIES - TECHNICAL
-- I can rename nodes (eg polSplit13)
  To create an expression offset relation I could have a naming convention for each “child” geoChange node (eg. armPulse_001).
  could I use my renamer to rename these nodes?
  working with sublime text open and having a script file (or several) open while I would might be just as valid and useful as trying to package my code for multipurpose use in the form of UIs etc.

--- The point of this project is to find ways not to start with “a blank canvas”, but rather to start with something and tweak that thing...eg I might animate the rig so that the “pointing finger” below looks good/convincing.

QUESTIONS
- what will happen when the arm is rigged and animated…
- can an interesting (narrative) animation emerge through play with the material qualities of the software?

NEXT STEP
- thinking now to add rig. - add head geo. - add some anim to arm
- do another model with hand to chin --- how to transition from this hand model to the next??
I really like this one because its reminiscent of a hand... a kind of impossible hand. maybe frozen but perceived moving quickly
⇒ armSpasm = 2.714936

Its not nearly so interesting when rendered low resolution; its not very hand-like.

⇒ “extrudeFingers1.keefacesTogether” = Off

Maybe the point is not to animate transitions and trembling but to allow the tweaks to suggest other forms with which to work. Ie. what would it be like to rig this hand as it seems to be positions now?
Notice how tweaks done after the polysmooth node rupture the form when smooth reactivated...

Thinking about how to represent this pic... thinking that it could be one model and some verts are merged... as arm moves verts unmerge...

QUESTIONS
--can the areas where the sketch “peters out” (eg the back of this guys head) be adequately represented with low resolution geo (ie. low poly)?

02_05_14

How to marry these two images together? the left hand looks better smoothed and the right one and head look better unsmoothed I think... Maybe do one smoothed version and one unsmoothed...

TODAY
Aim is to create something watchable (or a short series of watchable movs) from what I have done so far
- do short anim utilising both arms
- try with combined and uncombined head/and
- render with various animated transitions/geo deformations
- render the anim with the form in different states.
- try different lighting and texturing options (...mabe a textured BG in comp or colour as light rendition)
- maybe try “big picture” structural elements... eg geo or sound... sound could deform geo or animated BG element
- could add fragments of other people... eg could duplicate this guy and render him in different states, changing the anim on each character to suit his new geo

1. is quite “straight” ⇒ animate and then add detail to head and RT arm...
   -- just thought that he can look at his watch... watch could appear and disappear... (maybe Mas A)
   animation is taking too long and its feeling like a fairly “normal” workflow (ie a little tedious). Difference is
   that with partial models I think you “get away with more” ... ie, I dont think the animation has to be so good.
   I will now time limit myself to 30mins for animation (have already spent about 1 hour).

06_05_14
About Using System events (ie conditions and events) and Script jobs;

MAYA SCRIPT JOBS to record the modelling process... would it be interesting to model something moving? like
sketching something moving? I have found that with sketching the results are sometimes more interesting when
the form (often a human or animall figure or a moving object like a car etc) is moving rather than sitting stagnant in
front of you (like a “still life”). I’m not sure why this is... But I wonder how this idea might translate into 3D
modelling (and maybe texturing) practice...
I intend to explore SCRIPT JOBS to find ways of recording the process of creating a model (off the top of my head
it seems that playing through a “behind the scenes” record of a modelling process might reveal a narrative or
character that could be elaborated by the 3D artist [me]).

working out the syntax for using mc.scriptJob()...the following work;
mc.scriptJob(event=['SelectionChanged', 'mc.delete()'])
mc.scriptJob(conditionTrue=['timeChangeTemp', 'mc.polySphere()'], killWithScene=True)
to kill the job; mc.scriptJob(kill=myJob) OR mc.scriptJob(killAll=True)
to kill with UI element use -parent flag
I could use in conjunction with mc.condition() ⇒ This command creates a new named condition object
whose true/false value is calculated by running a mel script. This new condition can then be used for
dimming, or controlling other scripts, or whatever.
also mc.isTrue()... eg could use mc.isTrue('SomethingSelected')
to make something happen after a
certain number of selections..
BUT how to base the activity on system TIME
http://books.google.com.au/books?id=Bd-HKkbv8CsC&pg=PA10&lpg=PA10&dq=maya+scriptJob+after+elapsed+time&source=bl&ots=eBQr-CuLeX&sig=_G6Gftc_YXussBr4h6XIF3NcEw&hl=en&sa=X&ei=3ipoU_q5Ds7OkwX-o4CwDw&ved=0CCgQ6AEwAA#v=onepage&q=maya%20scriptJob%20after%20elapsed%20time&f=false
This guy says that timerX is like an internal stopwatch...
he suggests;
startTime = mc.timerX()
totalTime = mc.timerX(startTime=startTime)
maybe I could use elapsed time and selection in conjuction? Or maybe start time is the start of the script Job...
Found this autosave mel script on creative crash here;
http://www.creativecrash.com/maya/script/3867/download_page

I will try rebuilding in Python to learn some stuff. also found these comments re this script;
“people use jobScript to set a RecentCommandChanged or SelectionChanged events, and check their timer on these events.”
same forum is all about a guys external timeLapse script; he also has a youtube clip;
he uses the MEL match command. an example of what this does (from the maya docs);
match "this" "this is a test";
// Result: this
He also uses CALLBACK… in line 258 he uses commandName + “Callback”
I dont know if I need this… will see… The Callback function seems to be his own function so it may not be needed...

----------REBUILDING AUTOSAVE SCRIPT----
he uses also mc.optionVar() which “allows you to set and query variables which are persistent between different invocations of Maya. These variables are stored as part of the preferences”

07_05_14
Still working on autosave stuff…
wondering how to pass an argument with a function to Maya scriptJobs cmd…
self.myJob = mc.scriptJob(conditionTrue=['autoKeyframeState', self.checkTimer(10)], parent=self.mainColumn)
The above works but only without the 10.
---
Looking at his autosave script I think that he might use existing script Jobs to check the autosave… does that make sense? -- not sure about this
---
MAKE THE INTERVALS SLIDER ANIMATABLE?
- I’m using an intSliderGrp to set the autosave intervals. at teh moment it updates immediately when I adjust teh slider. Can I set keys or an expression on the slider (so that the saving intervals change throughout the work period)?

08_05_14
Today will do a simple AutoProcedure script and example project (each developed in conjunction with the other).
I will start by modelling a simple object in front of me (I may texture it too...but I wont duplicate the shading network at this stage,, I would prefer to apply the final shading network to all the models).
To start with the script will;
    duplicate the model
    put it in a group
    name it based on its timeStamp
    put visibility keyframes based on its timeStamp

My UI now provides a place to type in the procedure. This is cool cause use can change the procedure that is done throughout the session....
eval() seems to work
.isdigit() is handy ....

r = "456results string789"
s = ".join(x for x in r if x.isdigit())
IDEAS for script functionality and use; encountered while working;
- history play had its limitations because once I add some more modelling nodes previous ones become
unuseable… By duplicating models perhaps I can then hook up all nodes of a similar type to one controller (for
example I could hook up all “polyExtrudeFace.divisions” to one control and all “mergeVerices.tolerance” to
another… Perhas UI where the controller and the node and the attribute are selected would be good…this UI
could also provide keyframe or expression functionality.

MODELLING A CUP
the biggest issue I’m having is the script failing…
workaround would be to stop auto scriptJob and start it again…
However it would be better to fix the fail points…
Seems to fail on deleting faces…got this error;
# Warning: Object, 'cup67', skipped. It is already a child of the parent, 'duplicateMeshGrp'.

seems that sometime the new cup gets selected
-- another error is;
newModelParent is select -r pCube1 ;

currently it errors after;
>>> nothing selected; job skipped

seems you can have nothing selected once but it fails the second time….is it trying to pass in a selection of a
panel or something?
I may have fixed it by adding ‘protected = True’ and then (neccessarily) force=True to kill the SJ

---
autoProcedure_simplemodel_016.py --- script still failing
will check to see whether its something to do with the scriptJob event/condition or whether its the duplicateModel()
script…
It may have been because of where I had the return meshNumber command… moved it now…try again

--- hmmm---- error as follows;

>>> Performing function >>>> self.functionObject.duplicateModel()
currentSelection is [u'pCube1.e[290]', u'pCube1.e[292]', u'pCube1.e[294:295]']
currentSelection type is <type 'list'>
transformNode is [u'pCube1']
>>> duplicating the model
newModelParent is None
newModelParent type is <type 'NoneType'>

added provision for if the parent of the new transform node is ‘NonType’;
    if newModelParent == 'NoneType':
        print 'newModelParent newModelParent == NoneType'
… it seems to be more stable …. 
hmm….. still seems to fail after;
>>> duplicating the model
**TEXTURING**

it captures the mesh with different UVs --- is there a script job for different UV changes???

OR maybe for now I just texture the cup and then apply the same shader to all models

**ADD ability to execute script and also to save the shader with the model**

**QUESTIONS while making**

**BandW ramp** → how to set alpha is luminance?? → I can do it on a file node

maybe `mc.setAttr("rampNode.alphaIsLuminance", 1)`

**THOUGHTS while making →**

duplicating the models is of course costly in terms of memory... there's a lot of mesh in the scene.

I wonder what could be "harnessed from these models? ie I guess they have shape and transform nodes now... could they share transform nodes?

could they be stored as point information?

could they all be transferred somehow to the final mesh?

**Research Outcomes →** The best case scenario (in my mind at the moment) for the research outcomes would be plugins that contribute to a change in the 3D animation paradigm... a change which encourages tools to be made that harness the process of making... that make the making process visible in the final piece... that expose the 'thinking while making' that goes on during production... that encourages the exploration of the "materiality" of 3D software (by this I mean all the things that 'push back' against ideal intention -- all the contingent aspects of practice) ... that encourage exploration of the making process as "collaboration".

To this end I feel that explicit outcomes such as sharable plugin nodes and/or demo movies of nodes in use (even if they are just prototype / proof of concept) would be more valuable that exhibition pieces (ie rendered footage).

**Progress Pics →** Progress pics proove that different ways of working create different "mistakes", ie different unforeseen outcomes... the images evolve in a different way. I guess the best way to show this would be to work with teh same subject matter in several different ways..

These pics show how the way that the geo is created (ie the topology and even more so the UV layout influences teh way that the image emerges / or "takes shape".

→ - try these textures on the progress models---->

Q. → What are the differences and similarities between this process and my cup paint/sketch doe years ago with guoache?
A. → the images themselves emerge out of my perception of the cup… which qualities stand out to me.. how can I represent those qualities with teh material (and tools) at hand. Then as I’m working with the tools/materials I’m looking at the image that emerges… what I like about it/ what strikes me might not be what I initially intended… I might not have to go that far…

This reminds me of Dianna Coole on MP ; she said something like ~ “its about emulation not representation”

....

11_05_14

ADD TO SCRIPT -
- add manual execute button ---- (while the SJ is still going you can press a button and execute the script; will the mesh use the correct number?) - this will be particularly useful when the UVs have been changed
- get rid of keyframe buttons
- save with shading network option (alongside manual execute); maybe this one adds a flag of some sort so that I know not to add the original shader to it...

EXPLORE AND RENDER -
- animated ramp on progress models
- render from a different camera angle
- try the colour with a Lambert shader (so that its a type of hybrid shader).
- animate history as a whole
- deform all meshes (various deformers; transform component, joints

SPECIFIC PROJECT IDEAS -
- Nick’s car. open the door with a joint. perform door animation on preliminary model which doesnt yet have a door. Model the environment… texture at different times of the day… people passing

- My dog… lying in her basket…model her then maybe eventually put joints in her to animate all the meshes

- trees or plants blowing in the wind… model and maybe texture them… as I go; as I find the form… so that modelling is a way of exploring the form / a way of looking (Gina isnt it always?).

- Plein air street scene … model what I can… capture what I can (I guess i would have to use my memory; eg of a person walking by). I guess all would be modelled at the centre of the… then they would appear

QUESTIONS
-Philosophy - what are the links between MPs emphasis on the primacy of perception; H’s emphasis on human Being as always embedded and about practical engagement / coping (as opposed to an emphasis on ....

- Texturing - how to avoid lots of shading networks? I was just thinking that the easiest might be to start with a bunch of surface shaders (maybe with ramp swatches leading into them)... I model and select faces to add different colours to. I assume that assigning shaders at face level will involve Maya adding a groupParts node and that the component list will be saved with the model.

Palette script ⇒ Tools to enable application of shader to particular faces...

Colour Palette⇒
- add swatch (adds swatch of default --- maybe the first--- colour or of the colour selected
- delete swatch
- edit swatch (could be a button or double click)
-- assign swatch → select faces and click on pallette swatch >> assigns surface shader (which already exists) to the faces
-- toggle viewport colour

???

--- could have some called LIGHT and some called SHADE --- then sliders for light intensity and light hue ---
these sliders update all the swatches ---

Face Selection ⇒

--- save selection lists in a drop down menu

--- button to add to current selection list

=====================================> then later I can use the “select objects in shding group” to do different things with the

faces across the board.

So this way doesnt sound like it relies on UVs ..........but we will see............

thinking about smoothing --- shading on a per face level...perhaps best to subdivide (polySmooth) the models

and then grow or shrink the component list associated with componentList for each shader

----------

→ starting on teh colour swatch UI...useful cmds

mc.pallettePort()

flags ⇒ colorEditable == for current cell

changeCommand

colorEdited

rgbValue

redraw

eg # select cell #30

cmds.pallettePort( 'palette', edit=True, setCurCell=30 )

# return RGB value for this cell

cmds.pallettePort( 'palette', query=True, rgb=True )

mc.grabColor()

grabs the colour under the cursor…

mc.colorEditor()

---------- dont need these cause Maya automatically brings up the color picker ;)

....

12_05_14

// Create a Shading Group

string $SG = `sets -renderable true -noSurfaceShader true -empty -name "mayaLogoSG"`;

// Connect the material to the Shading Group.

connectAttr -f ( $material + ".outColor" ) ( $SG + ".surfaceShader" );

To determine which Material is associated to a Shading Group, get a list of all connection from the Material's

".outColor" attribute. Iterate this array to find all Shading Groups. The ‘nodeType’ command will identify a Shading

Group with the string "shadingEngine"

PIC OF “colourPallette_008.py” UI

I just realised that it might be good to automatically associate face selection sets with colour swatches.

- automatically update selection set when a colour is assigned

- create colour swatch next to the selection set
- maybe update button clears the set and adds the members afresh; and takes members from other sets

So I make an automatic link between;

\textbf{selectionSet} \rightarrow \textbf{swatchIndex} \rightarrow \textbf{shader}

...so that each updates the other...

the \textbf{number of sets} \Rightarrow \textbf{the number of swatches} \Rightarrow \textbf{the number of shaders}

\textbf{-------- could I test this workflow first to see if it is interesting\textbf{----------}?}\n
\textbf{Useful}\Rightarrow \# \textbf{Get a list of all the rendering sets which }\texttt{`coneShape1`} \textbf{belongs to}:

\texttt{cmds.listSets( type=1, object=`coneShape1` )}

\begin{verbatim}
13_05_14
\end{verbatim}

\textbf{colour pallette code fixes;}

- create relationship between selection set and SGset
- update set colour button
- delete selection set (and its UI elements)
- when I edit colour swatch ; add reselect original selection... in fact I never really want to select the swatch
- update pallette and selection groups according to SSs and selection sets already in the scene (by button or automatically on initialisation)

\textbf{code queries;}

- when assigning the swatch to the cube faces I get this warning;

\texttt{# Warning: Cannot add the following items to the set since the set has restrictions on membership: SScell_0 #}

...seems to be trying to add the shader to the selection set

\textbf{Thoughts;}

when i update the topology its a different result depending whether there are more or less faces...

It would be good to find a way to suggest what Maya should do with new faces, ie what group to put them in...

First I guess I would need to have a stronger link between selection set and SGset

-- could the weird way that the colour jumps around on the cube actually inspire movement/ character/ animation/
and/or form?
- Materiality; on the one hand for me this term refers to the way that the software (inadvertently) behaves.. the aspects of it that are not part of its perceived functionality. For example the way that the coloured faced move over a polygon model as the topology of the model is changed… or the strange way that a model deforms when its topology or history is changed.
However I also seem to be seeking ways to control (or harness) this behaviour… is this a contradiction? Perhaps not… perhaps the key word is “harness”… or explore the materiality of the software

ASSIGNING A SG TO SELECTED FACES
found this link;
http://ewertb.soundlinker.com/api/api.008.php

can I CONNECTATTR to connect the shading grp set to the selection set??
- creating an object selection set creates “groupId” node

---> print mc.listAttr('groupId1')
>> [{u'message', u'caching', u'isHistoricallyInteresting', u'nodeState', u'binMembership', u'groupId'}
---> print mc.getAttr('groupId1.groupId')
>> 27

Gina it seems that when I create a selection set which includes mesh components Maya creates a groupParts node and a groupId node…. This is interesting cause groupParts is the node created by Shipkovs mel script (to change the history on a face extrusion for example).
By default does creating a face extrusion create a groupParts node? ii dont think so…

...can i connect two componentList attrs? seems not

................
SETS and GP nodes----
Have just been trying to see if I can have one set that drives a selection set and also a shading GRP set…. havent found an easy way with cmds.
thinking it might be best to work with faces as lists
- eg setting selection set members updates SG set members
- and vice versa -- updating shading grp set members changes selection set members
- It seems strange that I cant make these the same set
here is a link that shows how to add objects (and components to a shadingGroup using the API… its in C++ but might be useful when I come to make a node that connects SG and selection set and maybe GroupParts component list…
Also that drives the component list or set membership with a bounding box….

14_05_14
beautiful day today - perfect for “plein air” 3D trial --
I would like to apply shaders as I go and also to save iterations of my model as I go…
I’m thinking of what I have written about as a painter’s common aim being to focus on/ or represent the “overall impression” of a scene before it is broken down into component parts…
Right now I’m wondering how this aim relates to the ideas of Gestalt theorists and MPs ideas about perception (and maybe stuff he talks about in “the Structure of behaviour”.
focus on Gestalt / Perceptual experience
So in making a plein air 3D study I will be focusing on my perceptual experience of the scene...I will try not to neccessarily break it down into component parts (treat the subject as I know it is; ie a collection of distinct...
My perceptual experience changes as my focus shifts and the lighting shifts and even the objects before me shift... in this way even an animation with static subject matter might be moving/changing/animating...

--- overall impression ---- shifting detail

**listen to the materials / remain open to conversation**

Allow strange use of tools to suggest content as I work

Also the software will suggest ways of moving from one model to another... or ways to treat / deform / texture all models.

**My tools**

save/ duplicate model script;

for now not auto -- just press button

it must duplicate connection to shaders

Colour swatch UI;

  to help me quickly apply shaders to the emerging model

Maybe put these both in the same window


---

**NB. I have continued this doc as pleinAir3D project notes**

This pic shows a 700f sequence that combines plein air building and figure sketch “phoneFigure”

16_05_14

I will now try combining coffee cup and hands figure.

Try coffee cup ?and figure coloured.

Try them both Black and White.

Try adding a screen grab of the animating ramp texture? or visualising this somehow? but how??... maybe add a NURBS plane with the same shader...

--texturing coffee cup and wondering how I could capture this process of working/ tweaking/ texturing...for now duplicate the shader network? or save UV maps?
...based on lif study but then the image takes on a character of its own…

**UV MAP REAL ESTATE**
above right is image constructed of layered ramps. the area in the middle (brown) is the table top. it has much less room on the UV map than any other area and this is why the splodgy shadow occurs.

………………

17_05_14

**WALK CYCLE**
- using “RapidRig” for simple biped rig
- starting with a few main controls and creating a biped typical walkcycle (following tips of internet site; http://blog.digitaltutors.com/12-steps-to-a-great-walk-cycle/

I keyframed the following controls to get a basic walkcycle “framework”...
footL - ty, tz, rx, heelball
footR - ty, tz, rx, heelball (could these be done once then offset? in the past I have done so with an expression so that one foot drives another)
Root - tx, ty, tz, rx

**AUTO ANIM** for walkcycle
Questions;
can my auto anim script be use to generate basic walkcylces which I then build on?
Does working this way (with a few keys set down) all a character to emerge?
ie. does it feel different to tweak something already in place and listen for “believability”, or “poignancy” rather than to work to a predefined goal?
Also, if the models are glitchy, dynamic, and incomplete, can I get away with more bizzare or “incorrect” walkcycle movement?

**AUTO ANIM** visibility of models
- autokeyframe
- child expressions
→ a quick test shows that this seems to work. The child delay = length of time that driver is visible. The driver is hidden for a length of time VisLength * number of children
-- why does there seem to be a limit to the number of children? is it to do with a limit in the no. of attributes I can add? test this by eliminating “offset” attribute --- it was the offset value --- could amp increase be used for the distance apart of the keys?
OR would it be better to set something up so that the vis of one relies on the vis of the previous? that might be a better way to get the spacing to vary across the timeline. Also to be able to move backwards and forwards.
→ Maya 2014 has a LATTICE DEFORM KEYS TOOL ...

**Auto keyframe script fixes / changes**
→ offset (as a fractional number) as doesn't work for the last key in the loop… I guess it needs to get the next key …??
→ it works for driver loop 4 + target loop 4 (all except the last key) but not when targetKeyIndex0 ref driverKeyIndex1,2,3 or 4
→ it seems to find the next key in the target loop not in the driver loop…???
⇒ maybe its to do with teh way it calculates key value???
⇒ is teh time correct?? check this first
---------------------AAAAHHHH-----
trying a new tack…. once I have the time of the current key… maybe I use mc.keyframe() functions to calculate the time of the next key…
eg I could use;
indexValue = Query-only flag that returns an int for the key's index
index =

    index of a key on an animCurve
    --------- TRY;
given a time and an animation channel;
find the key index then the time of key[index+1]
→ index = mc.keyframe('pSphere1.ty', time=(1,1), indexValue=True, query=True)
print index

    i = index[0] +1
    print i
nextKeyTime = mc.keyframe('pSphere1.ty', index=(i,i), timeChange=True, query=True)
print nextKeyTime

--------
now it doesn't seem to be getting the keys to lay

**GESTALT**
interesting how Gestalt visual perception theory emphasis relationships and context which is what my script are about (particularly animation scripts).
-----------------------
19_05_14
Using my autoKeyframe script (after having made some amendments) to create walkcycles on rapid rig. It wasn't successful on whippet run but this to I want figures to be more abstract / sketchy …

**SCRIPT CHANGES**
- How to optimize? maybe delete unnecessary print statements
---- call this one AUTOKEYFRAME and the other CREATEXPRESSION

working on WOMAN WITH BAG using “AutoAnimDev2_010.py”
Using a number of my custom UIs

THOUGHTS

→ AutoKey could be good for shadows etc
→ I have noticed that the pose of the sketch is “incorrect” for a walk...ie her arm and leg are both forward
...is there a way of using my script (autokey) to aquire “incorrect” walkcycle movement that represents the perceptual experience of a walk?
→ I just went out for a walk and noticed that no other pedestrians looked like a typical “walk cycle”. Some people swung their arms and had no up and down movement, other were teh opposite...
---NB. I have tweaked the keyframes from their default values to suit the exact pose of the sketch.
→ could try shadows as animated GEO or use SHADOW PLAY script

“GROUND ROUGHNESS”

making the womans shadow with poly planes connected with constraints and clusters;
I put this expression on the shadow geo and thought that I could add a “ground roughness” attribute;
\[ pPlane6.translateX = \text{generic\_ROOTC.translateY} \times \text{rand}(2,2.5) \]
increasing such an attribute would increase the random function making the shadow jitter...
A character emerges
I added a second expression based on her root ctrl and both expressions (one controlling spine ry and one controlling spine ry) are connected to the child offset and mult attrs. This is a mistake but it actually seems to work fine… fiddling with this attribute gives her attitude or makes her slumped and tired…

A ⅓ increase in speed on the driver node ‘

RECORDING CHARACTER/ MOVEMENT ITERATIONS
…altering the character’s animation. But these iterations of her character are not being recorded.
Would it be interesting to save the keyframes as a trax editor segment and then blend them all together???

