



Kent Academic Repository

Wood, Aylish (2013) *Talking about Maya*. . University of Kent, 60 pp. pdf.

Downloaded from

<https://kar.kent.ac.uk/37675/> The University of Kent's Academic Repository KAR

The version of record is available from

<http://www.kent.ac.uk/arts/staff-profiles/profiles/film/wood.html>

This document version

UNSPECIFIED

DOI for this version

Licence for this version

CC BY (Attribution)

Additional information

Versions of research works

Versions of Record

If this version is the version of record, it is the same as the published version available on the publisher's web site. Cite as the published version.

Author Accepted Manuscripts

If this document is identified as the Author Accepted Manuscript it is the version after peer review but before type setting, copy editing or publisher branding. Cite as Surname, Initial. (Year) 'Title of article'. To be published in *Title of Journal*, Volume and issue numbers [peer-reviewed accepted version]. Available at: DOI or URL (Accessed: date).

Enquiries

If you have questions about this document contact ResearchSupport@kent.ac.uk. Please include the URL of the record in KAR. If you believe that your, or a third party's rights have been compromised through this document please see our [Take Down policy](https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies) (available from <https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies>).

Talking about Maya

interviews with users of Autodesk Maya
a report by Dr Aylish Wood



acknowledgements

**Thanks to everyone who freely
gave their time to talk to me.**

**Listening to people speak about
their work has been endlessly
fascinating.**

foreword

What

‘Talking about Maya’ is a resource for people who want to know more about how animators create using software. The document brings together a detailed sample of answers given in response to questions focussed on using Autodesk Maya. The interviews were undertaken during an **Arts and Humanities Research Council** funded project exploring computer-generated (CG) animation. Part of the project involved asking animators (this loose use of the term ‘animator’ covers modellers, riggers and animators) to describe their experience of working with software. The reason for choosing to talk about Maya is that the software remains widely used within visual effects, animation studios, many games companies and data visualization studios. Though focussed on Maya, many of the comments are relevant to animating using 3D animation software more generally.

Consisting of five main sections, the document is structured to expand on responses to five key questions. These sections can be read in any order. Each opens with a brief outline of the rationale behind the question, which is followed by the diverse range of answers given by Maya users. The answers are placed in subsections to display the variety of responses around specific points, and also to provide a structure to the flow of the material.

Why

My purpose in carrying out these interviews was to find out what animators think about the creative process of using software to make animations. The interview questions were shaped by my particular interest in how users themselves explained their work with software, in this instance Autodesk Maya. I was not aiming to talk about specific images created for a particular game, animation or data visualization, something that is more usually associated with the publicity for a new feature film or game. Instead, the interviews sought to get users of software to explain what they do when using software. The answers brought together in this document explore five issues. The first two are concerned with matters central to animation: space and time. The last three are more abstract, and more explicitly draw out the ways in which users engage with software. The questions were framed around a challenge to the misconception that the automated processes of software did the ‘doing of animation;’ the idea that users ‘negotiated’ with software (in the sense of tussling or battling with it); and, finally, considering what users had to say about the technical aspects of using software in their creative work.

The answers to particular questions can be found on the following pages:

space	08
timing	20
who does the doing?	30
negotiations	36
technically speaking	46
final words	60

Who

What follows makes no attempt to be a 'how to' document for working with Maya, and neither does it presume to have all the answers. Instead, it is a synthesis of answers from 23 individuals working in diverse sectors of the animation industry. The individuals interviewed worked in larger visual effects houses (The Mill, Double Negative, Prime Focus) and games studios (Blizzard, thatgamecompany), medium sized animation studios and educational institutions (Digimania; The Arts University at Bournemouth; Institute of Creative Technology, USC), data visualization studios (GST, Inc. - NASA/Goddard Scientific Visualisation Studio; Walter and Eliza Hall Institute, PinPoint Visualisation). In addition, some people interviewed teach Maya and others were still in training programs, while others worked independently. The level of experience ranged from less than a year to 15 years of working with computer animation software, often though not exclusively with Maya. Where agreed with individuals I have used their names, otherwise I have used initials. Where requested, anonymity, including work place details, is maintained. I would also like to thank Escape Studios and the CG Society for generously allowing me view their training videos.