SHADOW GEO VIS
- an expression for shadow geo with an if statement…
if (pedestrian1_IShoulderJ.rotateY > 0)
    shadowGeo.visibility = 1

..................
**General Thoughts on Digital sketching;**

The screen is illuminated so can see colours for night sketching

Its 2D so can have blobs of colour (like the red in image 3) that almost hover in front of the object
tend to lay down a ‘ground’ colour and work from there
sensitivity to edge quality (blurry or sharp)
it has (surprising) intrinsic qualities such as blurry edged paint strokes then bucket fill of the same colour (image far right)

**- RAE ST NIGHT SCENE -**

AS a painter I usually start with a base colour…. sometimes its about blocking this base (‘ground’) colour in various ways… above digital sketches show the same thing

- start with BG colour ⇒ add this to colour tools
  **⇒** colour surface shaders via ramp
- dont use auto process but do record modelling process (via shape of transform)
- refrain from building/representing the st scene “as I know it to be”

**------ RECORD COLOUR TWEAKS**

Above shows a UV ramp type… can play with interp and ramp type for different transition between colour tweaks…

decided to record colour tweaks by adding a ramp swatch to the main ramp..

- the tweak number is the same as the colorEntryList number
- the interp type can initially be set to none
- the latest tweak is active when its down the bottom (ie tweak[1] is at position 0.001)
the latest is inactive when its at the top (position 1)
the first colour colorEntryList[0] stays at position 0

- link these the position of these colorEntryLists to a controller in PostProduction

I have decided to do this with nodes as pictured above.

**CNTRL ⇒ clamp ⇒ setRange ⇒ colorEntryList[x].position**

don't need multDivide unless I want finer cntrl

CNTRL is a float from 0 to numOfTweaks
numOfTweaks = number of colour entries (not including entry[0]; baseEntry)

entry0 stays at 0 (this is base colour)
entry1 moves from 0.999 (offPos) to 0.001 (onPos)
  clamp → min0, max1
  setRange → min0.999, max0.001, oldMin0, oldMax1
entry2 moves from 1 (offPos) to 0.002 (onPos)
  clamp → min1, max2
  setRange → min1, max0.002, oldMin1, oldMax2

movement range for both = 0.998 == 1- (0.001 * numOfTweaks)

-----------------

22_06_14
more on RECORD COLOUR TWEAKS...

**Production Tools:**
tweaks to the swatch add an entry list to the ramp with a rampSwatch attached
  startPos = 0.001 * tweakNumber/indexNumber

**Post Production Tools:**
add controls for cycling through the colour tweaks
  -add a tweak attr to the shader; float from 0 to n (numberOfTweaks)

------ got them all going from 1 to 0 at the correct range of tweakAttr ------

-----------------

23_06_14
- set correct entry range
- make one multDivide
DONE

page 123
Thoughts re my project - glitches as aesthetic suggestions

been thinking about how my project is basically about using glitches as aesthetic suggestions. Its also about challenging teh habitual way that we see/look at/perceive things.

eg normally we “look through” the way objects appear to their use value. Van Gogh presents a pair of shoes and I suppose that we contemplate the shoes and see them in a new way…

Interestingly this embeddedness/facticity is possibly what ensures that perceptual experience doesnt cohore with (or at least isn't limited by/ comprised of) the image ?projected onto/recieved by the retina. In other words I wonder if there is a paradox in me thinking that my work is about presenting ordinary objects in a new way, looking at them in ways that are beyond the habitual; and so I guess I dont want this just to be like approaching objects like a camera OR like Impressionists (or at least how Dreyfuss thinks of the Impressionists)...ie I dont want to approach them in a formulaic fashion. So its beyond the habitual (which means seeing objects as useful/ stable etc) but not neccessarily divorcing perception from use value, meaning, cultural assumptions ect. Perhaps the point is that you never can... you can never stand back and make sense of it all.

BIKE MAN
OLD DOG → 2hrs

tolerance for the incomplete.... strategy #? == time limit

I have used the strategy of observational work (ie still life etc). Now try timelimit.

- after 2hs I have 1 ("finished") model and 70 process models.

Next step is to;
- animate a walk cycle (sliding the main cntrl)
- bind all models to the joints (including shadow models which I want to include)
- add blend between each model and vis trail - I havent done this on bound / rigged mesh before...ie. I havent combined pleinAirStillLife workflows with MasA…

---- I just tried jumbling up the order of the models by using my renamer script
     could make a random order script (duplicate my favourite shape or tranfsform nodes and shuffle)
Move rig into position, to suit model (as closely as possible)...
Bind mesh (all of them?)
Create selection set of cntrls
key foot cntrls
→ I feel that I need ref material of dog walking………….  http://www.youtube.com/watch?v=xAJlcQP0Kv4

when I added the toe to the skin weights and added weight teh mesh changed considerably. it turns itself inside out and pulls some of teh belly downward…
its more dynamic; over the top, exagerated, but I quite like it…it suggests character…?
Because its a low res mesh the deformation is more radical/extreme/greater…

THOUGHTS
- I could use the same animation and just change the skinning to elicit a different character, a different model/form… maybe a group of dogs…

....
14_08_14

Back to still life cup and arm detail…
Rendering it with the models having different UVs then will run a script to transfer UVs from the final model to the previous ones and render again.

25_10_15

WHIPPET SKETCH

working on grid
from sketches
This document provides a place to record general project work that doesn't (yet) fit into a particular category...

29_05_14... I now think of this doc as a place to record thoughts re my 3d anim work, programming as well as reading; it feels like time that it all came together.

......

17_04_13

I have a gmLockAndHideScript that locks and hides attribute in teh channel box
...I have been reading about classes and inheritance with Python and OOP...
...haven't yet found a good use for this concept... best example seems to be with Maya nodes themselves... i.e.
make a series of cubes... change attributes... they all “inherit” from the cube class and are
changeable/customisable...
BUT gina this isn't quite polymorphism --- that would be more like creating a series of shapes and changing the
same attribute on all of them.... the same attribute means something different to each of them...
...I can imagine this concept for colour (light over a cigarette packet) as well as movement (a number of trees
blowing in the wind) ...
GINA perhaps start with the 3D visuals / experimentation and let scripts evolve from there...

POLYMORPHISM and colour / texturing...
laying out the UVs on faces and then applying a ramp is one way of achieving a different surface quality of
different areas of the surface i.e. animation of the ramp means something different to different areas of the
object...

UVs could move... or ramp could become more contrasting... or the contrasting area of the ramp could shift...

---

this reminds me of edge quality (in this case the edges of the object)...
Here's a geo test... I am just changing the creation parameters of a pyramid primitive....
as an animation it gets a great little wobble as divisions are added. GINA - alone this is not polymorphism but maybe with a couple of primitive objects which have been tweaked differently it would be??

and just by remapping the UVs of a few faces…

The recipe for the above experiment is;
make a polygon pyramid primitive (num of side = 5)
scale it on Y axis
polSplitRing on each side (it splits from bottom to almost to the top)
move new edges outward (the further you move them the more extreme to wobble)
animate the polyPyramid subdivisions height attribute (in this case of 1 to 100)

28_04_13
⇒ see more in “figure sketches” folder; https://drive.google.com/a/student.rmit.edu.au/?pli=1#folders/0B9tGxxPjQDclcWlJcGNDNlhfc2c

MATERIALITY
small scale sketches of figures almost always inspire me…
use RAPID RIG to create short anims of figures based on sketches,
The process is similar to making a painting based on figure sketches; in this case it seems possible to slap down some colour, smudge some paint (i.e. give the material qualities of the paint some freedom) and then tweak, enhance, cover up the results so that particularities of the character emerge. In this way the character’s sex, age, complexion, clothing might all be details that become solidified throughout the making process. In this way the resulting figure might hover between known and unknown; earth and world…. it emerged through this known unknown process (ie a juggling or exploration of these qualities, rather than having ambiguity applied after the fact).

IF THIS IS POSSIBLE with paint (i.e. its quite easy to imagine… and I have experienced it… and maybe illustrated it with Park painting) what is the equivalent with 3D? how to come from ambiguity and complexity?
HOW TO DO THIS?
- start with positioning the Rig; like an amateur, and build model (or bits of) around it
- start by modelling then add rig…
- build rig in symmetrical pose and then move into position? OR move proxies to pose and then build rig?

- I’m looking for forms that are not entirely intentional… maybe this is were tweaking history comes in… could tweak (or animate) history and stop at certain places when I recognise something “good” (ie a form that is reminiscent…)… Be interesting to see what textures do as form changes…
- use polygons or NURBS? or can I use a combination?
- working from pencil sketches often still makes me want to experiment with curves and surfaces… GINA remember that lofts, extruded forms etc can be polys as well as NURBS…

GENERAL THOUGHTS
- try in wire and in Das ?? are they equivalent?

29_05_13
Book / outcome Thoughts
I want to start collecting images and animations - photographs (of work environment etc), scanned drawings, animation stills and uncompressed movies, screencapture movies, process pics. File them under project headings.
Some pics should be very high res (eg 30cm), some high res (20cm), some medium res (10cm)...
very high res 4k = 33.87 cm
high res 2160 pix = 18.29 cm 1920 pix = 16.26 cm
medium res (10cm) 1080 pix = 9.14 cm

MP, Style, etc
I have attached Merleau-Ponty's essay "Cezanne's Doubt" and also a paper I read yesterday about MP's concept of style.
I'm interested in this paper (and more generally in MP's work) because it seems to argue against the idea of style as a superficial / surface effect (which is an idea that the application of NPR filters in 3D animation seems to promote) and talks about an artist's (or anyone's) style as inseperable from the way that they see the world.
(??...it could be argued that the artist chooses the filters or tweaks them or develops them in an attempt to find their expressive gesture…) Also its not that we have some types of representations that depict the world as it is and others that are stylised... I think there are only styles.

On page 157 the paper talks about MP's concept of "expressive gesture"; it says "The painter's project is to struggle to find the expressive gestures which will allow him to paint the way he wants to paint" ...."style emerges from and appears as an expressive gesture, which is an extension of the body's basic capacities to intentionally intertwine with the world."
It seems obvious how this idea of "expressive gesture" relates to painting..but what does it mean in relation to 3D animation?
I feel sure that the artist’s struggle to “find teh expressive gesture” is not necessarily won by using a Wacom instead of a mouse...

Regarding the last part of that quote about expressive gesture being and extension of "...the body's basic capacities to intentionally intertwine with the world."... it reminds me of a book called "The Zen of Seeing; Seeing/ Drawing as meditation" by Frederick Frank (a book that I borrowed of Andy and have read bits of). Here are a couple of quotes from this book;
"while drawing a rock I learn nothing 'about' rocks, but let this particular rock reveal its rockiness"
"In order to draw a horse, draw horses until you practically become a horse - not 'horses in general' but always that particular horse you are drawing at a given moment... feel the tense curving of its neck in your own neck."

Its interesting to think about this last quote in relation to a typical 3D animator’s way of depicting a horse (even a particular horse; Phar Lap for example) which involves gathering horse images and footage off the internet, studying these images and using them as reference to build, texture, rig and animate virtual geometry.
I guess it could be argued that even when using this typical 3D anim process there comes a point in production when you (the artists) "feel the tense curving of [the horse's] neck in your own neck"...[by this I mean that either the photographed horse you are studying or the virtual horse you are building might cause you to feel "the tense curving of its neck in your own neck"... this can of course be talked about in terms of mirror neurons].... but it still seems to me that typically the software requires the artist to do a lot of study of 'horses in general'...
(via film, photographs, anatomy diagrams etc) and that this makes it hard to do the "unlearning" required for attention to particularities of a given encounter......
Also this image of the 3D animator gathering source material as required and using it for a given purpose puts too much emphasis on the will of the animator/artist .... it seems at odds with a practice that aims to "let this particular rock reveal its rockiness"...

When I reread the first page of "The embodied Mind" the other day I noticed that the authors say they consider the book as a modern continuation of Merleau-Ponty’s "program of research".... I remember when you put me onto the work of Francisco varela...
So, given your interest in painting, cognition and embodiment, it strikes me the you might find MP interesting and that you might be in a great position to actually understand what he’s driving at .... I’m really not sure of the extent of my understanding ...
If you're interested, it would be great to read some stuff and have some discussions perhaps with the view to writing a paper together or something.
I have heaps more merleau ponty material including (papers and books). I know that you are ridiculously busy but let me know if you want any more material or if you want to discuss any of this further.
I'm also looking forward to hearing about your new artwork ideas. Let me know if and when you have time to catch up again. I'm going to send a meeting invite to you and Stefan for Wednesday 11th June but I'm around if you want to catch up sooner.

Cheers,
Gina

........

TO DO THIS WEEK -
-Read and transcribe notes from Dianna Coole in New Materialism book on Merleau-Ponty (how does she relate MPs work to Materialism?).
- Create short experimental figure anims. Use “materiality” of the software. Start ambiguous so that the result is Impressionistic/ and ambiguous.
-- ?maybe do equivalent painting projects

11_06_13

ORGANISING AND REUSING SCRIPTS
I am now thinking that I need to start using modules, packages etc. My codes are getting complex and I feel the need to make them modular....so that they reference each other.... This might help me to organise my codes into groups (I have tried this before but never know exactly how to organise them).

Some specific questions;
How to use (and what are) Sublime Text projects?
adding colour...
I laid out UVs after skinning. Not a good idea. Will do a save as and then delete history on mesh. that worked. Reduce tool doesn't change UV layout!!

Engagement = very high (3-5; scale 1-5)
I think there's something emotive in “LibraryMan3_004.mb”; he looks like he's trying to snap out of it, or suspicious of what's going on.

--------
I ended up getting it to work well enough. didn't use random function.

    Used this on dummy rotate;
    if (frame == 1) seed(1);
    if (frame == 200) seed(1);
    float $amp = amplitude.amplitude;
    float $clamp1 = scared.scared;
    float $clamp2 = -1 * scared.scared;
    dummy.rotateY = clamp($clamp2,$clamp1,(noise(time * 2)* $amp));
    and animated “scared” attribute

--------
Strangely the Blender doesn't work; is it taking the first blend value and using it for the whole animation? why?
It doesn't work when I animate “Scared” value either. It seems to take the first value and only use that.
I will render out a slightly repetitive one regardless then move on with some simple connections and see if I can combine keyframes etc.

“LibraryMan2_007.mb”

wondering why there is a glitch at frame 100 (a sort of spasm). It is because of the random expression designed to clip the noise curve at different values....I can't work out how to fix it yet. Will probably animate the scared value for now (maybe that would be a way to get different clipping values; plug noise into scared input?)

This is the expression on Dummy rotate Y;
    if (frame == 1) seed(1);
    if (frame == 200) seed(1);
    float $scared = scared.scared * 10;
    float $offset = offset.offset;
    float $lag = lag.lag * 0.01;
    float $amp = amplitude.amplitude;
    if ( frame % 100 == 0 ) $max = rand( 0,$scared );
    if ( frame % 100 == 0 ) $min = rand( -1 * $scared ,0 );
    dummy.rotateY = min($max,max($min,(noise(time)* $amp)));
Can use this expression;

```
int $timeoffset =2;
int $Time = ('currentTime-query') - $timeoffset ;
objectB.value = 'getAttr-$Time objectA.value';
```

NB. backticks for storing values from the `currentTime` and `getAttr` functions

**Start questions;**

how to best clamp values? try using clamp() instead of min() and max().
I cant seem to use a random function (cause it creates a jitter / per frame anim) but one way to get several values for $min;

```
float $min = 0;
    if (frame % 40 <= 15) $min = 1;
    else if (frame % 20 <= 15) $min = 2;
    else $min = 1.5;
```

I take that back. This really seems to work well to produce a ramdom value every 50 frames;

```
if ( frame % 50 == 0 ) $max = rand( 1,2 );
if ( frame % 50 == 0 ) $min = rand( -2,-1 );
```

Ok. I’m going to try and put it all together on my figure.........

```
........
24_01_13
```

**Intended activities;**

add more procedural animation to body controllers baring in mind the later desire to animate/alter the amount of animation, and maybe combining with keyframe animation.

**Start questions;**

how do I make the amount that the head turns non-constant?

```
- activities;

    found this to change direction;
    if ( frame % 60 == 0 ) $xDir = rand( -45, 45 );
pCube2.rotateY = $xDir;
```

Found a way of creating random loop!!!!

```
if (frame == 1) seed(1);
translateY = rand(5);
```

This works for rotating with noise expression and truncating the top and the bottom

```
float $offset = 10;
float $lag = lag.lag * 0.01;
float $amp = amplitude.amplitude;
if ( frame % 50 == 0 ) $max = rand( 10,25 );
if ( frame % 50 == 0 ) $min = rand( -25,-10 );
pCube5.rotateY = min($max,max($min,(noise(time)* $amp)));
```

I am wondering how to easily reduce or extend the truncation per object?

This makes it go further

```
if ( frame % 50 == 0 ) $max = rand( (10 * pCube5.trunk),(25 * pCube5.trunk) );
```
if ( frame % 50 == 0) $min = rand( -25 * pCube5.trunk),(-10 * pCube5.trunk) ;
pCube5.rotateY = min($max,max($min,(noise(time)* $amp)));

To make it go longer between....? would that look better?
float $offset = 10;
float $lag = lag.lag * 0.01;
float $amp = amplitude.amplitude;
if ( frame % 100 == 0) $max = rand( 10,25 );
if ( frame % 100 == 0) $min = rand( -25,-10 );
pCube6.rotateY = min($max,max($min,(noise(time)* $amp)));

I'm trying this and changing the frame % number. I'm not sure why it doesn't work??

questions arising:
  what exactly does linstep() do?
  would blend colour node be useful?

Surprises;

Engagement;
very high. I have to go home now but am intrigued, feel like staying.

thoughts / possibilities;

............... 23_01_13
spine top = 5.956, -7.089, -7.852
fig01_lKneeIKC = 8.72, -1.289, 0.99

  what does the animation blend node do? - I have been using “add double linear”, should I be using
anim blend node instead?

  checked out addDoubleLinear for use to make stretchy joint chains;

--------generally engaging experience, but I got sidetracked from the initial project and ended up doing the
stretchy skeleton tutorial.

............... 22_01_13
Intended activities;
rebind model to skeleton and experiment with animation.
- wondering if a tool (or method) to create animations from observational pencil sketches would be interesting?
i.e. animator directly observes poses, create figure in first pose, bind figure, move to second pose, remodel
figure...what would happen in between? in between have always been boring (straight, predictable, linear,
transparent), not adding anything, but homogenising and making generic, sapping the life. It could be said that
this is the default will of the software, but is it the nature of code, of the computer, of digital technology, of digital
culture?

Start questions;
thinking about animation - can I make interesting movements using combo of procedural and keyframe anim?
would procedural (or connections/data translation) feel/look like it was beyond the grid of knowledge/ the
regime of representation?
Also wondering the same with modelling - combining deliberate vertex positioning with accidental, random, or procedural positioning. In each case I recognise something desirable/interesting.
- Could these processes (of animation and modelling) provide a way of dealing with unknown, undifferentiated areas (areas not brought to conscious attention)?

activities;
automatic UV mapping
skinning -
animating

questions arising;
should I make the model detail homogenous and then loose detail with SOuP or similar? is it a case of getting the correct shadows without adding superfluous detail? in which case would normal or bump maps be useful?
should models be much much more minimal? I dont yet know the workflow
should I fix skinning issues?
should I bind to only a few joints? YES
can I rename joints? YES

I am left wondering how to change models during animated skeleton;
bind several and animate visibility
use poly reduce and transfer attribute tools to create a hi res mesh then reduce from one to another
Will Python be useful for expressions to animate procedurally, or will it be more useful for creating User interface windows etc? (eg, sliders that I can then animate)

Surprises;
mistake - bound to joint hierachy instead of selected joints

Engagement;
I am finding it very boring today (might go get cardboard for another figure model). could I use the physical nature of the paper (and folding technique etc) to influence the form of the figure? the movement of the figure? the colour of the figure?
end of day - ended up getting into it once I started exploring procedural animation using nodes in the node editor and mel expressions.

thoughts / possibilities;
could I make a very simple figure model where I work out/uncover/ evolve the logic of the model (ie edge moves up as knee bends)? could these connections be created in python (ie. connections between components and objects)?
ambiguous movements?

Do I only add detail as needed? (ie if he moves fingers detail added then)

How symetrical should I keep it? at the moment I default to symetry

- can I model in such a way that you can tell how the object was perceived? ie. experienced from above, from a distance, nochalantly, intensely.
- can 3D modelling tools help me “make it up” in such a way that it adds to” the model (makes it more specific/ more emotive) rather than detracts (makes it more generic).

Surprises;
- moving along normals a group of vertexes is a bit unpredictable

thoughts / possibilities;
i wanted him to tap finger; this could be by moving a vertex rather than rotating a joint
- thinking of Rodin; not neccessarily accurate, but kind of convincing
  also makes me think of the process/ doing, then recognising something (something attractive / convincing / something that works)
  this makes me think of metaphor (Schon “generative metaphor”, Bogost “metaphorize”?)
- How is this different from typical 3D sculpture?
  I dont aim for detail or “accuracy”
  approach not instrumental.
- i can imagine a street scene where polygons are “stretched to their limits” i.e someone gets up off a bench but they are joined,

------------------------------
15_01_13

started this project yesterday. continuing today. getting sidetracked with Python questions which is keeping me interested. so far I have not done anything that cant be redone with click in the script editor (I guess this is procedural or parametric modelling?).

........
import maya.cmds as mc

#create cube and camera
mc.polyCube(sw=2, sh=2)
m.cform(t=(0, 0.5, 0))
mc.camera(aspectRatio=1.333, displayGateMask = 0, displayFilmGate=1, displayResolution = 1, overscan = 1.25)
m.cform(t=(0, 0, 10))
mc.setAttr ('camera1.visibility',0)

........

I want to make a figure model with the same appeal , intrigue, charm as my pencil sketch =>
some shapes are well defines, others are a mess / hard to define.

music on now...maybe just model... try to focus.

....
after 10 mins, I’m finding it difficult to model, wondering if it would be best to start with a very generic model and rig.
Move into position and model from there???

tried this today. starting to get interesting when I am tweaking pose of man.
Pose tweaked ⇒ libraryMan_003
now alter mesh ⇒ [while bound? add edges first? or move points first?]
PRODUCTION NOTES
Subjective / secondary/ accidental/ extrinsic
working title: Colour and Light

working with colours and ramps...
will try to work fast and do an experiment each day or each two days of work.
Some starting by playing/exploring eg. python code (and allowing the exploration process to “suggest intent”),
some starting from a way I know and remaining open to unforeseen opportunities.

00_04_13
Aims and intent at outset of work session;
General steps;
Processes, tools, and techniques used;
Unusual (less common) processes, tools, and techniques used;
Problems arising during production;
Welcome surprises arising during production;
Questions arising during production;
Thoughts arising during production;
Summary;

24_04_13
Aims and intent at outset of work session;
- create an animation that captures the sensation of passing cars.
  First animation uses the ramp shadow “blur” and “interpolation” attributes.
  start subtle and with each car add more wierd ramp movement.
- maybe add some twigs to the top of the tree, or a foreground or midground shape.

General steps;
- I made a few variations of a 200 frame animation. Cars move through the scene and change the lighting.

Processes, tools, and techniques used;
- ramps within ramps; layered textures; layered shaders; projected textures; UV textures; expressions linking
  textures to controllers (to be centrally controlled)

Unusual (less common) processes, tools, and techniques used;
Problems arising during production;

Welcome surprises arising during production;
- I was able to introduce colour by using the offset time function in expressions which illuminated the tree trunk

Questions arising during production;
- ways to introduce colour without assigning coloured shaders...
- could all objects have the same shader and different expressions so that they react differently (a figure is what
  we differentiate from the BG).
- I sort of feel like my work needs a story and I’m wondering if the story could be the theme of my research? i.e my
  research is about knowledge, (maybe the body?) could it be about a painter and a 3D animator? It is about
  translation (interesting how translation is a word for moving things in 3D).

Thoughts arising during production;
- make an animation where the colour appears only through altering RGB values (start with black and white).
- try inverting the blurry BG
- try offsetting expressions like this;
  int $timeoffset =4;
  int $Time = (`currentTime -query`) - $timeoffset ;
  ramp7.colorEntryList[0].colorR = `getAttr -t $Time LIGHT.lightTree`;
  to acheive flashes of colour

Summary;
....
23_04_13
Aims and intent at outset of work session;
I am looking at the image that I created yesterday. It has a style of its own that is starting to emerge.
Next I intend to;
- get rid of (or change) the foliage because it doesnt suit the shadow
- hook up some ramp colour input colours and positions to show changes in light source and changes in crispness
  (detail / visible differentiation) of form.
- I also think it needs an overall tonal treatment.

Problems arising during production;
I decided to open the scene in Maya 2013 so that I can use the Node Editor to make connections (specifically to
link up the “crispness control”. Annoyingly the scene gives very

Welcome surprises arising during production;
- surface shader transparency works in a strange way...

Thoughts arising during production;
- Add transparency to the plane of buildings
  -My experiments are about ways to stay focused on the image as a whole (the big picture); ways to work from the
general to the particular, and not get bogged down in detail ; and not get wedded to an initial intention / idea.

Summary;
- several interuptions but overall engaging. Felt like I was able to play with the connects I have built up. Many
surprises and new ideas.
As I’m leaving I’m thinking of the projected texture feeding into the colour gain of BG buildings. should I duplicate
the plane and use that on the transparency (perhaps invert for transparency of the blurred plane).?
Aims and intent at outset of work session;
working with limited colour at this point (I might do a quick colour/shader test).
I am keen to test out my latest shadow making script.

Problems arising during production;
- I can't translate the ramps sideways via the 2D placement node of the main ramp. Answer is to use the placement nodes of the Vramps (a bit annoying to select them all)

Welcome surprises arising during production;
- I just duplicated the tree model and moved it a bit to one side. I applied a new surface shader and darkened the LT via a multiply divide node. It works well as a shadow and can be moved around with the changing light source.
- Using a series of ramps to create the impression of buildings in the background. It's a bit like painting with a limited type of brush (perhaps a flat one). I'm making it up as I go along which is more interesting than copying a picture.

Questions arising during production;
- Would it be interesting to constrain myself to only procedural texturing? It might allow unexpected animation opportunities.

Thoughts arising during production;
At the moment it feels very tentative. Should my moves be bolder? More dangerous?
- Would negative space (shapes) be useful?
- At the moment it's all very smudgy. Would it help to introduce texture?
- I am also sort of working across the image; from one object to another. Would it help to assign the same shader
to multiple objects to start with? so keep working from general to the particular. 
what if I only used default ramp colours?

Summary; 
next day - despite a lack of sleep the previous night, I was engrossed in my work yesterday. I enjoyed using the ramp shadow script and playing with the results.
The offset on some of the ramp Texture Placements put sharp edges on the road, therefore this is where I placed the white road lines. I particularly enjoyed the ease of making the row of background buildings (a simple plane with ramp textures linked to one surface shader). 
I did have a headache at the end of the day.

Aims and intent at outset of work session;
Try using my ramp script to make an interesting tree shadow. I feel like my previous attempts where uninspiring. The shapes arent as interesting as the actual shadow. The nuances of texture and colour are missing and there's nothing to take there place (ie. no added level of interest emerges from the abstraction, from the technique I have used). But I dont want to give up just yet. …

Issues with the script as it stands;
Z position doesn't quite work (it seems to make the shadow half as wide). for teh moment I will try with the UVs moved to compensate - or I could tweak the script...

General steps;
have decided to use UV placement node to alter Z pos of ramp.
I'm now wondering if I can offset on U for position of curve also?
I'm enjoying the challenge of revisiting the code but am not sure if I'm wasting my time.
How to go from excitement of personal discovery to exciting image?
Thoughts arising during production;
A general thought about my project -- its almost an argument against collaboration (lost in collaboration). Its about creating surprises for yourself, but somehow the surprises are not random; are they emergent (in the sense that something emerges/ it kicks into another gear/ becomes a new animal, no longer shades of grey).
- I could connect the attributes for zPos of placement offset...cant use connectAttr because cv position is not an attribute. cant easily constrain a locator to a CV either :
So I will leave that idea and If i want to animate the shadow i will do so in another way
- be good to add a falloff control
- be interesting to create a random variation for each input and then vary the ramp accordingly

Summary;
I’m happy with my progree with the script, “rampTextureShadow16.py”
It has the following features;
  - Z position
  - a controller for Main controllers called BLUR
  - a controller for type of Ramp
GENERALLY speaking I have learnt the necessity of defining functions; breaking my script down.
Goran spoke of ‘refactoring’.
He said;
  - dont be afraid to break the code
  - dont change things in the code if it can added as an argument

.......
19_04_13

Aims and intent at outset of work session;
Looking at ‘nightTree_007blinding’.mov - this achieves a degree of atmospheric qualities. It might be good for the BG to go black when the blinding cars go past (and I think there should be more cars, passing in quick succession) overall its too minimal. I would now like to add texture (and more models) to the scene to make it more subtle, like an eye searching around the dark landscape.
I will be looking for ways to pay attention to ‘edge qualilty’. and will be making notes about the possibilities for animation that I find.
I have decided to try using limited colours.

General steps;
start by laying down background colour (0.1,0.15,0.25).
- maybe duplicate BG colour and deviate from that...
- maybe use just a few ramp colours and mix them differently in layered shaders. I will try that as a starting point. This might replicate the situation when you paint (the way that you put colours out on the pallette and then mix them).

Processes, tools, and techniques used;

Unusual (less common) processes, tools, and techniques used;

Questions arising during production;
a computer crash. how annoying....a freeze...a frustration

Thoughts arising during production;
Summary:
not

16_04_13

General steps;
I finally got my ramp striated shadow script working well enough to use for night tree production ("rampTextureShadow8.py").
I was feeling bored and frustrated until I decided to play around with attributes of the “main” ramp (e.g. interpolation, ramp type, etc). I will render out a series of short clips. I’m interested in the unpredictable, but possibly interesting, visual results.
I want to set up the scene render flags so I have a folder for each set of images....

Opportunities arising during production;
- I just realised that I can change the driven keys that I have set to make the blur act differently; i.e. to change the relationship between the main blur cntrl cube and the local blur cubes. The relationship I have tweaked to that works is blur cntrl = 1, and others range from 1 - 0.6.

Thoughts arising during production;
- It would be good to hook up the width and width mult to a cube controller. i.e. I could probably set driven keys retrospectively.
BUT it is workable as it is so I would now like to introduce the Z axis to get wobbles on the tree.
Trying to set it up based on Z pos of CVs but I am failing. Can do Pos0 and Pos1 but cant get 2 and 3 to follow. I am now thinking to creat a cube and linking to it. I now realize that it might take another ramp or something...

15_04_13

summary;
I spent most of yesterday (15th) working on a script to generate a series of ramps. I have imagined this striated tree shadow but havent quite got far enough to try it out. I will now attempt to hook up input 2

... been trying to do my ramp shadow setup.
Wanted to make it so that the blur is dynamic; so I can adjust and animate it.
I cannot constrain a ramp input to an object.
I am having trouble creating my python function as an expression.
I managed to create a cube (named locator) for the blur amount.
and then to create a cube for each input ramp. I then set driven keys for each of these local cubes so that they are driven by the main cube.
these local cubes work quite well for input3 on each inputRamp.
My question now is whats the best way to hook up input 2 (bottom outer)??
input2 should equal 1- input3

if I use connect attr can I slip in some math nodes?
I will try to slip in a reverse math node
cube = mc.polyCube()[0]
baby = mc.polySphere()[0]
mc.xform(baby, translation = [0,4,0], worldSpace=True)
reverseNode = mc.createNode('reverse')
mc.connectAttr('{baby}.translateY'.format(baby = baby), '{reverseNode}.inputX'.format(reverseNode = reverseNode))
mc.connectAttr('{reverseNode}.outputX'.format(reverseNode = reverseNode), '{cube}.translateX'.format(cube = cube))

I did this test which sets up a reverse relationship between a cube and a sphere. It works fine will now try it in my ramp shadow script.

......

12_04_13
night Tree cont.

**Strategy/ intention;**
my ramp experiments yesterday didn't quite work as I expected. I am going to give one more go to striated ramp script. Hope to use the striations to do more with the road.

Generally, I feel today that what I am after is a 3D visual language and toolset that focuses on the phenomenology of perception; the way that we can shift our attention/focus from one object to another; the way that we “see through” objects (eg. I am seldom consciously aware of the street lights; I am just vaguely aware that I can't make out the details at the top of the tree where they are “burnt out” by the glare of a light.

2.45pm... got the ramp script working to a degree. need to add a few more controls etc to make it usable.

No I have decided to move onto another scene, one I can get some results quickly...

**Processes, tools, and techniques used;**
using NURBS curves to draw over trunk and extrude poly face along curve.

thinking where the curve cvs sit in zdepth; had the idea that it could be based on tone in the drawing.

**Unusual (less common) processes, tools, and techniques used;**
- extruding poly face along curves means that when I go back and change the number of divisions on the first extrusion, the form flips around.

I prefer the form when divisions is set to 30 (rather than 5)

- linked a couple of ramp textures in a LT to create a ball (might be an easier way without the LT?) and moved this ball through scene (reminiscent of a passing car. All geo except planes has the same shader. rendered a few movies...
Questions arising during production;
- could an animation of the tree original mesh divisions produce an effect like blinding light or some other kind of harsh perceptual phenomena?

----------well I had a go and think its somewhat successful. here are some pics; and I also rendered a movie...

Thoughts arising during production;
- DEFORM VIA HISTORY CHANGE; I like the deformation achieved on this tree trunk because it feels like the deformation is not arbitrary as it might be if I added a random element to the position of vertexes for example. In this case (by animating the number of divisions on the base extrusion) it feels like the nature of the deformation is due to the processes, techniques/tools/medium I have used.
- thought it could be cool to make the (concrete) light source first then add models into it
- idea for my antidote to paint fx, strikes me that trees are a good case study for phenomenology over objective reality. My tools for 3D users could be a kind of “spoof”.
-idea for a tree building UI; curve facilities = move pivot to position of cv[0]; ?maybe randomly rotate
-How and when does the experience change when you do something many times?
  what I mean is; while I am working out my code for creating a ramp connection according to the position and number of cvs on a nurbs curve, I start with 4 CVs. Once I get it working I will execute the code with many CVs. I would then like to see the point at which the experience changes, when its not just shades of grey, but becomes a “different animal”.
- LIGHTS, I am thinking it may be best to use lights as well as textures and projections.
Tom is using 1,2,3 Catch right now. That makes 3D models and textures them; it ostensibly “bakes in” the lighting.

Summary;
Absorbing day. first half coding. second half ‘playing’ in Maya building a scene based on a sketch. I managed not to do too much work on “auto-pilot”; I didn’t fall into the trap of “filling in”

........
11_04_13

night Tree

Strategy/ intention;
Start with the shadow. Use a ramp and vpos to change fuzziness.
I am deliberately starting with the phenomena (the shadow) instead of the object (the tree).
Chris Jones suggested I use lights, but I want to make a texture, something I can link up to recreate the visual phenomena as a texture. Why not use lights, specularity, etc? I am thinking of the shadow as an entity in itself, not a byproduct of lighting conditions (accidental qualities), and this leads me to make the shadow directly; before the tree and independent of the tree. I think the shadow will end up being stylized in a way that emerges through the tools and the processes.
...I’m also wondering if the shading network itself might end up looking like a tree.

........
10_04_13
Busy Intersection
Processes, tools, and techniques used;
start with camera BG colour?
set viewport colour to the same (maybe I can link these?) - making a python script to do this....

Summary;
I decided to create a scene from my ride home the night before. I sat down at the computer to start making; I was trying to recall the colours which were largely muted and subtle. Immediately I wanted to “lay down the background colour” as I might when starting a painting.
I didn’t actually end up building any objects on this day of work, but I enjoyed the process of writing a couple of scripts to help lay down a background gradient that is easily adjustable (via a ramp texture node) and is linked to the colour of image planes on the renderable cameras.

09_04_13
Thoughts arising during production;
I feel I need to simplify, look at the painting and simplify in a way that draws out the characteristics of the image. Don’t simplify along obvious lines (eg, foreground/background...). From this simplification I can then let the complexity emerge again.

Summary;
Not enough time and not enough surprises today for the process to be engaging.
But it has made me think that I’d like to go back to the source, ie make observational sketches or notes to work from. I’d like to engage with the visual phenomena and take notes with 3D software in mind.
Processes / Tools/ Techniques used;
working from the 2D reference, but I am adjusting the form of the car to make it look “right” from all angles. Do I have to work like this? To what degree should I make it look good in all viewports, and to what degree can I leave it imperfect, or even not making sense?

Less common (or unusual), techniques /ways of working;
modelling and texturing at the same time, thinking of the 3D form as a canvas for colours. Avoiding the idea that the form has to be correct or precise before textures/ colours are applied. NB of course it is not entirely uncommon practice to change the model after applying textures, but usually this isn't scene as ideal (you have to reorganise UVs etc).

Questions arising during production;
What happens when the form doesn't correspond to the colour/light? when the form depicted is largely achieved via shading/colour; or when the form depicted (represented) is at odds with the virtual 3D form...
Technical -- shape node / attribute editor > Smooth Mesh > Extra Controls > Boundary Rules = “Legacy”...do this to change so that the edges of a mesh are smoothed.

Thoughts arising during production;
RULES
do I need to make working rules for myself in order to avoid habitual working methods?
eg, can only spend 15 mins on a model before applying colour?
FILLING IN
I am reminded of warnings from my art school teachers about “filling in”. I guess this warning is against slipping into habitual ways of working. continuing to approach the work afresh (what do I mean by this?) can be a way of not filling in. I’m building the car now, and I’m filling in (it feels like, i.e. its boring and I want to get it done so that I can do the next bit). perhaps if I look at the reference again and start with the bits I find interesting...maybe the uninteresting parts of the image/the model/ the process, will take care of themselves.
SWAPPING MODELS
The car is now blocked in, should I keep a copy of the LR model (lots of iterations perhaps) for possible use in an animation? or is it the same to use the “reduce” tool (my hunch is its not the same).
NAMING THINGS
part of using 3D software is naming things. The user can name objects or “nodes” as they are created, or the software will create automatic names. I usually use a combination (name objects myself, and leave other objects with default name).

The software needs things to have unique names (it uses namespaces to do this I think).

As naming is already part of the process, I’m thinking of scripting opportunities that may arise from this;

- How could a script use default names? eg. utilising the time it takes to model or the sequence in which things are made; if a create a series of objects which each starts from a polycube and then apply a script to the series “pCube1”, “pCube2” etc, I could use the sequential numbers to drive something (eg, the deformation of the model or the colour of the model). Of course this would only take into account the sequence, not the time taken. Maya does have an internal clock so maybe I could also harness the time taken.

- Could I develop a naming strategy (convention) to use as I work so that I name things according to how I want my scripts to effect them?

Summary

End of day image - I now need to focus on achieving the sense of light. Perhaps even before blocking in the background.

Thoughts for tomorrow;

don’t be afraid to digress from the reference pic
maybe keeping it simple is the answer
think about process (tools/ techniques) and movement / animation

06_04_13

I want to move away from 3D as simulation. I’m using a starting (reference) image (one or more) to make this work but the reference is just a guide; a starting point. The idea is that the production process itself will have an impact on the final image. So in this case of the Petrol Station, the reproduction of one of my old paintings is a starting point. I will not be trying to replicate the painting as a whole (eg the texture of the canvas and the daubs of paint ), rather I will be aiming to re-present certain qualities of the image (eg. the colours; the “atmosphere”, or the quality of the light). So that the final work is not a mere copy of lesser worth than the original painting, my hunch is that I need to follow up unforeseen opportunities that arise through the 3D software production processes.

05_04_13

Starting a scene based on an old oil painting of a petrol station at night. will leave the figure out for now.

I will begin by using processes/tools/techniques that I have used in production (Denso, Honda). But this time I am open to unforeseen opportunities. By being open to unforeseen opportunities that arise via the production process
(using UVs, ramps, shaders, polygon models, stylus input) can I make something that is as interesting, or more interesting than the original painting? or interesting but different/ interesting in a different way?

Will the process be more engaging if I am open to surprise opportunities? Will I achieve more periods of flow state?

I tried using “crater” texture but it doesn’t look any good. I think its kind of the wrong metaphor (its trying to be organic; trying to replicate the canvas texture. I think I need to be more willing to translate into this new medium; let the old look (of the painting) go if and where and when it doesn’t feel right.

Reference Pic

Pic from end of day 1 working in Maya

Processes / Tools/ Techniques used;

Much of this process fits with the “normal” (common) 3D workflow;

Start with models as close as possible to the centre of the world.
duplicating objects
starting with grey models
using quades where its easy / possible
naming objects (transform nodes)
creating surface shaders, assign ramps as colour inputs.
use the colour picker to pick colours from the reference image
Projecting textures / using UVs
Grouping Objects
Use display layers to change the display of objects on a regular basis (so that I can see the image behind the objects for example, I have a display layer set to no texturing).

considering objects in turn (focusing on UVs and texture/colour for one object, one part of an object). However I am not considering lots of different attributes of an object such as "specularity", reflection etc. It is all colour to me.

Others are less common (or unusual) ways of working;
I am resisting the urge to look up reference for various objects (eg. what does a bowser pump look like?). I guess I am working from memory (my memory and intuitive knowledge of what a bowser pump must be shaped like).

Questions arising during production;
what would be the benefit of using one swatch (ramp) per colour? so that ramps feed into ramps?
How to work with the software? what does it do/ whats its logic when it lays out the UVs ?automatically”?
As colours are ostensibly assigned to faces, can I change lighting by altering the model/geometry?
what happens visually when colours dont align with the geometry? ie when the “baked in lighting” doesn't quite make sense with the geometry?
What is gained by making it “3 dimensional” (ie by having the computer (the drawing tool) store the graphical information in 3 dimsions? eg. you can move the camera, turn objects and change thereby potentially changing their colour and profile shape.
How much should I “fill in the gaps”; what is “actually” going on? what is the true form of that object that I cant see?
Maybe its good to roughly map out objects first; rough out many before refining?

Thoughts arising during production;
ramps can be sharp, or blurry. This reminds me of my artschool training where we were encouraged to focus on “edge quality”.
could use naming of geo to dictate how a script processes it; assign colour/ shader, layout UVs.
Maybe I should keep the camera still and use other aspects of the virtual 3D data.
- RAMPs - colours as numbers - threshold - if the numbers are above or below a certain level, do “this”

04_04_13
I could dive in and use the techniques I have used before; I have an idea in mind of what I want to make and the processes (tools and techniques) I want to use.
I could start with an image such as a past painting and recreate that using these techniques I have in mind.
But I also want to do my programming homework.
As I play around with code I am getting ideas for future projects, things I want to make;
eg. the randomized tranform allocation makes me think of swapping models as a way of animating. The “swap” could be random or it could have logic to it; it could be linked to another attribute (proximaty of camera, sound, or colour). The randomize colour function I wrote makes me think about colour in a different way (hueMin and hueMax when close together produce grey, numbers for R G abd B dictate how warm or cool). these functionalities remind me of using ultramarine blue and burnt sienna to make grey and then thinking in terms of adding warmth and “coolth” to that base grey, to “push” and “pull” the canvas.
It feels important to me to be discovering. I’m not excited by the idea of illustrating a design that I have already made or already decided upon.

import maya.cmds as mc
import random

def assignRandomColour (minHue = 0, maxHue = 1, channels = [0, 0, 1]):
assign a surface shader of random colour to each mesh
you can specify how much of each colour by entering a value from 0-1 in the corresponding channel
--how to make it update as you move a locator or change value in the viewport?
--I could use a math function (eg "cieling" to clam between 0 and 1)

# Make sure that channels is a list
if not isinstance(channels, list):
    channels = [channels]

# Get selected meshes
selectedMeshes = mc.ls(selection = True, dag = True, type = 'mesh', long = True)
# if no meshes selected use all meshes in the scene
if not selectedMeshes:
    selectedMeshes = mc.ls(geometry = True, dag = True, long = True)

# Error if no mesh selected
if not selectedMeshes:
    raise RuntimeError, 'You must have geometry in the scene to use this function'

# for mesh in selectedMeshes:
# generate a random number (one per mesh)

# create a surface shader
shader = mc.shadingNode('surfaceShader', asShader=True)
# create a ramp
ramp = mc.shadingNode("ramp",asTexture=True)
# a shading group
shadingGroup = mc.sets(renderable=True,noSurfaceShader=True,empty=True)
# connect shader to sg surface shader
mc.connectAttr('{shader}.outColor'.format(shader = shader) , '{shadingGroup}.surfaceShader'.format(shadingGroup = shadingGroup))
# connect ramp texture node to shader's color
mc.connectAttr('{ramp}.outColor'.format(ramp=ramp) , '{shader}.outColor'.format(shader=shader))
# assign mesh to SG
mc.sets(mesh, forceElement = shadingGroup)

colourR = channels[0] * (random.uniform(minHue,maxHue))
colourG = channels[1] * (random.uniform(minHue,maxHue))
colourB = channels[2] * (random.uniform(minHue,maxHue))

# set ramp colour
mc.setAttr('{ramp}.colorEntryList[0].color'.format(ramp=ramp) , colourR,colourG,colourB)

#assignRandomColour()
User specifies min and max hue (if these numbers are close together then colours tend toward grey), and a number tween 0 and 1 for each colour channel (so the hues can lean toward red, green or blue depending on this input)..
Modelling as Animation
Production notes

read from the bottom upward

25_05_13
Aims and intent at outset of work session;
General steps;
Processes, tools, and techniques used;
Unusual (less common) processes, tools, and techniques used;
Problems arising during production;
Welcome surprises arising during production;
Questions arising during production;
Thoughts arising during production;
Summary;

27_05_13
Questions;
mc.modelEditor('modelPanel4', edit=True, joints=False)
how to make a toggle switch for showing joints? I need to get the model editor’s name

Rigging
I am thinking how the rig; the choice of controls, maybe their names and colours, could contribute to the final animation / movement

Thoughts arising during production;
I am starting to get bogged down. sidetracked, and at times bored.
I am thinking about rigging and animation right now. From an initial desire/plan to model as animation my dog from sketches I have of her (most of which are lying down), I moved on to issues to do with rigging. For example I want to make a dog rig that is flexible so that I can move joints anywhere; so that I have a great flexibility and I can achieve any pose. But also so that
- looking forward and listening... affordances... animation techniques … illustrating ideas... How to move beyond these things? how to create something out of nothing? Who’s ideas could contribute to this question?

?Deleuze - cliche and chaos

23_05_13
Aims and intent at outset of work session;
- arrange drawings in viewport and set keyframes

General steps;
- in order to work through the timeline and get a sense of the animation, I am creating a few key models this time around.

Problems arising during production;
- looking through different cameras is a pain (trying to align the sketches)
  possible solution is to just work with the sketches to the side
- I cant get a sense of the movement without building different models for different poses.
  solution one is to make dummy objects for joints and animate between poses.
  solution two is to create models first for each pose. make rig and bind each model at the correct stage of movement. I could then save and bind each model (moving forward and back). The question is then HOW and WHEN do I transition from one model to another?
I think I will proceed this way and see how the models combing when I come to it.

**Questions arising during production;**
- should the drawings all be at the centre of the world overlapping the model? they might get in the way there
- RIGGING - “For a dog you are going to want to have the actual COG rig node to be movable, along the spine.
- then you can rig the hips/chest just like you would a regular spline ik spine.
- There are a few ways to do "animated" pivot nodes for controllers.

**22_05_13**

**Aims and intent at outset of work session;**
Start a longer ModellingAsAnimation test. This one is called ‘dog study’ or study of a dog, or studies of a dog.

**General steps;**
- going through drawings of my dog (fragments unfinished, different poses)
- choosing some that I can sense might form an animation (i.e. I can sense how she might move from one pose to the other).
- most sketches have more than one sketch on a page. I am duplicating the reference planes moving the UVs to put the images on different planes.
- the sketches are all from different angles so I will add many cameras this time.
- I will look for opportunities to introduce detail, and opportunities to loose detail. eg. when dog opens its mouth
- COLOUR; I’m wondering if I should introduce colour or tone into the models (I am imagining overlapping models, transparency, RGB values...).

**Problems arising during production;**
- import a scene and hate the way it puts letters. Make a batch renamer....

**21_05_13**

**Aims and intent at outset of work session;**
- do nodeInfo tute from the net [http://www.rtrowbridge.com/blog/2008/12/maya-python-api-getting-mesh-data/](http://www.rtrowbridge.com/blog/2008/12/maya-python-api-getting-mesh-data/) and then compare with PlugIn course node Info tutorial (which is done in C++).
- Despite my plans laid down yesterday (to continue with a dog ModellingAsAnimation), I think I will spend today investigating Maya plugins and more specifically the Maya Python API. It will be an interrupted day of working and so this hard slog rational enquiry might be more suited.