Finally, the answers given by participants in these interviews are not impartial. They come from people whose enthusiasm about making animations informs their discussion of the software. Just as any statement always comes from a particular position, both the questions and answers given here generate a perspective. For instance, while there are many things to say about software, since my questions were framed around how users engage with software, the answers begin from that position.

The responses to my questions are also outward facing descriptions of how modellers, animators and riggers work with software. By outward facing, I mean they describe a process of working to someone outside the community of Maya users. Some of the ideas expressed are examples of embedded production knowledge, the learnt practices of working within a studio system and also using Maya within a production pipeline. These elements of the answers describe the work and roles of modellers, animators, and riggers, and their relationships to one another. These are revealing for people outside of the animation industry as they provide insights into how modellers, animators and riggers situate themselves within the production system where they work. Such knowledge is interwoven into the direct responses to questions about engaging with animation software. Both of these features of the answers provide a fascinating resource for thinking about how computer generated animation happens.

space

do you think differently about space?

Asking this question allowed me to explore whether 3-D animation software made a difference to how a person approached creating an object when working in digital or computer generated space. Ways of answering this question included referring to the imaginative freedom that comes when working with software, the different physicality of digital objects, and the sameness of 3-D dimensionality in both physical and digital space. People's answers are frequently framed through comparison, usually with different kinds of animation techniques, such as drawn 2-D or stop-motion.

Overview of replies

- you can do anything since you are not bound by the real-world physics of gravity or the weight and density of materials
- it is necessary to think around all the angles in 3-D, which is different to 2-D
- engaging with 3-D occurs through both the immediacy of modelling and also absorption with the scenario being animated

Details

Answers to the question of whether or not Maya makes a difference to thinking about space suggest that many users understand themselves to be doing the same spatial

things only differently. That is, they create shapes and movements through the parameters of the software, but in ways that remain grounded in the principles of animation. One person's answer captured this very nicely: **“Well, I guess the difference would be the difference between playing guitar and playing piano** [Jacky Ke Jiang].” Through this analogy of making music with different instruments, the point is made that users are still creating animation, but the look, movements, and textures vary with the technique used.

A big difference between 2D animation and 3D animation is you basically have to take into account the 360 view [XZ].

Working with those objects in three-dimensional space, how you animate characters within Maya, you are having to think in 3 dimensions like you would with a puppet [Paul Hilton].

The way I work with polygons does affect how I see things [David O'Reilly].

Working with Maya is a lot like Lego building on screen...it's kind of done in front of you [XY].

“Well, I guess the difference would be the difference between playing guitar and playing piano.”

To tease out the insights offered by the diverse responses focussing on how users think about space, their answers are categorised below in terms of ‘building space,’ ‘thinking in 3-D space,’ and ‘engaging with the space of 3-D.’

Building space

Responding to the question of whether he thought there was a difference when building space within a computer, animator Aaron Chan comments:

I think it's very similar. What's great is that, as a 3-D artist, we're not bound by materials or required to do real physical labour. If the artist wants to move a mountain, they can. With a good amount of research, you can really create whatever you want to create. One of the best parts about 3-D is having the ability to build a world or bring something to life. That's the best part. [Aaron Chan].

“Whether it is in 2-D or 3-D, or stop motion, you can do whatever you want. Whatever you can imagine you can do it.”

Aaron's remarks suggest that building in digital space is the same but different. The dimensionality is the same but being unbound from material limits creates a different sense of freedom. Echoing the view that one of the liberating aspects of CG is not having to deal with the weight of materials as one does with a physical model, generalist Mike Kilminster says: **“I can create whatever I want in Maya and then I can make the rig fit. Where as in real life I've got to construct the rig and then get the puppet built onto that rig.”**

Hunter Grant makes a broader point in saying that the freedoms offered by animation can be found in all techniques, but he also indicates that 3-D software is continually developing, and so more ways of achieving a creative solution are constantly evolving:

And that's the beauty of animation in general. Whether it is in 2-D or 3-D, or stop motion, you can do whatever you want. Whatever you can imagine you can do it. And that is what is really cool, especially when, in 3-D, now the technology has got to the point where if you want to do something, there is a solution. It might not be an easy solution, but there is a way to do it. And so that's the fun part, you really can kind of go nuts [Hunter Grant].

“You think you’ve got it in your head, but when you get in on paper it changes. Same thing happens in animation software.”