**Why do I want to work with the API?**
- I have a hunch that it might be important for my project that I work with mayas API. I wont be using C++ which is the language that the API is written in as I understand it. But I will be using Python which is the next best thing and will hopefully still give me an understanding of Maya’s architecture, the components that comprise it; and also an understanding of how those components (Maya objects or classes) where (and are/continue to be) created.
- At this point I may want to bring Douglass Rushkoff to the table and his book *Program or Be Programmed.*
Some questions I currently have are;
- my programming attempts so far have been successful in opening up opportunities to create new workflows (by automating tasks that would otherwise be labourious and unworkable; by this I mean that they would be too long-winded to allow the artist (myself) to achieve a state of flow).
- How will a ‘hands on’ understanding of the architecture of a 3D software package such as Maya change the way that I interact with the software?
- can I continue my “modelling as animation” tests and also gain a deeper understanding of how Maya was made? would it be useful to also read about how paint is made? how pigment is discovered?
  Often by looking at a painting or a drawing a viewer can see how it was made. Even if they are not a painter or a drawer they will often have a sense of how someone else interacted with the tools. I noticed at the Drawing Out conference that the air of mystery surrounding coding seems to be at the core of its nature. What would Douglas Rushkoff say about this?
  I am also wondering if there is a difference between ‘knowing’ how programmes are made (by studying the theory of programming for example) and actually doing some programming. Does it make sense to think that there’s ways of knowing something from the inside, and other ways of knowing something from the outside? what would Cain or Varela say about this idea?

20_05_13
Aims and intent at outset of work session;
- thinking of a longer dog animation, or a longer bird animation; maybe with more than one bird (two for example). was feeling quite dark today and wondered if my animation could tap into my mood?
- will try for 1 hour to make script that sets keyFrames on materials according to vis keyframes.

MODELLING AS ANIMATION - next (longer) test
I am thinking of do a dog animation, maybe lying in bed (as part of the bed) then getting up.
RIG
i've rigged one quadraped recently (broken hierarchy).. and i'll tell u how i did it.. might help..
for the hind legs, and fore legs, we can create individual IK chains and you can have IK controls for each of ur legs, so u can move them independently...
and u build ur spine individually. and u can get a stretchy/squashy spine setup
(http://forums.cgsociety.org/showpost.php?p=3186832&postcount=4) real quick.. pls c that page , and u can do this same for ur tail and neck too.. its like u have individual parts which are made to stretch and squash and then u bind each of this parts via parent contraints , to have a connected structure.

Questions for this animation;
- what should be animated by joints and what by modelling?
- This is a subject matter I am very close to; how does that change my approach/ my expectations/ my outcome (eg am I more focused on a specific outcome?
- I would like the dog to be part of other objects at times; the bed? another dog? cone?
- do I really need to work out the full range of movement; and hence the extent of the rig; before commencing animation?
- do I need to complete the animation before continuing with the modelling? I suspect not and I’m going to test this hunch by just “diving into” the making process tomorrow. starting with a favourite drawing and continuing (forward as well as backward) from there.
  ---- in the back of my mind I may have a paper/ point/ exploration/ conversation/ in mind.

.......
17_05_13
Aims and intent at outset of work session;
- start to develop an understanding of OOP and classes
- try plugin tutorials.

Steps;
I am making a cube as perceived after coffee
   I can create a new class 'ball' that inherits from box. The functions work if the attributes are the same.

I am now writing a Hello world plugin with args;
he looks up the devkit for reference.

Problems arising during production;
a quick look shows that the plugin tutes are in C++ so I would need to convert them to Python;
   hence a deeper understanding of Python classes
   and a deeper understanding of the Maya Python API would probably be useful.

Welcome surprises arising during production;

Questions arising during production;

Thoughts arising during production;
- making custom 'objects' which are based on perception or context

Summary;

15_05_13
Aims and intent at outset of work session;
- finish UI and do a quick test of the workflow.
   - what it like to use? boring? fun? too prescribed? or allows me to go further?

   - when doing charcoal or plasticine animation the speed at which you work has an effect; you can seem to
     embody the scene/character/movement. Can these tools achieve a similar outcome?

   - would use of this tool change the way you model? so that the process of modelling becomes
     “expressive”, or at least becomes part of the final animation?

   - answer the question; what's the difference between doing the programming myself and getting a proper
     programmer to make something from my design/instructions?

Welcome surprises arising during production;
   You model according to the sketch with a focus on the camera view, but then by working in the perspective
   viewport you see get a sense of the character emerging and start to 'make up the model as you go along'. So you
   start with a particular form (or shape) in mind and then move on from there.

   when you advance forward in the animation, the original mesh has history on it and it messes up when the
   joints deform to a certain degree. You might be able to fix this by deleting history and rebinding???

......
Aims and intent at outset of work session;
get the UI as good as possible with the current design.
    errors re parenting
    Layout on tab 3
    add instructions

Do a quick test of the UI design
    starting with one model, changing it to two etc.
    rewrite instructions if necessary

Problems arising during production;
trying to add tools for keyframing but getting this error;
    # Error: UnboundLocalError: local variable 'activeControlSet' referenced before assignment #
    I worked out that because I wasn't querying the set name properly it didn't have anything selected and so
    nothing assigned to the return variable 'animControlSet'

Summary;
absorbing and good progress. script becoming a bit long and unwieldy. FOR TOMMORROW;
    add keyframe vis button (make a function to set key on when off and off when on)
    -- keyspacing
    -- keep/ delete history.
    error for when no control sets are selected

........
14_05_13
Aims and intent at outset of work session;
add radio buttons;
duplicate, bind, and hide model
    - maybe I should do this in a separate functions

Problems arising during production;
- I am going to gather heaps of "TWEAK SETS"
    I need to populate my list of sets differently
- When I rebuild the UI its starting with the setup tab
    if I don't find a way to fix this then reverse the order of the tabs

Thoughts;
I would like to also have shading options for the models

Summary;
I seem to have the duplicate, set keyframes, and bind models all working.
now I have to work out how to;
    populate the sets list and also
    have tab 3 active
    make the layout prettier

........
09_05_13 and 10_05_13
Aims and intent at outset of work session;
I’m wondering if I should get something very basic and simple made so that I at least have a bare minimum for next Wednesday. so perhaps I leave out any rigging stuff. Should it be guided “how to?”

General steps;
I am doing magpie test. I will write the steps and add some of these descriptions as optional notes for users;
- create a cube
- position camera according to cube (not the other way around)
- adjust playback settings, timeline length, set to 25fps
- set keyframes on visibility of ref pics
- choose favourite ref pic by moving to a point on the timeline
- create the simplest model possible to describe/capture/indicate the form or the character; work mainly in the perspective view keeping an eye on the camera view. refer to reference plane and also original sketch if possible.
- delete the history and name the model
- create rig/ skeleton - could use a preset rig but I lik eto keep it simple. Ultimately I would like to find a way of rigging that doesn’t use joints. I wonder what effect it would have on the user if they concieved of joints as pivot points rather than as a metaphor for a skeleton/ a set of bones.
- create set for easy selection of the controllers
- animate the model (Gina, what if the model needs to change to do the animation properly?)
- use the save model and set keyframe button

Particular qualities of my magpie sketch;
the linework is an attempt to capture the silouette of something moving fast. I cant tell if the bird is facing toward me or away.

To add to the toolset;
--optional; include a dropdown modelling toolset (maybe as a separate function or template?)
- move tools switch between world, object and normal.
- would be good to have input for the vis duration. I will default at 5 frames
- select all joints option plus user define joint set. as a activated button. add joint set // joint set from all joints

Problems arising during production;
- the magpie is like a biped and requires a foot setup. when I add an IK handle I cant stretch the bone---I just remembered the stretchy IK stuff I found before
- I followed Zeth Willie’s tute but found that using the distance between node from the top joint gives an incorrect distance. I just realised that it might be because I created the root joint second.

Questions arising during production;
- will the tool work for starting model in pieces and starting model as one?
- should models have transformations frozen before binding?
- what would be the equivalent of ik handles if I didn’t use joints? ie. how could I achive reverse kinematics?
- what are the implications of not creating a symetrical rig? what are the problems and what are the possible opportunities?

Thoughts arising during production;
I was just fiddling with colours for the tab backgrounds. This is done via numbers of course (which define the RGB values) and I was thinking that tools which allow artists to define the relationship between things (eg between R,G and B, or between colours, or between colour and speed) could be interesting. So rather than tools which urge the artist to define what something is, they would be tools that are about how things relate.

Aims and intent at outset of work session;
- further work on Modelling As Animation UI
- consider the template or OOP examples; are these two things the same?

General thoughts and questions arising during production;
am I wanting the UI to act as a set of instructions as well?

Summary;
- I spent a lot of time going over OOP;
  in the Python book,
  vids
  movies
  Template from Shaun
I now feel half asleep. Its time for a cup of tea and then back to the “task at hand”.
FOR NOW I’m not going to use classes, modules and methods.

Aims and intent at outset of work session;
In order to design the interface I need I will go through the process again take notes.
I may also do Python tests along the way; seeing what types of actions can be scripted and using alternative commands if neccessary (I dont mind the idea of the process dictating the form to some degre; as long as I am destroying connections/categories and then reforming them).

General steps;
- Make two magpie pictures by cleaning sketches and cropping the images
- put the pics in the source images folder
- open Maya
- create camera with planes and sketches

Summary;
see technical notes for more info
https://docs.google.com/a/student.rmit.edu.au/document/d/1jKRTzl04jLFORwgLIL4qFkpXCSNQQhu76glI0-__U/edit

I created a button that creates a camera, then checks the source images folder and creates a NURBS plane and a lambert shader for each image. Planes are parented under the camera.
-- COULD IMPLEMENT;
  a keyframe on the visibility of each plane
  move the camera back on the Z axis
  create a cube
  call the button ‘setup’ and add a warning about setting the project snd hving imsges in the sourceimages/refPics folder
Aims and intent at outset of work session;
- test how to rig a figure without using joints. This includes;
  pivot points to deform a mesh (i.e. to move vertexes in space by rotating them around a point)
  controllers to animate
Can I develop a flexible rig? where the animator only adds joints and controllers as needed? Why? because I want to avoid starting with 'it all on the table'. When I sketch I move the pencil across the page according to differentiations I perceive in the subject/scene I am observing. The page starts blank (or white) and it is carved up according to these differences I see.

Processes, tools, and techniques used;
- setup Sublime text with debugging functionality
- watching video tutorials on OOP taking notes on my Python Notes page (Google Drive).
  I did some tutes on functions, methods, and classes; maybe I should use this way to build a window (make my window class then create an instance of that class)
OR can I think of my polycubes (that I am modelling) as instances of my own class?
  eg can I give them their own attributes based on the frame number they are created on?

General thoughts and questions arising during production;
- I am using Python to make my modellingAsAnimation tools so I am now wondering if I can use the essential qualities of OOP to design my tools?
- classes; can I have a story emerge from my learning of OOP (classes etc)

Summary;
- learnt a bit about OOP,
- did several short tutorials about Modules, methods and functions.
- did the first of ten UI tutorials; learnt to populate an option menu; add an image banner
NEXT TIME;
perhaps I should just do a quick UI for duplicating a model as I did and use this as a starting point to see what functionality I need.
Reflection after my first attempt at plein air 3D;

GENERAL OBSERVATION
- this can be considered part of the “Colour as Light” pieces. However unlike most of those pieces which were created away from the subject matter, for this piece I wasn’t interested in exploring the tools (eg various shading nodes etc), I just wanted to “get it down”. There was a point where I delighted in various shapes and shades that I discovered as my eye moved across the forms, and I wanted to record these shapes and shades quickly.
- for this piece I was limited to 10 colours. It felt like not quite enough but perhaps it was good to be limited
- as the light changed I wanted to change the topology of the model (eg move verts of shadow faces upward)...
These topology changes were reflected or captured with the duplicated mesh…. however the colour changes where not reflected…
- have just played thru the 278 frames. Its interesting as a record of how teh model was made…. however I realise that I was hoping that, by recording my making process (the progression of constructing the image), the result would capture something similar to the process of looking/ of processing / of understanding.
Does it fail to do this because the model is cumulative… ie it goes from less detail to more detail… its not as though the detail is dynamic in more than one direction...This could perhaps be remedied later in teh studio… maybe this is where experimentation with the software’s “materiality” comes in.
---- yesterday I was obsessed with link my selection set to SG set. now I dont really like that idea anyway because I enjoyed the flexibility as is (especially as …
- I just noticed that my models have been duped without history !!!!!!!!!!-------
perhaps make this an option in the script

TOOL and SCRIPT AMENDMENTS and QUESTIONS
- animation settings to 25 frames per second
- ability to delete a selection set and its UI.
- set BG viewport colour, Maybe this sets the default colour of the swatch??
- ability to add swatch would be good
--- could I create a new tab with a new colour swatch and new selection sets
--duplicating the model and keeping history - this could be difficult cause I need to dupe input graph which
probably dupes the shaders and the selection sets. So faces from new set have to be fed to existing set and
shaders reassigned and new ones deleted...

--- script makes ramp swatches which feed into layered shader. when the colour of a swatch is changed (or when
I press a button) a new swatch is created and feeds into the same shader (via a ramp or a Layered texture).

--- maybe automatic duplication of the model would be good
--- could it occur when a certain number of topology changes are made?
--- and when colour changes are made (ie colours assigned to new faces)?

Maya and Wacom
- need easy access to the “split polygon” command
- disable zoom
- download latest drivers and see if that remedies the slight lag
---- try a cintiq?

POST stuff TO TRY
- try linking various history on nodes to a controller and animating this control
- try animating textures (eg projection texture, or animating a ramp / swatch)
- doesnt make sense on teh building, but try deforming all objects (via non-linear, joints, transform component etc)
- what attributes can I change with “transfer attributes”? ie does this bring the model “closer together”.
- try overlapping the models and using transparency (for this it would be good to have a script to work with vis
keyframes of each model).
- try moving the order around … does this really have to be manual?? remember that the models are named
sequentially.
---I lost the blinding glare that I saw before (ie the faces somehow got deselected). I could do something like
this with a ramp. it might entail me selecting the same face on many models… can have a script and UI that
selects particular faces (select some faces and then Maya extends this selection to all the models of same name).
- try a moving camera…camera could zoom in on window area…

IDEA - in post scenes I rename models with an “_” between name and number. This shows that teh models have
been worked on away from scene (ie “back in the studio”)

Useful scripts for post work
- select faces based on selection OR based on bounding box
  I already have a script that selects verts based on bounding box
- select all models
- move vis keys
- select all history nodes of type X on all models

BIG QUESTION is how could I use bits of my various models?
ie, how could I combine an area of detail from a late version with a low res version from a previos version?
This has always been a big question…
A. could I combine point info from one model to another? could I somehow use group parts nodes?
   YES: at least it seems that this works in “deleting” or bypassing geo of a certain colour when I go from one
   GP node directly to “input Geometry” of another. I imagine it would be possible to animate this effect with maya
   nodes ??? or with a purpose built UI
I tried to then create a new cube and “hijack” its shape node to feed both into and create a new Hybrid shape. I couldn't work out how... perhaps all the connections were’net made.

----------I could try giving the new cube its own GP node (so that maya makes the connections) and then changing its id???

A - I could do so in the composite; revealing one rendered image through another
A - I could do so with transparency

found a movie tutorial on how to use the API to create polygons. You provide MPoints and a bunch of other info. Seems its not far from this to make a node so that the position of the vertexes updates all the time (I assume this is done via a plugin)
Now back to building project;

Making tools for post work and updating plein air tools…

Attribute UI:
Format for setAttr when type is a component list??? here it is; (more in technicalNotes folder)
```
mc.setAttr('polyMergeVert25.inputComponents', 2, 'vtx[21]', 'vtx[36]', type='componentList')
```

Above pick shows results of setting the .inputComponents of MergeVerts node…
can I input my BB object to this?
do I need the groupParts node inbetween?

setAttr() - I want to have a way to store the original Value

connectAttr()
- ability to select history nodes of type: polyExtrude, polyCutPlane, polyMergeVerts and connect sets of them to sliders?

Can I retrospectively add a polyMergeVerts to each house model?? → YES

COMBINING MODELS by bypassing groupParts nodes
A separate question is how to get the groupParts nodes attached to a particular object cause they dont always
show...hmmm … seems when I add a nodeSuch as merge verts to the model then all the GP nodes show...

above is building 15, building 56, then both. In this case the combo is not very interesting but perhaps on a
character model the results might be more interesting

………………

SELECT COMPONENTS ON ALL MODELS VIA BOUNDING BOX
So I like the idea of ability to **select faces (to texture) or verts (to move or transform) based on bounding box**

21_05_14

about two hours of observing and modelling;
I was focused on capturing/ describing my dog as she sat in teh sun on the front veranda. Sometimes she stood up, wanting to go back inside but the door was closed so eventually she would sit back down again.

Tools == autoProcedure with duplicate model, laptop, Wacom, chair, camping table
Process == I ended up working on two separate models; one of the whippet standing and one of her seated

Post opportunities
- ?colour the models
- deform them with history (gina there’s only history on teh original model; however adding a group parts node means that component deformations can be added afterwards)
- deform them via “rigged” joints. NB. I assume that this would probably entail traditional keyframe animation… but could it be some kind of procedural or autoAnimation (eg music driven) ??

23_05_14

plant still life
remains to be done….. I got sidetracked making the code / tool work ...

AIM
- to explore 3d modelling and texturing the plant as a way of understanding / exploring it.
- to answer the question of whether this process of modelling / understanding can be captured… and if so does it
resemble aspects of visual perception?

Script questions and wishlist
- for greater post flexibility, attach ramp colour swatches to the surface shaders
- ability to delete selection sets
- add new swatch tab
- more feedback on autoProcedure so I can see if it fails

Combine auto and manual procedure
- can autoProcedure work with colour swatch stuff or will it try to duplicate the colour stuff (shaders etc)?
---- I would like to try to capture colour tweaks of a swatch --- perhaps by adding a new colour input to a ramp

DISCOVERIES
- I have just realised that I can duplicate the shape node rather than the transform node :)
I can also set keyframes on the shape node.

General Thoughts / Questions
- how (if at all) does the practice of plein air or still life 3D relate to "performativity" as discussed by Buttler, Bolt, Barret, John Law etc?
In Laws', "Seeing like a Survey", he says that methods in science can be thought of as techniques for describing reality OR as practices that do not simply describe realities but that also enact these into being (ie. they describe AND enact... maybe the point is that these come in tandem

-------------------
I have decided today to use this code in manual mode to do plant still life.
AIM is to see what (if any) new opportunities come from duplicating shape node only instead of transform as well; and what (if any) opportunities are lost.

shape nodes and skin clusters:
--- it seems that if I select the transform and then the joints and bind Maya binds the first 3 shape nodes.
is this a setting or is there a limit to the shape nodes being bound?
...and deformers such as bend:
seems I can manually (or by script) connect the outGeometry of a bend deformer to multiple shape nodes. eg;
mc.connectAttr('bend3.outputGeometry[0]', 'mesh_10.inMesh')
mc.connectAttr('bend3.outputGeometry[0]', 'mesh_9.inMesh')

EDITING THE geo of original shape node:

TEXTURING the duplicated shape node:
do I have to do this on a face by face basis?
getting the shading grp from the mesh using the api;
Given the DagPath for any Mesh object, listing the Shading Groups connected to it is easy. The
MFnMesh::getConnectedSetsAndMembers()

24_05_14

def applyShadingInfoToMesh(mesh1, mesh2):
    # Create a selection list object and add the node
    selectionList = om.MSelectionList()
    ...
selectionList.add(mesh1)

# return the MDagPath of only element 0 in the active selection list
dagPathToMesh = om.MDagPath()
selectionList.getDagPath(0, dagPathToMesh)

# make sure the path is to the shape node (not the transform)
dagPathToMesh.extendToShape()

# finally create the mesh function set
mFnMesh = om.MFnMesh(dagPathToMesh)

shaders = om.MObjectArray()
indices = om.MIntArray()
print indices

mFnMesh.getConnectedShaders(0, shaders, indices)

noOfShaders = shaders.length()

shaderNodeList = []
index = 0
for shader in range(noOfShaders):
    shader = shaders[index]

    tempSelectionList = om.MSelectionList()
    tempSelectionList.add(shader)
    # meshObject = om.MObject()
    shaderMObject = om.MObject()
    tempSelectionList.getDependNode(0, shaderMObject)

    shaderNode = getNodeName(shaderMObject)
    shaderNodeList.append(shaderNode)

    index+=1
print shaderNodeList

faceNumber = 0
for shaderIndex in indices:
    currentFace = '{mesh2}.f[{faceNumber}]'.format(mesh2=mesh2, faceNumber=faceNumber)

    currentShader = shaderNodeList[shaderIndex]
    print currentFace, currentShader
    mc.sets(currentFace, edit=True, forceElement=currentShader)

    faceNumber += 1

def getPointInfoFromMesh(mesh1):
    #### get the point info from the first mesh
    selectionList = om.MSelectionList()
    selectionList.add(mesh1)

    # return the MDagPath of only element 0 in the active selection list
dagPathToMesh = om.MDagPath()
    selectionList.getDagPath(0, dagPathToMesh)

    # make sure the path is to the shape node (not the transform)
dagPathToMesh.extendToShape()

    # finally create the mesh function set
    mFnMesh = om.MFnMesh(dagPathToMesh)
# create point array to store the points
inMeshPointArray = om.MPointArray()
mFnMesh.getPoints(inMeshPointArray, om.MSpace.kWorld)
return inMeshPointArray

def applyPointInfoToMesh(mesh1, mesh2):
    # get mesh1 point info
    pointArray = getPointInfoFromMesh(mesh1)

    ### make a function set etc for mesh2
    selectionList = om.MSelectionList()
    selectionList.add(mesh2)
    # # return the MDagPath of only element 0 in the active selection list
    dagPathToMesh = om.MDagPath()
    selectionList.getDagPath(0, dagPathToMesh)

    # finally create the mesh function set
    mFnSecondMesh = om.MFnMesh(dagPathToMesh)

    # apply point info to mesh2
    mFnSecondMesh.setPoints(pointArray, om.MSpace.kWorld)

In the end I didnt need to use the 3 scripts above… because I found that a lot of issues (such as texture info not
being replicated on the new shape node) were overcome when I parented all the new mesh objects under a new
transform.
Currently this transform is refered to by name. I now realise that it would be better to refer to it by pointer so that I
could have different ones… or just make it reference the meshName…

--- wishlist
auto option
apply deformer to all mesh nodes (of same name)
make one mesh node transform toward the next
add dropdown with existing mesh names?
try adding blend shapes so that one meshNode deforms toward the next

…………………………………………………………………………………………………….
27_05_14
Trying plant still life again. I hope I dont have too much of a preconcieved idea of what this experiment (this
encounter) should look like… I have imagined the results several time now.

3D still life GENERAL QUESTIONS
Would it be good to work with teh subject matter (in this case the plant) from different angles? or under different
lighting conditions; or looking at it from different distances?

PLEIN AIR script ammendments
- set BG colour (toggle)
- delete selection sets
- update UI when it is remade with surface shader swatches
- put Post stuff on a different tab
  select all shape nodes
- make blend shapes work by selecting node (again)
  - make blend shape tangents straight
  ____SET split poly, merge verts, keyboard shortcuts

GENERAL OBSERVATIONS while making
- resisting the urge to make sense...ie resisting the urge to ‘wrap it up’, to simplify, to make the model look obviously ‘plant like’; by this I mean that I resisted extruding a stem from the base and then leaves from the stem/trunk. I tried to sit with it being chaotic.

After preffering the version with occlusion on it, I now prefer the flat version....
I would like to now go further and keep working on the plant… what happens if I make some areas very detailed...

SCRIPT IDEAS
I think the latest script is “duplicateShapeFunction_007.py”
- maybe I should add more swatches to the pallette ....
- the blend shape could be added only if the topology doesn't change

30_05_14
continuing on plant still life...

THOUGHTS
what makes this different from yr average timelapse modelling movie?
- the fact that in practicing this technique I actually work (ie model and texture) differently... just like when doing a charcoal animation you might work differently than you would when just doing a charcoal drawing...
--- the differences I have percieved so far are that I am trying not to order the plant... I am trying to sit with disorder… to allow the form of the plant to dictate how I approach it... rather than to approach the plant in terms of how I “know “ it to be (ie to have trunk/ stem with leaves extruded off etc)...
There is the constant desire to “clean up” and order the model... to make sense of it as a whole.... but I am reminded of MP's explanation of Cezanne as painting teh world as it comes into being and also to teh Zen drawing book which talks about drawing the particular horse (not “horses in general”).
_ Im not neccessarily taking the EASY PATH/ the EASY OPTION in terms of making a model...
_____today I was CHANGING COLOURS a lot on the plant… as I defined areas more and as I saw different colours / as the threshold from one hue to another seemed to be crossed. Instead of

TOOLS
done - toggle use default material
to consider - add projected ramp texture
- duplicate colour swatch / add colour to ramp
Animate camera to be looking where I am working… at least animate the aim constraint object…or create a new one… every thirty seconds is too often to animate the camera… so have a separate interval for the camera…

04_06_14

I am writing some separate blend shape tools…

Whippet in the sun ⇒
- try adding blend shapes in world space

QUESTIONS:
like Kentridges charcoal anims, could the animation here be derived from the production process (gina what does this mean?)…
Kentridges’s anims are constituted by capturing the process of drawing.
- What is the potential of 3D animations that capture the process of 3D modelling?
- what are the similarities between 3D modelling and drawing (or how can 3D modelling be like drawing/sketching or “seeing/drawing”), and what are the differences?.

05_06_14

BLEND SHAPE Tools;
I am using my “whippet in the sun” still life models as the basis to experiment with how to blend (or interpolate) between one model and the next…. and setting vis keyframes on each model…

SCRIPT thoughts / suggestions
- set the num of frames between vis keys (ie numFrames that each model is visible)
- set the num of frames between blend states (target state achieved on frame before vis0 )

TO DO
- create demo movie with cubes - modelling dupe script - key vis - create blends

THOUGHTS / DISCOVERIES / SURPRISES
- the blend script seems to create a complex blend node.. playing through the timeline with my simple cube example the movement is surprising but convincing, or evocative… I kind of like it… I now need to “reverse engineer” to see what I’ve actually done…

06_06_14
I thought it was a mistake that each cube deforms “all the way” but actually that’s because each is referencing the previous. I assume that this is happening on my shape blend script too -- ? will check it out

NOW putting world space blends on whippetInTheSun…

as I do so noting Post Production scripts / helpers as follows;
- select all transform nodes (gina this could be done with a selection set)
- a way to vary visibility keys in the timeline
  - this could be done during production (eg use a hotkey to execute the duplicate model function and the vis key duration reflects the amount of time elapsed between button push)
  - or post production (eg using music to distribute the keys)
- a way to procedurally animate the camera location and aim
  - this could be done during production
  - or post production
- a way to key transparency on models to use a ghosting of previous iterations
  - how to key transparency without a shader per object?
  - with the singleShadingSwitch node
    mc.createNode('singleShadingSwitch')
    mc.createNode('singleShadingSwitch')
    - where to start and finish fade keys?

07_06_14
I just had a thought that these blend shape tests (and auto duplicate tests) also come under the heading of “modelling as animation”

… MORE MODELLING AS ANIMATION
TRANSPARENCY TEST RESULTS
I tried my new transparency script on whippet renders…
Initially I thought it would be good to have each model always partially transparent. but because the models ‘pop’ on (ie there visibility is animated) this produces a flickering effect when animated

So I found the best results achieved by fading the transparency from 1 to zero…

THOUGHTS
- currently each model is deformed according to the subsequent models (ie the ones that come after)… What if
they were deformed also by the previous models?  
- turn off double sided

---

theres a problem with the models casting and recieving shadows... *edit... just need to up the sample rate*

---

09_06_14

**TO DO**

some more modelling as animation > still Life or Plein air… this time focus on details…  
eg with a plant…

-- try modelling one area and then moving the verts to another area?  
or have several cubes (remember that the one that I have selected will be the one to be duped).. the blending of one object across to another is a mapping of the movement of my attention…

[at a granular level would this be like a vector/line visible from one vert to another as I move one vert then another?]
One limitation of utilising the transparency script is that many of the models largely overlap [I just had a thought that this could be the basis for positioning the camera... ie for finding out which aspect of the model has been worked on... where one models verts differs from the previous is presumable where it has been worked on...]. To get around this I tried using “transform component” with some randomness to the transformation. This could be scripted to operate across all the models and to be animated from 0 to X to coincide with the appearance of the next model and then the disapearance of the current (component transformed) model.

11_06_14

“interior_004.ma”
Intrigued by the blend shapes making the form (ie the verts) move from the lying dog to the standing dog... I tried this rendition of a room in my house. The room is big enough that I dont really take it in in one go... my eye moves around it... I found that I was grouping (visually) different aspects of the room at different times...
I imagined that each mesh (or each shape node) could include what constitutes a particular GESTALT...
I was listening to Dreyfuss on MP’s Phenomenology of Perception as I worked and liked the comment that to say that the message of the gestalt theorists was “The whole is greater than the sum of the parts”is misleading.
Dreyfuss says that actually “the whole determines what counts as a part”...
I was also interested to hear that Dreyfuss thinks that the project of varela et all is misguided regarding MP’s work (or the significance thereof).

Ways I am guessing that I might visually group the objects in the room are;
by colour or tone
by proximity (in 3D or across the 2D “picture plane”)

The practical questions that arose from the above example where;
- what is one object and what is the same object (ie mesh or shape node)?
- is it better to dupe mesh or shape? is it good to have the option of each? would it be useful to do both in the same scene?...an “unraveling” at different levels...
- should two nodes be onscreen at one time? this could happen by changing their name...might look interesting when one blends to another with yet another stable / stagnant. ---- for now I might answer no to this
- I have tried blending “forward” so that one mesh leads to teh next.... but what if they “crossover” so that one mesh comes from the previous? or just the option to do either?
- working with shapes is annoying... not sure if I can create switch node for the transparency for example
- also not sure if transparency (and switch node) works with surface shaders --- edit...yes, I got it working... see below.

SO My SCRIPT needs
- check keys spacing
- separate production tab and post-production tab
- duplicate shape node or mesh
- blend on shape node or mesh
....
- add ramp as colour swatch and keyframe its transparency in a LT when colour of swatch is tweaked (production tab)
- add camera animation based on vertexes worked on or based on the position of the camera for the current viewport (production tab).... the camera position could be noted at the same time as the model is duped...
  options include;
  animated cam or still
  .... seems the camera would have to average between the different views and make a cut when the view changed significantly and stayed changed....
  .... can I assume that I will always work in the perspective view?
  ....could the script job actually run when the perspective camera moves?

above is two consecutive frames... they dont match up at all because the colours are so different...
I cant even make the colours fade because I cant add transparency to surface shader...
I need to link (or at least also animate) the colour toward black as well as the transparency to get the desired (and intuitively expected) result.

```
“interior_006pp.ma”
```

I got the transparency working as I expected with the dupes shape nodes by using a single switch animated from black to white for the transparency (this can be used for all the shaders on the shape node)... and a triple switch for the colour (NB. you must set the default colour before connecting it with the shader). This triple can only be used once per colour and is animated from specific colour to black.
above is “interior003_withTransformMistake” f111 and f112…
This scene file gave me a shock when I played back through the timeline because the transform node translations
and scales hadnt been picked up when teh shape node was duped… so the models I was working in where in
different places; the were different sizes and locations than the duplicated shape nodes…
Nevertheless this flawed anim (which is far from what I intended) has qualities that I like...

CAMERA ANIM SCRIPT

taking the decisions away or at least having a starting point to work with
I now want to return to the camera animation script...
working on “autoCameraAnim_001.py”

QUESTIONS
what will it be like working with two script jobs going?
to streamline should dupe model check execution time with change in persp tz?
   if nothing is selected and it skips the job could it try again sooner?
   if not, camera change might work better anyway...
or should I do camera thing with selection change (as per current dupe model)

14_06_14
continuing with…”autCameraAnim_004.py”
some thoughts;

KEYFRAMING CAMERA MAIN
need to associate each camera with a meshName
   - might be easiest to combine with the dupe mesh script ---- ???
then as a post process the constraining of camera MAIN is based on the
vis keyframes on each mesh

ANIMATING CAMS
- would be good to have some movement on at least some of the cameras
what would this movement be based on?
   - the position of the next camera? These keys could be laid in post but their values could be stored during
production… so I keep a list of created cameras, a list of positions and rotations… then I can access the position
and rotation before the current camera index and after the current camera index…
- OR I could just animate the constraint of camMain from one camera to the next over 10 frames for example
the DuplicateMesh function returns mesh number... how to add the camera creation and have it acquire mesh number???

AUTO CAMERA ANIM continued
How to associate a mesh number with the newly created camera?
How to deal with meshNumber = None?
    How to make meshNumber = the last mesh number?

→ “autoProcedure_003.py”
seems to be successful in creating cameras during production based on distance threshold.
Each camera has a “correspondingMesh” attribute which is the cameras “most recently created mesh”; so supposedly the one that was being worked on at the time of camera creation (although I can think of times that this wasn't the case.

Next is to work on a post production camera animation script that does the following;
creates camMAIN
parent constrains camMAIN to each of the working cams
checks the “correspondingMesh” attr of each working cam
animates the parent constraints of camMAIN according to the “correspondingMesh” attr’s

---seems there might be pros and cons of storing mesh number or whole mesh name with the corresponding camera... mesh name would allow me to have multiple names of meshes... mesh numbers will make it easier to change the corresponding mesh manually
----------- maybe store the mesh name separately as a string variable ----------

AUTO CAMERA ANIM continued............
currently I have split the camera anim into two parts;
Production;
    multiple camera creation while working
PostProduction;
    create camMAIN
    constrain it to workingCams
    key camMAIN constraints so that it coresponds to vis of meshes
    ---- I need to get the visOnKey of the next mesh also as the constraint0 keyTime
    + I would like to have other anim options such as;
        add “drift” to the camMAIN (so that it slowly moves to the next position
        add pan or track motion to working cams (this would entail getting the constraint1 and constraint0
        keys associated with the working cam...
        + remove cameras that are in exactly the same place (this could be done manually)
        + animate cameras that only move a small amount (check a cam with the next in the list - first position; -
        then rotation...if it falls within a certain range then make it one camera, animated).
        + ---- animateCurrentCam(towardNext, fromPrevious, ease=1,2, or 3) ⇒ drifts the cam from current
        position or into current position
        ---- subsumeNextCam(ease=1,2, or 3) ⇒ deletes the next working cam and animates from current position
        and rotation to nextCam position and rotation
        ---- insertKeysCurrentCam(insertStartKeys, insertEndKeys
        ++ animate camMAIN constraints... so that camMAIN drifts slowly toward the next cam then quickly snaps
into place

+++ ability to delete a particular working cam manually and have the parent constraints update...

dealing with ERRORS
I’m at teh point now where I see the value of dealing with errors in code... so I’m wondering;
should a function always have a return value?
    - I think not...
how do you best raise errors in Python?
    - “If you need to indicate an error always raise an exception. I usually tend to avoid raising errors
      unless it is necessary”

IXIES
- mesh_0 must be renamed
- set vis0 keys to frame 0

- make threshold (for making cams) pertain to rotation as well
- maybe create new camera with same pos and rots rather than dupe persp… cause even changing the vis on
  cams they still seem to be hidden.
- hide working cam group
done in ; autoProcedure_005.py

ADDITIONS
- maybe need a locator parented under each camera… with each camera constrained to the locator… then I can
  shift the locator for a particular camera and/or I can offset the parent constraint on camMAIN. This is in order to
  “take a step back” or forward.

--------
---- subsumeNextCam(ease=1,2, or 3) ⇒ deletes the next working cam and animates from current position and
rotation to nextCam position and rotation
--------
	got this function working. NB. the rotation doesnt work too well when auto animated. I might need a
“camera rig” or maybe just reduce the rotation threshold and change threshold test to AND not OR…
--------
	but I’m guessing the same problem will come up with the “addDriftOnMaincam”

--------
SHOULD I USE A RECURSIVE FUNCTION ??? YEP it worked...see below -----
btw; 1:] is equivalent to "1 to end"

SORTING A DICTIONARY
constrainTimeDict = {"cam5": [10, 20], "cam4": [5,10], "cam3":[3,5], "cam2":[0,3], "cam6":[20,40]}
sortedTimeLists = sorted(constrainTimeDict.iteritems(), key=operator.itemgetter(1))

--------
trying to work on a list and change it at the same time…. got it working;

import math
import operator
def returnNewList(myList):
    threshold = 1
    if not myList:
        newList = []
    else:
        print myList
        listLen = len(myList)
        currentItem = myList[0]
        print 'currentItem ', currentItem
        print 'len ()', len()
if listLen > 1:
    nextItem = myList[1]
else:
    nextItem = myList[0]
print 'nextItem', nextItem
value = abs(currentItem - nextItem)
if value > threshold:
    myList.remove(nextItem)
    print 'removed', nextItem
newList = returnNewList(myList[1:])
return newList

myList = [1, 2, 3, 4, 5]
print returnNewList(myList)

----------
“cameraAnimTools_009.py”

….. now for drift on main cam

----- NB. better results are achieved by breaking connection on rotation and adding locator and aim constraint...

21_06_14

Getting sick of camera anim tools now… feel I need to use them before spending much more time on designing and programming tools…

Last additions I plan to do today;

- Add offset Z Attrs to camMAIN, one attr for each camera
  what if these attrs want to be updated? check the new workingCamList; if camAttr is not in there delete the Attr; if camera doesn't have an attr, add the attr
- add drift on camMAIN

CHAIR still life----------

this amount of cameras created after 2.5 hrs of modelling with settings;
distanceThreshold = 20
rotation Threshold = 40

renderRaw from test camera moving
delete some cameras
add offset Z attr

manipulate model visKeys
create blend shapes

QUESTIONS--
why are ramps not shifting properly?

Try projected ramp colours (could add projectors to baseRamp?)
change ramp interp to linear?

A night of insomnia… stuck in programming mental loop… do something> press go > see what happens > make changes (trying to solve a problem > press go again> see what happens > search for glitches etc etc

--- by day this process can be intriguing and enjoyable… by night its pointless, repetative, a never ending loop. Feels like my mind is gripped by this loop which is ultimately pointless…

OBSERVATIONS re process and results
It was engrossing as a process… felt like there was an intriguing, and always changing array of colours and shapes to explore, to capture. At times I was trying to capture the forms (relationship of forms), at other times I was trying to capture the colours (and teh relationship of colours to each other).

None of the objects where moving, but I shifted my attention across the scene… around the chair… and then around the surrounding room. The light shifted throughout the time I spent modelling… from sunlight to artificial light (it got dark outside).

There were areas of hard shapes/ edges and lots of soft areas as well… I wanted to add ramps to the textures (I remember this particularly on the floor)... but I was compelled to work fast and didnt want to deal with maya nodes; I wanted to focus as little as possible on teh program.

I feel that teh “raw scene file” rendered is uninteresting compared to teh intrigue of making it….How to make the intrigue of watching match the intrigue of making???
CAMERAS and VIEWPOINT
I ended up with hundreds of “working cameras”. each of these rendered from in turn presents an interesting viewpoint. But when I constrain cam main it jumps around way to much.

_**Maybe I dont need to think in terms of linear cam contraints?**_

Maybe I render teh same time period from different cameras?
would I then present them in a linear sequence? or splitscreen; transparency; linear overlay;

_**testCam**_ represents the the place I was standing when viewing teh chair… _**working cams**_ represent various virtual positions I took up around the model chair while making it… I guess these are like the fact that my ability to move around teh chair is part of my experience of viewing it from a fixed point…

**How to combine these viewpoints in one image?**
render teh same animation from diff cams and play one after another? like a loop
one big image with lots of smaller ones (eg along bottom of frame)
render from different cams and project back onto geometry? ---- maybe a wireframe?

CAMERA VIEWPOINT AS TRANSPARENCY
-try projecting transparency onto models based on camera position and direction….
- use final (original) models as teh starting point for this (I could find my favourite colours or combine colours).

![two comp tests with wireframe render comped over the top; could be projected from camera view](image)

Testing projection of transparency onto model from working cam.
Projection could be under camera or could be on object and constrained to cameras…

----------CameraMAIN constraint key is not working!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
MODEL VIS ANIMATION
- don't feel that I have to move forward with model vis...
- maybe use the last model as the starting point. others are secondary to this

COLOUR ANIMATION
- maybe this needs to be more deliberate (less random). eg I could use the colours I have to find certain combinations. These combos could be hooked up to a control (eg lighter, artificial light, chair focus, greater contrast etc). Then they could be animated from a central control.

-not sure why some cycling colour tweaks are not working

SCRIPT THOUGHTS and CHANGES
CAMERA SCRIPT -
successful as a way to get unusual camera angles that I would never think of.
Not successful as a way to automatically animate a render camera

PLANT continued… dupes SHAPE nodes
Creating a camera animation which simulates working cams.
Going for abstract shapes so that a picture builds up

modelling books…dupes SHAPE nodes
- why do I lose texture on faces when I add a new face (sometimes)?
- where is the face list kept when I delete history on the model and the groupParts nodes are deleted (as they seem to be.

- finding that I am sticking with a stable filofax… should I later insert some book and table shapes?
- COLOUR… should I be thinking less abut form and more about colour??

---POST PRODUCTION
--- consider deliberately combining and splitting models for a good blend effect
ie using teh materiality of the software (its intrinsic qualities / what it does well).

GROUP PARTS node

---Page 183---
two frames from PPchair_016…. this anim uses the final model and cycles thru the colour tweaks. Ramps are set to “circular”. Ramp uses objects UVs.

PPchair_015 also cycles through colours on the final models. This time ramps are projected and so the colours move consistently across the models. Ramps are set to “Vramp”.

→ I like this one and would like to try it on cycling models… and maybe a moving camera…
A thought on cycling models… maybe play with groupId nodes also … and maybe mix up the models so that they are not going from LR to HR. and also make keyframes faster …. some very fast and some slower...

FINDING A WAY THAT IS NOT JUST ADDITIVE →
I want a technique(s) that explores vision as dynamic….

THE CAMERA →
I still haven't found a way to resolve the arbitrary nature of the camera…
--- I am imaging a slow move from POV camera and then interspersed with quick shots of working cameras. when slow POV cam, models are frantic… when frantic working cams models are slow..

BOOKS
For some reason I really like this image (again, its an image that is born out of the process; not thought up in advance).........Should I find a way to start and end on this frame. shift from BG to FG… currently I have the figure… work out how to treat the BG

28_06_14
I like the way that this image contains “observed colours”. Its not an image I designed but its also not arbitrary; in two ways 1) I chose to keep it and I chose to render it/ it attracted/intrigued me 2) the colours are not “made up” imagined, they are attempts to capture what I perceived. So I want to be stretched/surprised (but not in an arbitrary way) called/pulled

I have just been writing about how we perceive getsalts (I think) according to MP. D’ says that teh whole determines what are the parts... so once we perceive this image as books we might see the pink and orange as Postit notes. According to MP the colour of these notes would be different if they were fruit or leaves for example. Maybe this is why the colours (and teh shapes) only need be approximate; because they work in conjunction with each other (ie together they form the gestalt and that gestalt? --which I think is itself indeterminate--- ?adds to the colours and forms (makes it a papery orange)....

interesting that my swatches are limited... the light pink is also the highlight of the orange... in that context the pink is orange.... GIna how does this compare to Dreyfuss on middle C being different depending on the tune it is in? D’ says (I think) that middle C “announces” the particular tune...?
THOUGHTS on Scotchmere;
Had a lot of fun. was sorry when my battery ran out. was very absorbed. It was hard to “get anything down”. At first I thought it was a pointless, impossible task. first attempts were conventional lego men; I liked it when the man sat down and I moved the verts into position without trying to make sense …or at least while suspending teh sense making… or was it that I was “starting afresh” in a way…. ie move a vert to the position of a foot and let the edges be dragged where they will… try to capture an aspect of the form while letting other aspects “go to the dogs”.
I’m looking forward to turning it into a moving sequence (above shows the final models and colours I am left with after that 2 hour session. I’d like to carry on the practice each day and see what changes in my approach…. will a style emerge?
should/could I carry on a painting practice in conjunction?
How much should/can I work on teh anim “post-production” (ie back in the “studio; away from the objects of enquiry)

- I like/enjoy/am intrigued by the colours and shapes in these images even though they are not what I would “choose”.

I have spent a lot of time making tools (ie scripting) but I’m starting to think that the most remarkable thing (or the thing that impacts most on teh final anims) is the way that I use the tools… And I have noticed that as I use these plein air production tools I can begin to see in a different way; to approack teh subject matter differently/ to see/attend to/ notice different aspects.

- The weather has an impact. eg the figures above were created on a very grey day. you can probably tell by the shapes that most people were wearing coats also I have not broken the figures up in terms of different
shades/colors. they are mostly silhouettes.

THE FORMS I DIDNT BUILD ARE THE MOST INTERESTING
- im thinking of keeping teh anim moving so that teh forms are never still… perhaps they could slow (or even stop) in parts where theres an interesting form that is not of my design…

----- perhaps likewise; I make a lot of having a WYSIWYG workflow (eg no lights), however this scene as a sun and sky render is quite interesting;
Above is 4 frames from test comp combining moBlur and colour render

COMBINING TRANSFORMS (sets of meshes) FROM TWO MODELLING SESSIONS
"scotchmereStCombined_001" ⇒ two transform nodes and shadows with toon shader on teh ground plane

I guess this is like painting at the same locations on different days.. ie painting over an already commenced work…

- the two objects on screen all the time seem to increase the visual complexity more than twice as much I guess because there are a myriad of relationships that emerge)

USING THE UVS FOR FACETED COLOUR
- flat colours dont make the most of (dont utilise) the fact that the forms are 3D. adding colour tweaks to each ramp shows up the facets; it does so depending on the UVs (which are not of my design)

THOUGHTS
- it would be interesting to put some of the shapes under new transform nodes and rig (and animate) them…
  its a red man so maybe they each get skinned and then animated via my autoKey tools...

..........  

SELF PORTRAIT
Three frames from "PPselfPortrait003_06" rendered with SR 3D motion blur

I prefer these stills to the animation for some reason…

I do like the way that these images above are not deliberate… Would it therefore be good to take these “inbetween” model states and work with them? eg I could pick 10 of them, combine the geo and work with those models

MR, vector fill, and vector line renders…would be good to do a shaded trail render (like the whippet)

BINDING MULTIPLE SHAPE NODES

:) I just found out that I can bind the one transform node to joints (and it effects all shape nodes) as long as they are all visible.

This need to make them visible is the same as when I apply Blend shapes; I’m guessing it will be the case for all deformers.
transparency on SS shaders and other shaders; for shape nodes and transform nodes.

from above;

“interior_006pp.ma”

I got the transparency working as I expected with the dupes shape nodes by using a single switch animated from black to white for the transparency (this can be used for all the shaders on the shape node)... and a triple switch for the colour (NB. you must set the default colour before connecting it with the shader).

This triple can only be used once per colour and is animated from specific colour to black.

One triple switch per Material/SS
One blend node per shape entry into the triple switch
One single switch for all shape nodes

check if a triple switch is connected to outColor
if not:
    colorRamp = get the ramp attached to outColor
    tripleSwitch = make the switch
    connect tripleSwitch to outColor
    connect colorRamp.outColor to the triple switch default colour
if yes:
    tripleSwitch = the connected switch

check if a blend node is connected to triple switch at the current index
if not:
    blendNode = make blend node
    set blendNode  "blendColors1.color1"  to 0
    connect
    if colorRamp:
        connect colorRamp to "blendColors1.color2"
    connect the singleSwitch mesh[1]_trans attr to the “.blender”
connect the current shape node to the tripleSwitch shape index

NB. the alpha works very strangely on the surface shader. Seems it would work to connect a black and white ramp to the SS matte opacity (with a single switch in between) so that it goes from white to black as the mesh_trans goes from 0 to 1 (ie its the inverse of mesh_trans).

NNB. at the moment the trans keys are not looking at the mesh vis keys… this would be a good feature for when the mesh vis keys are not uniform.

05_07_14 and 06_07_14

“PPshaderCntrls_006.py” doesn't seem to be coping with face shader assignment :( 
------------------

It seems to be related to the triple switch node

AHA -------- it is not connecting the ramp to the blend.color2

ie. it only connects the ramp to one blendNode

I think this is cause it disconnects teh ramp and then there is no ramp connected…

--- how to store the ramp for all ? maybe check if a ramp by that name exists……

→ maybe make a dictionary with material as key and ramp as value

A DIFICULTY i just realised is that I need to get the correct entry into the switch node…

it might not be the index

→ however the empty slots dont seem to matter

NEED a separate button for keying transparency

and a separate button for network setup

PPshaderCntrls_008.py works apart from this

07_07_14

Make trans attrs go from 0 - 1

Cubetest_001.ma

A mistake - I want keys to be set on all the switch nodes (ie to control all the colours on the geo). However the script currently only does some… but that could be interesting as a way of breaking the models up and combining different versions of the one model. 
WHIPPET IN THE SUN now to test shader trans on lambert shader and also working on transform nodes

All the models in “whippetInTheSun” ie. a render with vis keys deleted
- would it be interesting to just have them all visible and cycle trans between them?
- or do something within the mesh? ie its interesting how they overlap

working with TRANSFORM
- mesh vis ⇒ works well
- add blend ⇒ blend duration does seem to work ...not sure about blend align?
- key shader trans ⇒ doesnt get the node list. try making node list always return shape nodes
NB first save as “PPshaderCntrls_009.py” this works well for shapes

getNodeList() gets a list of shapes or transforms
use shapeNode = self.getShapeFromNode(node) if I need the shape node rather than the transform. which i think I always do.
--- made that change. Now its making a shading switch for everyshape node...so its not connecting and then returning the connected singleSwitch.

“PPshaderCntrls_010.py”
SURPRISE!!! adding the trans attrs to a single shading switch (which is what happens when all models share one material), the attrs line up in the channel box… this visual configuration reminds me of my driver-multiple target setup script. Could I modify that to work on the shading switch? so that I get waves/ripples of transparency. Presumably the driver could be animated so that its not always the same speed?

Seems to be working well ---

BEFTER WAY OF SETTING TRANS KEYS and blend keys
keyspacing

--- I’m think it would be great to be able to vary the trail size
--- also be good to set initial trans off keys and trans on keys (so that model fades on ) …. in addition transparency could be also be projected…
---- what would it look like to just wash over trans over UVs???? not interesting cause they all have separate texture space. maybe if meshes are combined and use Auto UV projection…….that might not matter- might just make it interesting.

08_07_14
just got the maya command line and maya.bat render happening. Not sure why it wasnt before…
- maybe environment variables (I put the maya\bin location first on the list)
- maybe the writing in the .bat file

**template below**

render -s 1 e- 10 -cam camera1 -rd D:\GINA\RESEARCH\PROJECT\00_stillLife\images\plant D:\GINA\RESEARCH\PROJECT\00_stillLife\scenes\plant
Now I will set up renders and render straight onto my red external hard-drive
- should I do LR test renders first?

RENDER WISHLIST

→ whippetInTheSun_018.ma - coloured transparency

→ whippetInTheSun - black and white trail

========================== starting a finRen ==================

THOUGHTS
- I just realised that part of deciding on a render look/style is rendering and assessing a still frame (or two or three). Once the whole sequence is rendered and experience as moving/ as a seamless whole there are surprises. The same goes for Kentridge watching the fruits of his labour. Both his process and the 3D animation process are “one removed” in the sense that the artists is not always experiencing the final product (as it could be argued that the painting is). I guess I am thinking though that (perhaps even more than Kentridge - although of course he does edit his film) the 3D animation process involves making a lot of decisions about what's in and what's out (“between accident and critique”, Bacon)

Options for shader vis keys
- I want to be able to change the mesh vis and set the shader vis accordingly. So ability to set shader vis based on mesh vis or stand alone
- I want to try all the models on and vis keys from off to on, on to off. So 4 keys laid in total

I also want to try using expression setup to drive vis keys-------------
“transparencyWave_001.ma”
currently “createExpressions” has a tab for multiple targets;
NOW I want to add one for one target (eg switch node), multiple channels;
createExpressions.py seems to be working :)

--------------------------
14_07_14
→ whippetInTheSun
I don't like the renders so far
try changing vis key spacing especially on that last bit that I really like
“whippetInTheSun_020.ma”→
frame rate = 25 fps
deleted some models that I didn't like (eg too boxy)

--------------------------
18_07_14
Bought an 8 inch digital photo frame and took it apart. Feeling much more excited about the idea of showing work or making work to be shown. Seems that dealing with the specific characteristics of the display platform (ie the physical screen; its surface properties, its wiring, its scale) gets me interested and thinking about the possibilities that emerge... eg the dialogue between one screen and another, or its location (gallery/hallway/lounge room)
SCRIPT changes;
- *making blend shapes be done according to vis keys*
  - make it work for shape

![Image of a 3D model with blue and orange colors.]

this is a test of transform component (local with random set to 10)
I like the way it could be a way of combining colours (like mixing on the pallette/canvas/or in the viewers eye)

- *do the same for shader vis keys*
- do the same for transform component

MESH SHUFFLE CTRLS
-- work out a way to easily add/duplicate/delete transforms and shapes so that I can change the order and its not always going from LR to HR
  - make vis keys be applied according to name (ie mesh number is last number of mesh)
  - blend doesn't apply properly …. maybe I need to order the list …
*BETTER TO ORDER THE LIST so that it all works on meshes not necessarily of consecutive numbers……

-- try booleans? daisy chain them so that I end up with one model …. DOESNT WORK

THOUGHTS
looking at the beautiful visual feedback in tgraph editor makes me think;
ITS **SELECTIVE MAPPING** “ALL THE WAY DOWN” … from the interface feedback to the final output…

………………

19_07_14
Added a couple of shape tools to Plein air UI
  - *shuffle names*
  - *duplicate shape*

NOW thinking of a tool to (create and) animate curves according to the morphing polys
“PolyToCurves()”
something like the following;
create curves, snapping points to poly verts
“bind” curve CVs to closest vert
? when mesh disappears find next closest vert
how would I get it to compute getClosestPoint when the mesh disappears?

21_07_14
Working on Coloured Whippet…

Discoveries;
- dont work on heaps of shape nodes with the graph editor open
- now that I have setup the shaders selecting the transform node (which has thousands of shapes attached)
stalls the computer…. :/ I’m not sure why…………
  _____ went back to a version before shader setup and its still the same….slow!!!!!!! ⇒ 7.04pm

Questions;
- should I just use a selection of the models I have created, not all?
- should I add blend, vis keys, and shader trans before rigging or after??
- how do I bind different poses to the same skeleton? or do I need to split it up further?
---------- or do I bind the ones that are visible?------YEP  ----------

To Do;
- fix shader setup so that “Orig” etc are not in node list
- try binding whippet to skeleton

----- maybe start from all shapes together --- cull them first ---------- then bind in visible chunks------- maybe a
  script to make selected visible and others not visible (dont need to be keyed)...........
- HOW to extend visibility on some models without selecting in the graph editor
GINA at least the blend etc is according to vis keys…. and presumably i can put Blend etc on selected only…
?and vis keys.

TRY:
(I might be able to parent all these shapes under the same node then reapply vis keys)
cull shapes
add vis keys
extend some vis keys (might need a script for this)
test some blends (make sure I can apply them to selected only)
Basic animation on skeleton…
bind bunches of the shape nodes to skeleton …. (need small script)
----- would this be the point that I apply blends?
----- would a selective vis trail be interesting?
I like the vector render with line but it seems that Surface Shader is not supported by vector render

...btw … this is (part of) what the 3D artist sees and responds to….

NB. I have started to save helper scripts in the scripts directory of the Maya project
eg:

```python
→ selection = ['GINGER:head', 'GINGER:FK_ankle_FR', 'GINGER:FK_ankle_FR'
'GINGER:foot_BL_CTRL', 'GINGER:COG', 'GINGER:FK_neck1']
mc.select(selection)
```

They can be dragged to the shelf for quick access….

----------

24_07_14

RENDERS FROM COLOURED WHIPPET_09
same frame from different camera angles… first is deliberately placed camera, second is a camera created during production, third and 4th start with cameras made during production and adjust them to compose what looks right/interesting/dynamic/compelling. They forms and shapes that result (particularly in the bottom 2) are not forms that I could make up myself (ie I wouldnt concieve of them in advance)..

THOUGHTS for various renders
- Binding as Animation  ????? bind shapes in vaious ways so that the anim is different…. maybe work with each anim and then combine in the one file?

ANIMATE JUST TO BIND
have decided to animate the rig just enough to bind it…

ANIMATE FOR CONVINCING MOVEMENT
working with what I've got I then i change the animation on the skeleton…

→ get a good animation just once and then try with binding different shape nodes…. eg I could delete all except 10 and leave them all on the skeleton.…

Try;
blend shapes
vis keys
transform component

............... 27_07_14

Arranging the skeleton inside the changing dog models…
I notice that the forms (many of the interstitial forms at least) are ambiguous… I position the skeleton somehow that "makes
I have roughly aligned the skeleton with the mesh at each frame…
I now want to write a script that gets the currently visible mesh and binds that mesh to the joints…

→ bindVisibleShapeNode_01.py

Interestingly Maya creates a new skin cluster for each shape node

I have a script to duplicate a shape node but actually I think I want a script to shuffle all the vis keys along
so that the currently visible shape stays visible longer

THOUGHTS
- I have now bound all the shape nodes and am animating the skeleton… do I animate with mesh off; ie just using
the joints as the visual indicator?
- its about the smooth and consistent deformation of the skeleton in conjunction with the fast jittery deformation
that comes with meshes snapping on and off; ie. changing topology
In this sense its a lot like Modelling As Animation
- 

TO DO
- get animation looking believable (approximately 1 day) then try playing the animation with various combinations
of shape nodes… could maybe offset the anim and play with lots of whippets in the scene?

TO TRY / TEST
- transparency trails on shader (selective use of that)
- blend shapes
I found this old pic of spline whippet and am thinking that once I have animated the “colouredWhippet” skeleton I can bind this one perhaps?

Although GINA would that defeat the purpose because this one is more upright as perceived / observed ?? perhaps I could bind this one to “lying whippet” anim…

--------- I just had a thought about stringing all my anims together as a [narrative] film… the same way that The Lost Thing was made by Shaun Tan starting with sketches (with the story emerging). I start with 3D anim “sketches”….interesting how I get a little buzz of excitement and energy when concieving of an outcome/ a project…

above is a screenGrab from Default whippet…. I could tell the TR story by having her “behave well”…. I could even use my original model of her and have it blend into the photographic model ….

→ Could I tell a similar story with a COFFEE CUP? ie dragging the cup onto the “grid of knowledge” ??? or the “grid of presence” as Harman might say …
One style among many

Maybe my point is that the default workflow leads to one style among many → it doesn’t have “the last word”. Like MPs complaints about science, it’s not that I’m against images created with the default workflow but that they should be considered one style/one approach among many… that workflow shouldn’t be “transparent” taken for granted and assumed to present “the world as it is” or maybe not even an “accurate” representation.

31_07_14
SHOES
THOUGHTS
I still don’t really know the dif tween using transforms and shapes…

START == 2pm
FINISH == 8.30pm

02_08_14
Did greenJumper yesterday… about 3 or 4 hours worth...

Looking again at greenJumper model I can see that at some point the colours (as assigned to faces) have become jumbled up. I really don’t remember that being teh case when I closed the file yesterday….

I will now open my shoe file and see if the same has occurred………………………….

well it didn’t happen to shoes so I’m guessing its to do with not deleting history while modelling….

THOUGHTS
→ maybe the script should include deleting history on current mesh (or at least an option for this).

11_08_14
I realise that I haven’t used the transparency key options on surface shaders yet….

With whippetInTheSun the transparency did produce render glitches where the models overlap
I intend to remedy that using a script to add random “component transformation” to one object whilst it is temporally coinciding on screen.

WHAT WOULD IT LOOK LIKE TO set the vis keys to one with a trail of 4 or 5 and the blend to start at 1 and go to the end?

WHERE DID I PUT MY TRANSFORM COMPONENT HELPER SCRIPT???

maybe I haven’t written it yet….

I’m currently trying trans trail on green jumper and thinking that a SELECTIVE VIS TRAIL could be good…
15_08_14
just finished ~3 hrs of night modelling

26_08_14

whippetInTheSun2_002_transformComponent
renders still kind of pulsate Im thinking the next thing to try is to have the trails
selectively only...ie when the model actually trails
- could this work in with extending the blend shapes? or is the point to use just a select few models and rig them and animate
them (maybe repeat them) to create an animation?
- could I repurpose the animation I have already created?

28_07_15
working on Urban Animators submission…
Using it as an opportunity to amend my PaSI tools…
Blend and Vis attrs now are added to transform node and hooked up via nodes (condition, clamp and set range) this means that
I can MMB drag in the viewport to scroll through the shape nodes … scrolling slowly will enable the blend nodes, scrolling faster will make the models jump from one to the next...

Did this render of all models and thought it looks cool … Here all the models are completely opaque.

NOW →

page 205
hook shader transparency to model Vis;
… create new shader from same swatch (so other models with the same shader are not changed
… OR use switch node … which is simpler? probably switch …

make nodes so I can have trail

delete blend controls
amend Blend cntrls
… NB vis kicks over when over 0.5 > fix by changing setRange max value to 2
**Whippet**  
Production notes

**Default Whippet, Procedural Whippet, Auto Anim Whippet, Modelling as Anim Whippet**

*read from the bottom upward*

**Thoughts on Overall Project Intentions;**

Default Whippet

use “default” workflow to create a whippet animation.

I am still after images which represent a specific dog, observed under specific conditions, in a specific context.

What are the limits and benefits of this workflow with regard to that intention?

How does it feel to use this workflow (in terms of enjoyment, immersion, flow)?

Do I get anything “for free”? I.e. does anything special, unforeseen, emerge from the technique to add something to the character of the image; something that I like/ appreciate as “affective” but wouldn’t have thought to do?

Procedural Whippet

Auto Anim Whippet

Modelling as Anim

**TO DO IN GINGER FILE;**

Rig ear Dynamic

skin default whippet (tweak skin weights and save)

make hind legs upper adjustable

jaw bone

work on texture default whippet

   sss
   fur
   normal map

...  

13_11_13

**Questions and thoughts arising while making;**

I’m thinking that more shadow tests would be interesting. With multiple objects, some moving, some not…

ICONS for shelves

put them here;

C:\Users\e46908\Documents\maya\2013-x64\prefs\icons

...  

12_11_13

continue with MasA…. 2hrs max. starting from frame 984...

**Questions and thoughts arising while making;**

Q. is it more interesting to create a model at a particular pose and keep the history on it. Fade it off (transparency ) either side of that perfect pose.

---- to do this I would need to duplicate bound mesh (complete with bind history)...

--- OR shall I create each model afresh so as not to be thinking in terms of the previous form? A. It is definately a different thing to start a low res model than to alter a higher res one. I can now imagine that the animation, the process, could take a variety of different shapes…

-- The joints do to a certain degree dictate the forms I am making. I guess these could be equivalent to sketchy lines mapping the placement of a figure in a painting.

- Remember to animate history like “merge vertices” distance threshold. Hence using modelling tools (designed to achieve a finished product) as animation tools.

- I notice that when the model is smoothed changes in topology are much more noticeable.
-- I just realised that you can animate the polysplit edge location also….
- when you delete faces after binding, scrolling through the timeline produces very bizzare results.

Maybe start from frame 249 and create models to “fill in the gaps” (ie. body model)...
TRY zoning in on a leg for example, the rest is very low res (selective detail) -- what happens when you then scroll through the timeline?

Thoughts on tools to make;
- maybe a tool that links all the “divisions” of the extrude faces to an easy control to animate.

Thoughts on what I’ve done so far;

...

11_11_13
continue with MasA…. 2hrs max. starting from frame 984…

Questions and thoughts arising while making;
- is it better to work just on one area or a little bit over the whole model?
--- there is such a difference between deleting the history on the working model after tweaking (and then rebinding skin) and not deleting the history. If history isn't deleted the model jumps around and becomes something unpredictable.


07_11_13

Thoughts on my shadow script; I first had the idea of working on the shadow (instead of the mesh) using a technique similar to MasA. I did a quick test via a “brute force” approach. Then I decided to write some scripts to automate some of the processes involved. Everything that this script does can be achieved via tool or menu items available in the default Maya interface but using my script meant that it took only a couple of hours to do what could take a day or more. What's interesting though, is not that the process is faster but that the new tools (procedures which automate other procedures) allow you to do different things, the process feels different (it IS different); you end up doing things that you wouldn't without the tools (it's not that you do the same things faster).

In the case of the shadow script, the number of rays I could make means I saw a pattern emerge (where the inner rays get revealed when the dog turns). Playing around with the script on a cube, it looked interesting when I turn the cube. I can add lots of rays, delete ones I don't want, add more; and the same with the geometry. Laying out the UVs for the rays was a bit labourious; interesting to note that when a process (like that) is labourious I feel less likely to play around and test different alternatives, or just undo what I have done…

Also I didn't know that I would create rays and bits of shadow as separate bits of geometry. I also didn't predict (and didn't discover till very late) that the shadow would go through the floor when joints get close to the ground, I'm hoping that will look interesting when the dog sits down.

3D animators very commonly write scripts and plugins (3D software is very adaptable) but very often these scripts aim to expedite already decided upon processes and gain predetermined visual outcomes (animation styles/images). It's not so common for these scripts to deliberately stay open to revealing unforeseen aesthetic opportunities.

TODAY

Today I feel like trying my low res whippetShadow with a run cycle. I'm not sure if I will use the Auto keyframe script, I might just have the dog running through tunnels and maybe turning as well. Maybe the auto Keyframe/music could change the background (ie. the location of tunnels)?

To do,
get the cycle working with the default whippet
OR start new file with shadow whippet and reference in the shadowRun…?
OR start with default whippet just the anim… import shadow whippet geo… zero out transforms on the rig, adjust and bind shadowWhippet geo.
- -rfs = start frame number; overrides scene file frame numbers
- -precentRes = renders a percentage of the image of the set resolution

Ideas for new scripts;
- I’m think of a colour script (it could setup ramps and gamma correct controls, maybe these are attached to a set of sliders controlling deviation from a norm; set norm colour/ overall multiplication (deviation); direction of deviation

Questions and thoughts arising while making;
- TRY more than one shadow whippet interacting….

PROBLEMS;
it seems that a bunch of joints became unattached while I was fixing the LR geo….
----looking at shadowWhippet I dont mind most of the time that the geo goes through the floor but I dont like the way that some joints go through and then keep going… why is that? Is it because the model locaators go through the floor? Yep, I think so… Workaround is to animate the offsetUVW value on the riveted locator.

... 06_11_13
I like the way that as the dog turns the inner rays are revealed…

TODAY
planning to work on some renders for whippet shadow. Let the camera angles, texturing, and further modelling opportunities emerge…

3.35pm - have a render going for shadow whippet. Will now spend an hour or two on M as A.
at some point (when I get batch render working) I will set off a whole bunch of renders (maybe this is a good reason to have separate working files and import or reference cameras).

MasA
- so far a bit “bitsy”. I’m wondering how to get in the zone.
---- I’m working from a chaotic model; trying to “straighten it out and find the whippet form. From meesy/meaningless to meaningful/recognisable (reverse of usual from order/symmetry to disorder/asymmetry).
----------Got to frame 984
PROBLEMS
- I cant batch render from my computer because it logs me out after 10 minutes!!!!
- "shadowWhippet_025.ma" works as a batch render but not a command line render. This is quite annoying.
  it may be because it is trying to render with the wrong version of Maya and or Mental Ray
I learnt to use the -log flag to record the errors in a text file which is very useful.
  also read this;
  Probably the one doing the rendering is the one you installed last. I'm not exactly sure where that information is stored. If you want to explicitly set which one does it, you could have an absolute path to the render.exe instead of using the 'render' command. C:\Program Files\Autodesk\Maya2009\bin\Render.exe.
  C:\Program Files\Autodesk\Maya2010\bin\Render.exe
  my path is; C:\Program Files\Autodesk\Maya2013\bin\Render.exe
  ------I tried just putting this in place of the "render" cmd but that didnt work.
  ------ I now have edited the Path variable in the system environment variables (I basically added this path - or changed the 2012 one to 2013------- it seems to be working now.

Questions and thoughts arising while making;
- I am reminded of bats wings; inners revealed, unfurled, unfolded

Ideas for new scripts;
- a script to assign colours (maybe points on a ramp) according to the name of a mesh (ie, mesh1, mesh2, etc)

Ideas for amending existing scripts;
- delete history on polys (rays and shadows).

TO DO:
- group bottom shadow locators.

Tried to build the whippet shadow yesterday using my script but it is quite complicated to manually create the geo from the joints. I will now try to add the functionality to auto create ground geo and two side geo planes. This will involve selecting four verts in order…??
- it works easily for the bottom poly, but how to make it work for the sides? or is it a separate tool to select two joints and create a face between them?

04_11_13
Very happy with how my ShadowPlay script went yesterday.
There is obviously a lot of functionality that could be added …
  for now I will have a quick look at creating a polygon for the base shadow… A work around for this is to create locators etc for all the outside verts and then do this inside ones separately.
  -need to keep a list of the second joint positions----
  hmmm….lots of troubles trying to make the poly shadow. I will leave that for now.
  furthur Stuff --- I could assign a toon shader to the object?

So now I am trying it on my whippet mesh.
Why does the locator sometimes not work (it jumps to another spot)? Maybe to do with UVs? YES its the UVs :) interestingly the dog cant go back to bind pose. I should remember that.
Why does the orderedSelection on the list command work sometimes but not others? Is it to do with setting this command each time - `mc.selectPref(trackSelectionOrder=True)`

Starting to use two scripts on the dog file now;
- create poly between selected joints
- “shadowPlayMain” which creates joints, two rays and one ground shadow
- script that just creates joints with no geo
- script that creates ground shadow and binds it when 3 or 4 ground joints are selected
---I might need a script to create just one joint at the foot bottom ; but then again best if he can move feet upwards and retain shadow.
I worked around this by moving main control upward while making joints.

Interestingly the joints that are low sometimes go through the floor. Not sure how to fix this---
maybe make the original 3,4,5 triangle much smaller (eg .3,.4,.5)?
--- I could just break the scale constraint and set it to a small number ----

it would be good if UVs could be laid out when rays are made. I just finished laying them out on the dog rays and it took quite a long time….

Questions and thoughts arising while making;
- one simplistic justification for ShadowPlay experiments is that a 2D designer/artist can play around with shadows and negative space as visual/design elements. Its harder for a 3D artist to do this with the same degree of flexibility and maybe harder for them to work with shadows as elastic things (and things through which forms emerge). Shadows normally come after; as a result of modelling. BUT could model (forms/objects) emerge from shadows.
- I’m sure I read that you can constraint a locator to two points on a polygon. I now cant remember how… I can easily see how to offset the UV and W which could be useful.
--thinking now about texturing the shadow rays….can i use the UVs? Or I turn the light so that all shadow ray meshes point straight down, and then create texture reference objects and project texture.
CONSIDER making the ray planes NURBS objects -----too hard
INSTEAD just arrange the UVs on the polys I think
-- I could assign different shaders to different shadow bits (eg bottom, head) so that I can fade in rays for those sections as form or light direction changes.

...
03_11_13
starting work on shadow whippet… I have started by using the LR whippet model I used for the test and manipulating this model to fit the skeleton. I’m not sure if it will look any good. It makes me wonder if it would be better to reduce the original model to achieve the low res one? or maybe to achieve something similar then fade them in to each other…?

Questions and thoughts arising while making;
- what if I selected faces based on camera and detached surfaces (ie split model into two or more pieces) and then assigned different colours or different opacities to each model. If models where bound to the same skeleton would they look like shadings of the same model?

Useful Tools;
- tool to create locator and point on poly constrain it to a vertex

PROBLEMS;
-creating this tool brings up the problem of getting the number of verts in the selection… because maya lists verts
in the format “pCube.vtx[2:4]” it makes it hard to get the list of verts as strings to operate on. I think the simplest in this instance would be to use the API iterator, but I don’t know how…. A: apparently “flatten” will fix this!!!!

--trouble getting constraint to work. Finally got it working as follows;

```python
mc.select(['pCubeShape1.vtx[2]'])
mc.select(['pCube2', tgl=True])
mc.PointOnPolyConstraint()
```

----how to constrain the bottom joint to the ground??

I thount that I could just constrain to the Y axis but that doesn’t work (probably cause its a joint).

Can I alter the scale in a similar way that I have made stretchy IK in the past?

A: my script seems to be working quite well but some numbers don’t work when you enter them at the start for joint position values… I’m not sure why.

ALSO when I test it on a cube it almost works but doesn’t quite get the correct scale. I’m not sure why? Is it something to do with the cyclecheck error? ANSWER: well it seems to be fine when I start with the default joint positions (ie for a 3 4 5 triangle)

In the “divide’ multDiv I think I need to enter the length of Z rather than “5” for the second input.???

……the above is wrong but it seems that there’s a simpler way of changing the shaodw (lighting direction) which is to rotate the joint..

--------this makes me wonder about offsetting the rotation of the joints (ie. I could use my offset script to setup this relationship). Interesting how the scripts start to fit together and form an idiosyncratic visual language.

...

01_11_13

I have rendered two views of my default whippet which is not so default as it happens;

I have left the texture flat, like photographs (this makes me think that it might be interesting to incorporate the fingers that were in my original photomontage?

I hooked the nose up to noise and started to ripple this throughout the body. This makes me wonder about starting an animation from the nose (rather than the COG); could the larger movements eminate (or be inspired by) these micro-movements and somehow be convincing?

The animation main poses are derived from observational sketches; an hour in the park with a particular whippet. this makes me wonder about the value/difference between refering to/studying/observing an object first hand rather than using digital or other representational sources. I wonder if, because you can be aware of more ways of interpreting, I might be open to different ways of representing the whippet (via the way she interacts with the world through her nose for example).

I have 4 hrs of work time left today and will now work on a “modelling as animation” whippet. first a cup of tea…

Observations about what I am doing/ aiming for/ thinking:

- maybe I need to change the animation a bit for the MasA project… maybe keep the same in and out point.
- turning back to the sketches I’m wondering again how to leave areas “sketchy” in 3D. Also how to capture the beautiful areas / shapes of the sketch without having to carefully describe everything (selective detail).

I realise that I have assumed that sketchy areas would be translated as low resolution or maybe transparent; but now I wonder if the areas could be messy/complex? something easy and “automatic”; glitch or tool outcome.

…or moving a lot (Gina remember that this is temporal)?

- I would love to use the “divisions” as an emotive/expressive/ usefule animation element; something that conveys a similar revelatory sensation when you make the model (use the tool/ add the divisions and watch the form change and the character emerge).

- create wireframe as a way of drawing (perhaps the wireframe could be superimposed over the render?)
- or am I just going for shading? … using a specific light source and matching the shading to the sketch?
Questions and thoughts arising while making;
- dont forget that working model should be bound........
-- I am shifting the emphasis back onto process... the activity of making is important because it is a way of thinking, and a way of relating (?to the world).
-- I could use any one of my bound models and extend its visibility throughout the timeline.
- I have lots of models overlapping... I could rig it so that the more models, the more transparent each model. Would this give the equivalent to a "sketchy" look?
- should I be working to camera/viewpoint?
- will overlapping models be a good idea? I mean, using the same animation (i.e. the same animated rig) and working over that; will it be interesting to see different ways of "solving the problems"? I mean its all about process, I'm working with the same subject matter to be represented (the same starting point/inspiration) and I am also working over the exact same animation (at times. Could be interesting...

31_10_13

Animation expressions wishList
I would like to be able to dictate what axis the “Offset” attribute effects

Observations about what I am doing/ aiming for/ thinking;
- I am creating expressions on the rig using my “animation Tools” script. Is this any different to just creating these expressions “by hand”? The answer could be yes, in the sense that if the tool is there, I am more likely to create the connection that way. I am more likely to think in that way.
- can the dog character be “convincing”, i.e. be animated (gain the illusion of life) even if the imperfections of teh model and rig (the 3D representational process) are evident? i.e. can it look like a “convincing” 3D model/3D representation? i.e. can it be what it is and be engaging (not, can it be a convincing representation of something else/Ginger).
---maybe 3D is an interesting representational system, but I can imagine other representational systems... and I wonder when a representational system becomes performative.

Questions and thoughts arising while making;
- Gina try making different whippet characters with the same base animation. I can try using the “Neck Expr” for example.
- Try creating my own renderable Manipulators?

30_10_13
I feel like I’m getting into it now after several days of animating (hard slog), I can see a character of sorts emerging.

Observations about what I am doing/ aiming for/ thinking;
I created some noise animation on a group and hooked the nose up to that.
I wonder if I can "ripple" that noise movement throughout the body??
btw, I currently did that in the animation file (so its not part of the GINGER rig.

Questions and thoughts arising while making;
- should “breathe” be linked to nose anim? or just have a control called sniff? I think with anim layers we could
have breathe constrained to sniff just on one layer.

29_10_13 ⇒

Intention for today is to finish a rough pass of the animation and then move on to using animation layers. Or add breath animation on a layer early on…

I will refer to Ginger who is beside me today and also my movies of her that I took on my phone in the park…

Observations about what I am doing/ aiming for/ thinking;
I am making curves (drawings, interrelated blue green and red lines) in the graph editor.

Questions and thoughts arising while making;
With animation does it matter if you work quickly?
- is it useful to think of a point of emination such as the nose (eg the nose through which the fold occurs, rippling through the monads of the dog).
- IDEA for whippet that “sniffs a song”. ie music as data ripples through the whippet.
- is it the timing or the location of the keys that is most important? I mean what can I get away with in terms of location (eg randomly or musically generate keys) if the location of the keys is “correct”…
- bitzers VS purebred dogs as an analogy for performativity VS representation (and chaos VS cliche; algorithmic/procedural/random/glitch VS deliberate/keyframe/stylised)
- i want the work to be about taking up opportunities as they present themselves. Listening— What (if anything does this have to do with Heidegger’s concept of the struggle between earth and world?).
- Painting metaphor - keyframe in and out tangents are to an animator as“edge quality” is to a painter.

ideas for animation tools;
- generate random keys or move/nudge keys random amount…(eg. I could position the keys and then use the tool to create their value). Perhaps work with layered animation so I generate the general movement on main control for example then generate other moves in relation to this.
- create keys on a curve then move them along (this would make a fold-like visual impression). Would it make a good animation?

28_10_13
to transfer; scenes - GINGER + MasATEXTURES; UV snapshot; tga texture;
Still animating… have gone through teh timeline once but its still very mechanical movement… wondering about adding breathing and also blinking blend shape…

Animation Layers-
I want to try adding noise to the animation. Have thought about doing it with animation layers… A problem seems to be That you cant see the combination of the curves (keys) in the graph editor.
It looks like you can have constraints just applied to one animation layer… and on another the same node can be keyframed. SO I’m wondering if one anim layer with procedural animation could be the inspiration for a character animation to emerge????

You can set keys on the weight of an animation layer…so stupidly a layer could contain animation that counteranimates keys on another layer and the weight of the layers could be used to merge between them

Animation layers could be useful because they might make me feel less precious about working on/changing the animation. It might be like applying a wash over the top of an image and then

Some alternatives might be using a blend node and noise texture…Maybe be I should look again at my “Library Man” files...

Thoughts while making:
Still I find looking at her is so much more complex and interesting than I could ever capture…

Perhaps there is a point where you have to stop looking at the reference material (and even stop looking at the whippet?) and focus on the materials/character emerging… attention and care and intrigue about what can emerge… tune in to what is emerging moment by moment…

Maybe like Varela on mindfulness, its about not proceeding mechanistically but proceeding with an [intuitive] awareness of what is unfolding…


25_10_13

I am still animating the whippet in a relatively straight forward and “normal” procedure/workflow. All the while I am thinking about how to use the tools/ software/ code procedures [how to develop or manipulate a system] so that something convincing emerges.

I think I will time limit myself on this initial animation. I will work on the worst bits first and finish the animation to an acceptable standard by end of day today (or earlier). i will then be freed up to move onto modelling as anim and shadow etc when I get the urge.

Helmut asked today why I am working from sketches, and from a real, particular dog rather than from my imagination. Its a good question… the answer is that its not accident. I like to work from “life”, from personal first hand experience of objects/world, from my intrigue and wonder about access [ie. questions about the world I am born into, as an object amongst other objects, and yet a world which I interpret/perceive according to my biology, culture, needs and desires].

-- All objects only partially acces other objects; art is about translation [Halsall]. How can 3D [character] animation be convincing but not “stylized”/ reductive/ caricature.

Observations:
I’m finding it a bit boring today. Not “more than I’m putting in”.
Interesting how I look at the movement but also look at the shape of the curves in the graph editor.

24_10_13

Heidegger and TRUTH as UNCONCEALMENT vs TRUTH as CORRESPONDENCE
What does this difference mean for animation?
Is this what some painters explore implicitly?

How to develop an animation based on observation? - why not use motion capture to accurately represent how Ginger moves? because I want the movement (the exact character or style of the dog creature) to emerge via the process; in the end it is not Ginger (?chaliceness), it is not me (artists intention) and it is not the software (silver), we are co-responsible.

也许 I could observe the minor movements of the whippet and make attributes (or link attributes) that achieve these movements? so maybe again its about making connections/ defining relations rather than doing keyframe animation (does this make it more like procedural animation? what would “processual animation” be?).

23_10_13

I enjoyed making the whippet shadow animation and thought I would continue today… But I now feel like I might like to do some research about the best way to rig a shadow… For this I was thinking of using the QR tools to orient the joints but would this mean getting the vertex normal? Im not sure. But is might certainly mean getting the vertex position and then moving the position to there. I would like to test someways of getting some of this
automated using Python. I tried to find my Denso tree script but with no luck so far (it might be on my broken harddrive).

So today I might carry on with the animation and even do some “modelling as Anim”.

ANIMATION;
I find myself looking up whippet movement and forcing this movement upon the model… going through the usual animation method I know… its boring and always looks worse than I want. How can I surprise myself… and get the machine to do some of the work?
Perhaps its about experimentation, ie trying something and seeing how that looks. Allowing the character/movement to emerge rather than having predetermined idea of what the movement should be. 

Interestingly I’m thinking about all this in relation to affect now --- micromovements 
modular design of 3D can be interesting, ie encourage surprises (eg animate rig with one model then turn on another; set up controls such as footroll then add anim of these controls on top of other movement.

Thoughts while making: 
I am animating with the default whippet because I think that one might be less forgiving… 
- wondering what it would be like to do a rough pass then tweak animation later; maybe keep tweaking it and maybe even try different iterations keeping the start and end point to same. 
I am now wondering whether “final animation” should include different iterations to stay true to “show process in final images”??
- maybe making the rig (ie. setting up driven keys etc) is like mixing colours and finding tools etc… and so just like a painting where easy brush strokes bring a figure or object to life, in the case of 3D sliding teh control brings character to life.
--how to combine different whippets? Maybe pull camera back past whippets; they can be angled differently and maybe all slightly different
--- I can see a shadow anima… and maybe an anim with different coloured planes so no solid object… I think this one might need a tool for rigging shadow and also snapping verts to verts...

I have created key poses based on my observational sketches.
I now plan to space the poses to give the animation more variation in tempo.

IDEA FOR A FUTURE TOOL
I would like to be able to choose a music track and use that to space the poses, I imagine this would be part of the working process (getting feedback from the software before moving on to the next phase of production).
I guess this is like introducing a “random” element into the process - How does this relate to B and G’s idea of immediacy and hypermediacy? I don’t know, maybe it doesn’t. It feels more related to Bacons comments about “accident and criticism” - which in turn feels a bit like Varela on “natural drift”.

For the moment I will press on with this MasA piece. I can always come back to this file and take it in another direction with music driven spacing at this point.

-----Chris is beside me clapping and clicking… it makes me think that input via clapping (while music plays for example) might be a good way of moving from pose to pose … this seems like the type of thing that Brett Victor calls for… a type of performative element to animation production (also reminds me of J power’s camera demo and Digiglove).

THOUGHTS during production
would there be a way for the whippet to interact with her shadow (I imagine it as a simple piece of geometry).
??use the same method that Brian used with joints to get the location of the PV…

SHADOW TEST
How to make the shadow only on the Whippet Overlap?
Can I make the material only render when two surfaces overlap?
- would joints be the best way to alter shadow? ie. i could make another rig for the shadow?... Gina first try out just swapping models and then you can see how it looks...

---this is reminding me of my ideas for anims that wcould be grouped under the heading “stop making sense”.  
When using 2d software or paint or drawing the lighting obviously isn’t algorithmic.

----------
09_10_13
THinking, reading, and even writing have been more interesting to me lately than making.
I have 2 hours this afternoon and will now start making…
What has been stopping me? maybe the idea that I first need to animate the skeleton before I can enjoy the modelling… I have wondered where to start… how to approach it. I will now dive in and see what happens in two hours…

THOUGHTS arising while making:
If my project is about the activity of making then it makes sense to record the time spent doing/making something.
Maybe I think of it in terms of hours of interaction rather than finished products.

well, the time flew by once I’d started.
I ended up doing no modelling and only animation, using stepped keys and plotting in some positions as per my sketches and observations.
I’m now wondering whether, having got the key poses) should I use music to set the distance between these key poses? and maybe music to move some controls so that they quiver a bit or whatever?
It does look as though she’s listening to something...
Maybe its about the making process actually adding something so that its not a lesser representation of a pregiven idea.
I like this idea and its interesting how it only came to me after (or during) making...

07_10_13

Modelling as Anim Whippet
TO DO (intention today and this week):
refer to storyboard.
bring skeleton and rig in as a reference (turn off default whippet model OR leave it on as reference for 1st pass animation?).
animate skeleton a bit
create some key models…

TO DO (longer term):
make icon for modelling as anim UI

RIG FIXES;
neck connect skin joint scale to FK joint scale
…it seems that this is done by connecting “neck2” rather than the “bone_neck2” joint
delete sets
   head verts and calisthenic sets

THOUGHTS arising while making:
could the skeleton (curves and joints) be useful for making the models? as in it makes me think of the form holistically (i.e. turn the hips this way to go with the chest turned that way) or is it an example of “prescribing the outcome”?
   I am inclined to use colour or at least shading… but do I instead just model to the camera and lighting?

18 and 20_09_13
---------------------fixed read file function so that i can enter driver keys in format “1,2” etc.
---------- saving and loading of defaults;
------------ enter time offset as a fraction of the distance between one key and another
   ie. the use of “time Ofset” requires two driver keys.
---------- delete Tab; I worked out how to do this usinh the “mc.deleteUI()” command

rename Tab
“refresh all keys” button;
   how to return a list of the AutoKeyframeTab objects that I have made???
---------------------I am putting the folder text field on each tab so that it can be accessed when writing and reading the files.
---------------------It also needs to be on the main tab for the “create tabs from files”
XXX - create Tab button should load the target node - this might be too hard

17_09_13

Fix Auto Anim tools;
get the script to loop through all the frames on the driver channel in lots of “numberOfkeys”. use the remainder operand for this.

load folder of tab values [create one tab per file]
create a folder for the scene or name of the animation
save the presets to the folder

save all tabs ??not sure if I need this. might be best not to because I may not want all tabs...
saves a defaults txt file for each tab.

---------- Refresh all keys
delete Tab

Time Offset ---- I just realised that teh time offset should probably be a fractional (relative) number. because when I speed the driver up the key gets laid on the other side. …
Or maybe I try harder to save the indices that are written as a list (they could be enclosed in brackets or something when they are a list).

13_09_13

TO DO;
finish default anim run cycle
  2nd ear
  2nd toes
  skinning
    I am painting weights in the anim file . is that a bad idea (ie i will try exporting the weights).
  texturing (Zbrush)

Auto anim run cycle
  setup node relationships

Fix Auto Anim tools
load folder of tab values [create one tab per file]
save all tabs ??not sure if I need this
Refresh all keys

Rig wish list;
   Eye blend shapes
   add stretch and squash on legs so that IK handle doesn’t “pop”
   ?? dynamics on ears

Things to be Auto Animed;
   delay on second foot etc [gina hook this up to one master attribute - it can alter per node by + or -]

**Thoughts occuring in production;**
GRADIENT colours of PAINT SKIN WEIGHTS TOOL;
   would it look interesting to do a hardware render of this display feature?

SOUND could also generate objects. eg objects for the whippet to jump over.

----------
10_09_13
working on my default whippet… its so disappointing. The movements and look of the stationary model are always less than what I’m trying to represent.

----------
03_09_13
   doing some default animation; ie trying to do a basic run cycle
   ref imagery = slow mo film of dog running
   I will now try to animate using the layered approach

Thoughts arising during production;
The dog model/form looks so different once its moving. This is obviously something to do with the skinning but also other things I think…

… to get the animation looping through the timeline it seems that it has to be quite perfect in terms of exact numbers…
Maybe I need to start again with basic numbers…

----------
31_08_13
RIGGING AS ANIMATION
I’m still working on the Ginger rig and I was just thinking how so much of the quality (“convincingness”) of the animation is embedded in/dependent on, the rig. The rigger kind of pre-programmes the character’s movements and then the animator choreographs / keyframes those movements. The rigger decides / dictates what movements can be made, what are the controls, whats controlled by what, what are the relationships between the controls etc…
Maybe the character is largely dictated here/ at this stage/ ie in the rigging…?
As with the image below, the above is a movement (animation) of the rear leg testing the use of a blend node in the rig to turn on and off the IK and FK control curves. This is standard practice for rigging (making and utilising these connections). I’m wondering if its a useful or interesting or generative way of producing narratives/animations/bringing a character or “story” to life.

Working through the rigging tutorials...I am rewriting a script that makes 2 duplicates of a joint chain and orient constrains the original chain to both of the others. The pic below is an animation created by animating the influence of each of the duplicate joint chains...

Today I decided (until the next decision) that I would like my research to be about the importance/influence of process. How does each set of tools/processes/materials influence/alter/dictate my relationship with objects? And by “objects” I mean the things around me; mainly my subject matter, in this case Ginger. If vision/perception is subtractive and we only see what is important for our particular concerns, then what aspects of Ginger does each set of tools encourage me to engage with? Does this type of engagement extend to other objects in my environment? And by objects I also mean the tools, materials themselves [how do the tools encourage me to engage with them]?

today I installed Zbrush. Have just taken heaps of photos..
Today I feel like just creating a model (then maybe later an animation) derived from my observational sketches of Ginger…

Looking at these sketches makes me think about the difference between using polygons vs NURBS… the sketch I am working from now is curvy linework which I can imagine as curves in Maya easier than polys.

I have just noticed something about the sketch; the back right leg has lots of lines; you can see where I have grappled for the right line to describe the back leg…
This could be about the sketch translated via polys and also via NURBS?...

Perhaps for now I use the rig I have, and limit the movement to not having the foot off the ground…

Interesting how the geo always wants to align itself with the grid (world space) ….

-------

17_08_13
Doing my rigging course homework which entails building a whippet rig according to the workflow/techniques of Brian Evans; industry expert.
Its a lot about connects, constraints etc...
as I work I think about how these processes could be repurposed to act as metaphors.... do I mean the work process as metaphor for a way of being/interacting with the world? or the results as metaphors? I’m not sure but I think more the former....

-------

14_08_13
looking at my pics from yesterday.... reflecting on the fairly boring day or two that I have had...........

I tried to finalise the form of the model and also to lay out the UVs and do at least a stand in texture.

The moments of interest / excitement were one glimmer when painting onto the 3D model in Maya vieport and I noticed how the rotaion on the viewport in relation to the model impacted on how the “paint was applied”.... cant say if that was helpful or if I used this quality/alignment in the end, but it was a glimmer of interest (“what could this be a metaphor for?”)

The pictures below represent the few moments in teh process which interested me. The outcomes were unexpected (or not entirely predictable) and also the character (form) seemed to take on a life of its own. Even the abstract image below (the texture map painted with maya “ovelapping seams”) has a quality, I wouldnt say that a “form or character” emerged but it does “take on a life of its own”.

If I am now to continue with texturing this default whippet I could paint a bump map and a clour map, a spec map, and an incnadescence map for a Blinn shader. Or I could use an mia material X and use a similar collection of maps. Or I could create maps for use with fur (baldness, length, colour tip, colour base, specularity). and then other colour and incandescence maps for skin.

My intentions with this is to create a “default” whippet representation, but I’m not clear in my mind what the rules are for digressing from this default workflow. I suppose already I am on the lookout for glimmers of materiality imposing itself in a way which resonates.... maybe the process and outcomes is tree-like in the sense that I keep moving forward with default workflow and explore unexpected/ hunches/ interesting offshoots as they appear.

QUESTIONS:
from looking at the pics above;
how could I make an animation from the UV texture image above?
- bake projected texture as automated process
- move the model and project the UVs in a different way........

what would the empty furbag look like running?
How do I gain and lose hair? make it grow on/ grow off? would that look interesting?

I couldnt resist a “reduce” test
It makes me wonder what would happen to fur when the model is reduced?
It would probably behave like the texture map because I think fur also is dependant on UVs.

13_08_13
Aims:
finish whippet model

Thoughts:
I have taken some photos and videos to work from. But how do I ascertain the colour of something (ie Gingers fur) without lighting conditions? it doesnt really make sense as I think about it now.

To date the dog model has just looked like a very poor second compared to all my reference material, and it seems to hardly compare at all to my actual experience of my dog (who makes me smile almost every time I look at her).

I am finding this process (modelling, laying out UVs, and now texturing) quite boring on the whole. So far I havent really felt that I can “work with” the processes to create something unexpected...or I mean that I am looking for ways of working that are easy, are aligned with the tools, ways that the tools, and teh nature of the materials make easy. I am sort of looking for a “good fit” (and maybe a surprising fit) between my intentions (the outcome I envisage) and the cause and effect of my actions on / toward the tools. For example I am right now painting a rough colour texture and the only glimmer of interest/excitement I have had thus far today has been when I saw an alignment between the colour map I’m painting and the orientation of the Maya viewport....what I mean is that if I paint flat in screen space, I realise that its all about which way I “tumble”.
So I’m looking for an alignment between the physical (material) qualities of making and my intention for representation, or the ?illusion of a character, the quality of the representational image emerging. Gina does this relate to the guy in Camal Knowledge who talks about us being part of the physical world and therefore are way of understanding/ percieving is a metaphor/ or is aligned to material processes [something like
playing with fur for 15 mins at the end of the day is the most like fun, play, interest that I have had today and yesterday.

Questions:
How does working from photos differ to working from observational sketches?
Why is it that I am compelled to work from photos (from net and of Ginger) when modelling “default whippet” in Maya?

12_08_13
Aims:
revisit my whippet model
    shoulder
    feet
    pinched geo
    fur/hair
work through workshop which represents industry workflow.
    He uses anatomy pics. Hes doing a dragon and starts with human anatomy (comparative anatomy).
- go through rigging scripts that he has supplied

Thoughts:
as I will be always binding the mesh to a bind joint hierarchy, I can work with that and then drive it via different rigs, and even Mo Capture.
    Even though I have the whippet right beside me, I feel that creating an accurate model that will achieve all her amazing shapes is an impossible task. and I am becoming quite bored and frustrated today. Sick of modelling... It is especially hard knowing that the mesh should be quite uniform so that it deforms niceley (predictably).

09_08_13
Aims:
combine 3 animation tabs
test the usage of the tabs
    record my experience noting the following;
        is it fun, immersive?
        are the outcomes expected?
        are the outcomes interesting?
        functionality that should be changed / added / or deleted

functionality notes:
- make buttons the same colour
- make buttons in the same position
- add music input tab, text or image file input

Questions / Thoughts:
how would these tools be translated into equivalent modelling tools?
I aim to use my abstract whippet model/rig to create a test run cycle.
I will establish the expressions and the Auto key presets all in relation to the movement on the ROOT node
I will then test what happens when the movement on the root node changes........

08_08_13
Aims:
---------populate defaults after getting number of keyframes
---------read and write default values
         write a text file for the values in a tab, ie. the dictionary for each of the keys. Include Driver and target
         nodes and number of keys as well. Name the file with the target nodes and a prefix (such as “run”).
         read a text file (dictionary) for each tab
         ---save a text file (in the same format -?dictionary) for the values of the keys - gina that would be difficult
for the value... I suppose that the driver key index(es) would be as in the form. I would reverse engineer the
algorithm to find the value mult and the value offset(maybe value offset would represent the remainder.
---------incorporate three different tabs

Activities:
writing and reading files in Python
PROBLEM - I have been getting an IO error
Answer:  it seems that folder names cannot start with numbers.... I will continue to test this theory. I might change
them to letters or just let them alone.

I can save the file fine... now trying to load it...
I was trying to make a dictionary but this might not be neccessary
         maybe I just have a list for each key and populate the table with that
I'€™M STUCK on populating the key values.... need to find a way of iterating through the textfields
ANSWER --- I made them into a list :)
Now I just need to work out how to enter a name for the file
         why is it now not writing to a new line? A. it seems that I need to use “\n” on windows XP. But will that
work on windows & at uni?
here is a link to answer to the issue,
and populate the defaults maybe after number of keys is decided

Insights:
It seems that the file has to be closed in order to write to it but it can be open in order to read it....

Thoughts:
Reading in a text file;
could I read the python text file that I am working on (or maybe save it in txt format) and feed it into maya in a
different way?
 ie. as a texture map, or as a shape/form/ model; or maybe as a type of movement, or as colours....
what parameters of the file would I use? the length of each line? the number and placement of non-asci
characters?
Reading in an image file;
how could I interpret a sketch? tone is the obvious thing
would I make particles or a point cloud? but how would they get depth data? would I do crisp dark lines for
foreground and fuzzier for BG?
Or more divisions on a mesh according to tone, then a random function alters the mesh; it deforms differently
depending on the mesh resolution (density). 
would it be interesting to interpelate between one sketch and another?

............................
from last night:
-- add default values
-- try saving presets files
-- Refresh all keys button (maybe make this same as new tab and pass the button into the class?)
-- test using the tools on low res whippet
---- error checking;
    a message when list index exceeds the number of keys on target curve. Our maybe I wouldn't need this if I populated the default text after entering the number of keys.

-- driver from tab 1 is deafult on next tab (I dont know how I would do this.... maybe it would set as default?)
-- bundle the tools together (maybe in a class so that I can have separate windows??)
-- work out how to call the tool
and/or make a shelf icon for this and Modelling as Anim
---- make a music tab--------
----------maybe adjust the script so that it includes Offset, Multiplier, and maybe Delay attributes. Or would this be better done with a node?
or maybe its not needed because you can always adjust the keys?
--- would it be useful to be able to populate the key table with the value and placement of the keys (so reverse engineer if you like)????
......

07_08_13
Summary:
frustrating yesterday, but I'm happy with the functionality today. good enough to test.
latest file is “AutoAnimTab_48.py”

Activities:
I am partly redoing my anim tools UI. This time making sure that the key value text fields have unique names (unique variables). This so far means not populating the key value text fields dynamically. I have it currently set up for 5 keys. The user enters how many keys there are and that number determines how many dictionaries are made.

as before i have made a dict...DICTIONARY: I have created a dictionary for key values. I think that might also be a good way of storing default values????
driverKey(s) = 0
time offset = 1
value mult = 2
value offset = 3

I just made a realisation:
I was getting the number of keys from the text field in one function (the populate tab function) and passing that number to the make dictionary function. This meant that it didn't reread the number of keys after the tab had been populated. So the lesson is; heed when the function will be called, this will (help) determine what args are passed
from one function to another.

Wish list:

- save presets
  - delete tab (tabs must have unique names. so user must select a channel and node before creating a new tab. If they mess this up the tab becomes redundant. It would be nice to be able to delete this tab.
  - Refresh all keys -- I cant imagine how to do this one....maybe a list of target nodes and channels? or a list of all the different tab objects? the tab button objects?

Thoughts:

well, its evening now and the tool seems to be working. I find that if I increase or decrease the number of keys after creating a curve, I sometimes need to execute the “create Keys” function twice for the extra keyframes to take effect. I guess thats to do with when the “getNumber of Keys” function is executed. But, I am happy with the tool so far. I think its good enough for testing out to see if its fun, useful, relevant.

---------

06_08_13

Thoughts:

looking back over some movies it seems like the most interesting might be the loRes procedural whippet animations. Perhaps because they are “not all given”. The eye somewhat fills in the gaps....

Maybe when (if;) I finish this autoKeyframe thing, I should/could have another go on the lo res whippet.

It seems that I can become obsessed with working out (solving) a particular coding problem. But the work that comes from that problem is not necessarily interesting.

Maybe the problem is that when I make a new tab it also has to be a unique instance of the class, How could I do that from the button??
--actually it doesnt seem to matter, it all seems to be working as long as the user selects a new channel for the new tab name

---------

still trying to write this class so that I can have multiple Auto keyframe tabs. it seems that even if I make the tab a class and make a second instance of that class it says “Too many children in layout: buttonsRow”

A POSSIBLE WORK AROUND - can I create multiple instances of the tab object from the start?

yes that seems to work
It seems that the tabs need different names --- maybe I can build a UI within a UI---- this seems to have worked for “create new tab” as long as user selects an object and channel before creating the tab.

“AutoAnimTab_36.py” -- this file works :)

---------

How to do the “Refresh all Keys” command?
I will try adding it to the AutoKeyframeTab class.

VERY DISAPONTED AND FRUSTRATED ----------- it seems that it is not finding the correct text fields.... How to deal with this namespace business????

It might work if I give each text field a distinct variable name (within the class; so “self.jhhjg”)
so this seems to mean that I need to decide how many keyframes from the start....
Because I dont know how to assign a distinct variable in a loop

05_08_13
should I use a dictionary to attach key numbers and keys values?
A: it was quite easy to create the dictionary.
Currently its not used in my script But it might be a useful way of storing default values
ie. write teh dictionary to a file and read it from the file.

>>>I have just noticed that my multiple target expressions dont work :].........
$offset = pCube1.offset3;
$multiplier = pCube1.childMultiplier;
$delay = -1 * (pCube1.childDelay * 3);
$time = frame + $delay;
$value = `getAttr -time($time) pCube1.rz`;
$ampIncrease = pCube1.ampIncrease;
$increaseAmmnt = $ampIncrease * (3- 2);
if ($ampIncrease == 0)
{
 $increaseAmmnt = 1;
}
if ($ampIncrease > 0)
{
 $increaseAmmnt = $ampIncrease * 3;
}
pCube4.rotateZ = (($value * $multiplier) * $increaseAmmnt) + $offset;
I have changed the expression and the above seems to work...
Unfortunately you cant animate the “Amp increase from 1 to -1 without getting jumps and flips....

DICTIONARIES: I have created a dictionary for key values. I think that might also be a good way of storing default values????

driverKey(s) = 0
time offset = 1
value mult = 2
value offset = 3
tangent in = 4
tangent out = 5

QUERYING keys-------
mc.keyframe("pCube1.rz", query=True, index=(2,2))
seems strange that the index value is given as a list of two integers....”Note that the indices are 0-based”; I dont
know what that means....
the above statement returns the time of the key of that index.

How to create a new tab????
If I try to create a new tab from within the class, I seem to get problems with the naming of rows etc.
I deleted the row names but that doesnt help;
eg I get errors like # Error: RuntimeError: Too many children in layout: rowLayout84 #
--- do I create the new tab from outside the class?
--- or name the rows etc according to the node and channel for example?
---------- it would be good to rename the tab with node name and channel
-----MAYBE changing the name of the tab would alleviate the naming issues?
04_08_13

Problems arising during production;
why is my conditional expression not working???

>>> $increaseAmmnt = $ampIncrease * $nodeSub;
so it seems to be the first statement inside the loop that doesn't work...
... A = it seems that in an expression that uses "{" I have to use the other string formatting with "%" etc.... will try that now.
...

YAY!!! I finally got it working like this;

exprString = "$offset = %(%driverNode)s.%(%driverOffset)s;
$multiplier = %(%driverNode)s.childMultiplier;
$delay = %(%driverNode)s.childDelay * %(%nodeName)d;
$time = frame + $delay;
$value = getAttr -time($time) %(%driverNode)s.%(%driverChannel)s;
$ampIncrease = %(%driverNode)s.ampIncrease;
$increaseAmmnt = 1;\n\nif ($ampIncrease == 0){\n$increaseAmmnt = 1;\n}\nif ($ampIncrease < 0){\n$increaseAmmnt = $ampIncrease * (%(noOfTargetNodes)d­ (%nodeIndex)d);\n}\nelse{\n$increaseAmmnt = $ampIncrease * (%(nodeName)d);\n}\n%(targetNode)s.%(%targetChannel)s = ($value * $multiplier) * $increaseAmmnt + $offset;"
%(driverNode)': driverNode, 'driverOffset': driverOffset, 
'nodeName': nodeName, 'noOfTargetNodes': noOfTargetNodes, 'nodeIndex': nodeIndex, 
'driverChannel': driverChannel, 'targetNode': targetNode, 'targetChannel': targetChannel}

I now have the two first tabs working (single expr and multiple expressions).

--- Why cant I use a default arg like "parent=self.newTab" when defining a function? I have declared "self.newTab" in the __init__ function so I dont understand why I get the error "NameError: name 'self' is not defined #? ????

--- trying to add columns to the row layout so that the number of columns = number of keyframes. But I cant seem to edit the number of columns. Can I add a second row layout to the first row layout?

02_08_13

Problems arising during production;
I cant seem to execute two commands from a button.
Even when I make a function that executes 2 commands (ie this double command function doesn't work)

How to get the expression name????

Technical findings;
if I need to access values in other functions I need to declare them in __init__, eg;

self.textField_Target = "

Thoughts arising during production;
I am realising the importance of designing the script in advance
I am now using classes to reuse the code even more. One thing to remember with Maya Python and UIs is how to execute functions with a button; you have to use “from functools import partial” and then “command=partial(blahblah, arg1, arg2, etc)” for example, “command=(partial(tab2.getText, 'input1'))”

30_07_13

General steps;
> create tab for setting keyframes according to a driver channel
> later after using this I can create and store presets
    maybe I have a tab for creating and storing presets? such as slow run
> create tab for linking a channel to music (again add a Mult, Delay, and Offset attributes)

Questions arising during production;
    can I use single target for left foot links? YES
    can I do a decent run cycle using this method????
SHOULD I change the multiple connections (children) so that each is connected to the previous?

Thoughts arising during production;
    It strikes me that what I’m doing when I write tools for myself (create blocks of reusable code) is that I am operating within the logic of representationalism (am I creating immutable mobiles as described by Latour?). I am reducing the world to data, I am downplaying variability [for example in the way that I bumble through the animation of the whippet; connecting one node (via a relationship) to another...and then I dissect or recognise what I have done and I reduce what I have done to reusable parts. For example it feels like I have linked the left foot cntrls to the rt in a particular way,, but is this process essentially the same as the way I have linked the back_cntrl to the main_cntrl?
    How can I argue that my work is against representationalism? is it the way that I use these tools? is it the way that I engage with the essential (is there such a thing?) nature of the digital? do I play with that essential quality in a way that might expose it? am I modelling the world (mapping it in a new way) and then sort of exposing the gaps? or allowing the unexpected to intervene?
    Maybe I want to set up systems and then break them? or at least set up systems and then see what surprising outcomes emerge.
    MAYBE I create this UI as a class and the whippet UI as an instance of that class?

29_07_13
had a thought about how hard it is to watch/ study/ experience Ginger while modelling or even texturing her using 3D software. When I’m sketching her I am also relating to her, being surprised by her, wondering what she experiences, wondering at how I experience her. But is the difference between sketching and painting the same as the difference between sketching and making in 3D?
Well I have done painting still lives which are about that same 3-way relationship, but I havent done 3D software still lives or the equivalent; could I ? or is it more interesting to pursue a conversation with the materiality of the software/ packages of code/ code/ translating data...

Today I will finish my UI for creating offset (secondary animation) expressions.
Also still thinking of using music as incoming data.
It would be a way of using music as inspiration, and then somewhat playing, or improvising from there.

Questions arising during production;
Thoughts arising during production;
From my expressions UI ⇒ I could also duplicate objects and translate them? I’m not sure what I mean by that.

— Limitation already about the Child Multiplier. A negative number for this inverts the child movement. I can increase the value down the child chain, but I can’t decrease it.

Functionality fixes;
> check if attributes exist - DONE
> add other attributes to second button - DONE
this working version is called delayExpressionsUI.py

Functionality to add in future is;
> input music track
> input what to drive with it [create keyframes; at what intervals]

NOW more whippet animation -

INTENTIONS for this session are;
to animate feet
create a .stl file for print
create a pepakura file for paper construction ⇒ should they be the same pose?

TO DO;
in order to work toward having one node to animate (with the others following)
> link tail animation to MAIN;

this includes Delay, Multiplier, and Offset attributes [conceivably the tail could still be animated with keyframes only by making the Multiplier 0, and animating the offset value].

SUMMARY:
I enjoyed animating today with this tool and the whippet keyframe expressions for the right feet.
I will now fiddle with these keys by hand and at the same time try to clarify the relationship (expressions between the nodes as well as the offset and delay values for each node).

I think there is three types of controls and relationships;
multiple target expressions
single target expressions
set keyframes according to curve

to do;

>>> tab for feet right keyframe creation.

>>> link left feet to right feet

………………
29_07_13
It seems that my process in 3D is to start a project and make tools (blocks of code) as I need them. I then rationalise the process, i.e. work out what I have done, and then make tools (eg a UI or window) to do it again (ie use the same process on different objects and in different projects). What happens when I use the same process again (i.e. when I interact with my tools)? is it still interesting? do the tools then allow me to push the process further, to evolve it? or is the interesting part making tools to suit my process which has emerged?
I notice that I haven’t used my ramp shader script again. I haven’t used my ModellingAsAnimation UI again. But I am certainly planning to.
26_07_13
I feel the urge to create a relatively robust UI so that I can reuse the tools/ functionalities that I have created. And so that I can see where I can push them.
I am making a window to create secondary anim expressions....
> enter the driver node and channel
> create attributes on driver node
> ?? enter expression name [could the expression get printed out as it executes?]
> select the driven nodes in order
> create expressions

24_07_13
I found that using only procedurally generated curves was limiting, particularly because changing the frequency of the sine curve produces bizarre results.

This week I have been writing functions to create;
expressions linking cntrls to MAIN cntrl
expressions to duplicate and delay or predict movement from cntrl joints further down the chain.
keysframes on Right foot cntrls;
   value linked to;
      Stride length
      step height
      offset value
      MAIN_translateZ
Time/ location linked to;
   ground length
   offset time
   MAIN_translateY
expressions linking left foot to right foot

Now I will spend 1 hr trying these functions on my original rig.
I am working with “rig_050” but I seem to have messed up the names of the joints and groups.
Might need to go back and work with a prior rig.

I like the way that it looks like the whippet is struggling (reflects my struggle to get it all to work)....

19_07_13
summary of today;
I made a slow mo playblast. It looks fine slow. interesting how it all screws up when I increase “Speed” Attribute. Is it just because the splines are a bit out? or is it cause its always unpredictable changing frequency? ---- Gina its probably the AMPLITUDE. You have to change this as well....
I now think it needs to be a combination of expressions and generating keframes. And I need a UI to regenerate keyframes. So I could;
**add second feet**

**add change for amplitude (move keyframes according to speed)**

Also the expressions in general seem unstable. Maybe because names are not exclusive? (when I duplicated the whole rig it got the feet confused. Also sometimes I repeated expression writing and got different results.

Maybe import several of these whippets and render a movie

...........

Gina it seems that you can offset the time stuff using the API;

...Inside your compute function you just get your data via an MPlug.asMObject(MDGContext(MTime)) instead of MDataHandle, which can force the DG to evaluate at given time - for geometry it will be something like this:

```cpp
// get current time and offset it
MTime currentTime = MAnimControl::currentTime();
MTime evalTime(currentTime);
evalTime += offset; // some value to offset with
// create the context
MDGContext ctx(evalTime);
// find the plug to the input attribute
MPlug pGeo(thisMObject(), geoAttribute);
// get the data as MObject in the current time context
MObject inGeoData = pGeo.asMObject(ctx);
// do stuff... i.e.:
MFnMesh geoFn(inGeoData);
```

This has turned into a procedural animation project. I’m currently working on a low res, entirely procedural whippet...

just had a thought about using a ramp....hmm didnt work because its not time based

**REAR FOOT**

probs- delay seems to through it out of whack cause amp also has to move

```cpp
translateZ =
$delay = footBack_R_P.delay - 5;
$offset = footBack_R_P.offset;
$frequency = torsoP.Overallfrequency ;
$amplitude = torsoP.amplitude;
$time = frame + $delay;
$addZ = `getAttr torsoP.translateZ`;
footBack_R_P.translateZ = (((sin((time + $delay)*$frequency))* $amplitude) + $offset) + $addZ;
```

```cpp
translateY =
$delay = footBack_R_P.delay - 0.45;
$amplitude = footBack_R_P.amplitudeY;
$offset = footBack_R_P.offset + 2;
$frequency = torsoP.Overallfrequency * 2;

$sinValue = ((sin((time + $delay)*$frequency))* $amplitude) + $offset;
footBack_R_P.translateY = max(0,$sinValue);
```

AmplitudeY =

currently a keframed curve … almost sine but “lop sided”.

**FRONT FOOT**

```cpp
translateZ =
$delay = footFront_R_P.delay + .8 ;
$offset = footFront_R_P.offset + 1.9;
$frequency = torsoP.Overallfrequency ;
$amplitude = torsoP.amplitude - 2.8;
$time = frame + $delay;
```
$addZ = `getAttr torsoP.translateZ`;
footFront_R_P.translateZ = (((sin((time + $delay)*$frequency))* $amplitude) + $offset) + $addZ;
translateY=

hmmm...cant seem to do the translate Ys
$delay = footFront_R_P.delay + 1.2;
$amplitude = footFront_R_P.amplitudeY + 1.7;
$offset = footFront_R_P.offset + 2;
$frequency = torsoP.Overallfrequency * 1.6;
footFront_R_P.translateY = ((sin((time + $delay)* $frequency)) * $amplitude) + $offset;

CONTROL
control.speed

......
09_07_13

General steps;
- locate bones, best place for rotations (note; it is possible to move these once bound).

Questions arising during production;

Thoughts arising during production;
- Materiality - saving and retrieving digital documents - I am making changes to the dog model now in a file that I have named “rig_00#”. it makes me wonder if I will find this new model later. And that line of thought makes me realise how important the material I archive is. If I have done so much reading, but do I forget what I haven’t made notes on or recorded links to? In this age of digital files (with lots of iterative saved documents or animation files) and lots of digital information, is it even more important (or influential) what we choose to record, how we structure our information?
- interesting that I have made her without her collar.

......
08_07_13
I used the website below to convert an obj model into an stl file. It triangulated the model and I had to rotate it 90 degrees on the X axis.
http://www.greentoken.de/onlineconv/

Now I am using the skeleton from the “dog” Modelling as Animation experiment..
Want to add two pivot points - front and back

have bound the mesh...its not working well... I will now make low res (cut up mesh and parent under bones). Then animate and see what mesh looks like...

Rig problems...very tiresome.
TOMORROW I will spend one day building a rig...

......
07_07_13
Thoughts and questions arising during production;
maybe a “realistic” model that looks convincing when still, but it looks “convincing-when-still” when moving...
-   do convincing 3D models have to “make sense”. I find myself trying to match the logic of bones, skeletons

05_07_13
- made a script to automate my poly modelling setup
- I am modelling over my side sketch
aiming to get edge loop flow, quads, working from low res to higher. save model and screen shots on the way.
- I am also thinking about how the model will need to deform... ie how a dog moves... will it bend its neck and straighten head to look up.
- I like the initial stages of modelling when moving one vertex makes big change to the shape; the personality of the dog.

General steps;
- download and study anatomy drawings

Questions arising during production;
- can I get more than I give?
- “that looks good...how far can I push it until it breaks (ie. doesnt look better and better but actually looks bad, loses the character that was emerging)

Thoughts arising during production;
- is there the equivalent of working like Rodin punching clay

04_07_13
Aims and intent of this project;
I am aiming to use the “default” Maya workflow to depict a particular whippet as perceived by me at particular moments, in particular contexts. The “default” Maya workflow I mean using mainly the UI tools to model, texture, rig; make one model that achieves a variety of poses.

Subjective visual qualities and context specific visual qualities will be achieved via an “objective” approach (breaking the qualities down into attributes).

**General steps:**

Look at previous sketches. Looks at pics from the net of whippets and whippet skeleton. Draw relaxed pose with reference to my sketches and pics of whippets online.

**Questions arising during production:**

Should I collect photos from the net even though I know that they will effect my creation in other mediums?

Or should I try primarily not to use photos? Instead sketch relaxed poses poses....

**Summary:**

I could have taken a photo of her but drawing the relaxed pose was a way of familiarising myself with the dog’s form.

I’m looking forward to modelling.........Already I can imagine crossovers and deviations;
- a whippet who shakes off her keyframe animation and moves via motion capture, ....
def getName():
    nodeList = mc.ls(selection=True)
    node = nodeList[0]
    channel = getChannels()
    name = "{node}.{channel}".format(node=node, channel=channel)
    # print name
    return name, node, channel

def addAttribute(node, attributeName, defaultValue=0):
    node = node
    attributeName = attributeName
    status = mc.attributeQuery(attributeName, node=node, exists=True)
    if status == False:
        mc.addAttr(node, longName=attributeName, keyable=True,
                    attributeType="double", defaultValue=defaultValue)
    else:
        pass

def getChannels(*args):
    getChannelBoxName = mel.eval('$temp=$gChannelBoxName')
    chList = mc.channelBox (getChannelBoxName, q=True,
                            selectedMainAttributes = True)
    if chList:
        for channel in chList:
            print channel
        return channel
    else:
        print 'No channels selected!'
        return ''

def getChannelList():
    # very like teh procedure above but returns a list of channels
    channleList = []

    getChannelBoxName = mel.eval('$temp=$gChannelBoxName')
    chList = mc.channelBox (getChannelBoxName, q=True,
                            selectedMainAttributes = True)
    if chList:
        for channel in chList:
            print channel
            channleList.append(channel)
        return channleList
    else:
print 'No channels selected!'
return ''

class AnimExpressionsTab:
    def __init__(self, tabLayout, tabName, tabDescription="", multiple=False):
        self.cw1 = 100
        self.cw2 = 200
        self.tabLayout = tabLayout
        self.tabName = tabName
        self.tabDescription = tabDescription
        self.multiple = multiple

        self.newTab = mc.columnLayout(tabName, adjustableColumn=True,
                                      rowSpacing=20, parent=tabLayout)
        self.textField_Driver = ''
        self.textField_Target = ''
        self.expressionName = ''
        self.expressionString = ''

    # Mutators
    def populateTab(self, button1='add driver', button2='add target',
                     insertText1="driverNode.driverChannel",
                     insertText2="targetNode.targetChannel"):
        buttonWidth = 150
        row1 = mc.rowLayout('row1', parent=self.newTab)
        mc.text(self.tabDescription, parent=row1)
        row2 = mc.rowLayout('row2', numberOfColumns=2,
                            columnWidth2=(self.cw1, self.cw2), parent=self.newTab)
        self.textField_Driver = mc.textField("textField_Driver", w=200,
                                             insertText=insertText1)
        mc.button(button1, w=buttonWidth,
                   command=(partial(self.populateDriverText, self.textField_Driver)))

        row3 = mc.rowLayout('row3', numberOfColumns=2,
                            columnWidth2=(self.cw1, self.cw2), parent=self.newTab)
        self.textField_Target = mc.textField("textField_Target", w=200,
                                             insertText=insertText2)
        mc.button(button2, w=buttonWidth,
                   command=(partial(self.populateTargetText, self.textField_Target)))

    def populateMultiChannelTab(self, ):
        buttonWidth = 150
        buttonHeight = 30

        rowl = mc.rowLayout('row1', parent=self.newTab)
        mc.text(self.tabDescription, parent=rowl)
        row2 = mc.rowLayout('row2', numberOfColumns=2,
                            columnWidth2=(self.cw1, self.cw2), parent=self.newTab)
        self.textField_Driver = mc.textField("textField_Driver", w=200,
                                             insertText="driverNode.driverChannel")
        mc.button('add driver', w=buttonWidth,
                   command=(partial(self.populateDriverText, self.textField_Driver)))
Experimental animations: results as a list

This research has identified 3 general themes around which the animations, strategies and tools described in section 4 of this document have tended to coalesce. These themes include Emergent Content, Perceptual Experience and Mapping Process to Outcome.

Each creative strategy and custom tool addresses one or more of these themes and the list below indicates how strategies, tools and themes are aligned.

- **Encouraging Emergent Content**
  - Playing with Software
  - Playing with History
  - Animation as Relation
  - Auto Expression
  - Auto Keyframe

- **Exploring Perceptual Experience**
  - Working from Sketches
  - Modelling as Animation
  - Working from Life
  - Plein air Still life
  - Colour as Light
  - Geometry as Shadow
  - Shadow Play

- **Mapping Process to Outcome**
  - Modelling as Animation
  - Plein air Still life

The following list indicates how these themes relate to methods suggested by phenomenologist philosophers as well as methods suggested by the work of other creative practitioners discussed throughout this document.

- **Encouraging Emergent Content**
Being attentively present in the domain of things
Listening to things and responding based on what feels right
Disrupting your own working process
Disrupting your own easy interpretation of an image
Exploring qualities inherent in 3D animation software
Exploring capacities of 3D software that extend beyond its design

- **Exploring Perceptual Experience**
  Returning to the world of perception
  Not accepting simulation tools as an explanation
  Interrupting habitual ways of seeing
  Making things strange and seeing the ordinary as extraordinary
  Exploring free perception and the experience of teeming
  Not pulling things apart in order to understand or describe them
  Beginning in poverty
  Avoiding animation pre-sets and taking a back to basics approach to 3D software
  Trying to forget what it is that you’re looking at and describing physical things as shapes and colours

- **Mapping Process to Outcome**
  Inviting a viewer to take up the gesture which created the work
  Make intuitive decisions evident in a finished animation
  keeping the work sketchy
  Excercising a tolerance for the incomplete
  Resisting the urge to set things straight and clean things up