The freer possibilities of CG space described above are not only useful for creating fictions. For people working on molecular movies and protein visualizations, using 3-D software allows protein structures to be built and also manipulated in ways that allow for hypotheses testing, or modelling ideas about how a protein might react to particular cellular events:

So what Maya allows you to do is to visualize these structures but manipulate them in any way you please, which I think opens up new possibilities in terms of visualizing hypotheses, so you don’t face the same limitations as you have with a scientific software [XX].

Drew Berry describes animation as being like other forms of drawing, in that the very act of making or drawing an object reveals it to the artist anew.

Really, as far as the nature of the content I’m dealing with, animation really isn’t any different from drawing with a pencil and paper. It’s really the action of creating an image that reveals. It alters your thinking about it. You can have an idea in your head, but when you actually start to create a visual of it, or start to sketch the idea down, your ideas will change because you start to really see it. You think you’ve got it in your head, but when you get in on paper it changes. Same thing happens in animation software. I try to not have a preconceived notion when I create. I read the literature, it’s full of verbal descriptions and lots of forms of data, to describe what we think is going on at the very small scales beyond what we can actually see. And in the act of creating the visualization, the animation reveals. I rely on that to show me what the system is like [Drew Berry].

But, even as software frees up imaginative potential in both fictions and visualizations, it also can demand more effort. The following compares Maya with 2-D and stop-motion puppet animation to explain how some things in 3-D are not always straightforward:

One thing that is interesting about 3-D versus 2-D, or even stop motion, is that some of the easiest things to do in 2-D are some of the hardest things to do in 3-D. If I need a character to grab a cape, or grab someone's shirt, in 2-D you just draw it. No big deal. But in 3-D, you are probably generally grabbing nothing. And then you have to have a simulator go in and make that work with your animation, so it kind of goes back and forth. Okay, your grab really isn't working because it's making the simulation go weird, so you need to fix this, this and this. Stuff like that, or stuff that is fire, or effects, or it is wet, that's where the differences really become involved [Hunter Grant].

Two points emerge from this focus on building space in 3-D: users of Maya have the sense that they can do anything with the software, but at the same time freedom comes expensively in that gravity, points of contact, or particle based effects and water, can be complicated to do well.

“The way I work with polygons does affect how I see things.”

Thinking in 3-D space

Where the answers focussing on building in space often revolve around imaginative freedoms, those touching on ways of thinking in 3-D space cover a range of different possibilities. They include the intriguing view that: **“The way I work with polygons does affect how I see things** [David O'Reilly].”

Others recount early experiences of learning to work in 3-D space, with modelers and animators recalling the adjustments they made to working in a CG environment:

I think the first thing you grapple with is just navigating in three-dimensions. I think we probably all take that for granted, but it does take a bit of getting used to. Being able to move around and stuff, with 3 buttons on the modeler and being able to move all around in that space [Martyn Gutteridge].

Another user outlines advice given by their tutor when dealing with the fun of being able to move anywhere and anyhow: **“Don't get caught up in just having the ability to go anywhere. Still think about what's the purpose of the shot, what you're trying to say, what you're trying to achieve** [Ben Thomas].”

This last remark draws attention to the ways in which conventions of shot composition matter, regardless of the technique. Lead animator James Farrington puts his experience of working with Maya, and over-seeing other animators working with Maya, in terms of film language and having a sense of when things should move to make the shot work:

The way you observe space, you work quite quickly just to put something in, and then once you've got your bones in there you will get quite into detail. And everything becomes very focused and so you're looking at your camera view and you'll be thinking about things happening, if that moves too early or that moves too late and shift that a few frames. I mean the language is a film language to an extent. As an animator you're giving your shot, and you'll work on that shot, it's what you focus on. As a lead animator I'm seeing a group of animators working on a series of shots, so when I'm in the lead animator role I'll be thinking about how all these shots linkup to each other [James Farrington].

“Don't get caught up in just having the ability to go anywhere.”

The importance of shot composition becomes most clear when animators talk about being aware of working within a 360° space. As a novice user of Maya, Mike Eckton comments: **“It is a good thing to have the three-dimensional camera. It makes you more aware of how your animation will look on all sides rather than one. Such as if you do something on one side, like make a person walk. You can make a person walk okay on one side, but if you turn it around you find his legs are going in the opposite direction and his hands are going in all sorts of different directions.”** XZ makes a similar point through a comparison between 2-D and 3-D animation:

“A big difference between 2-D animation and 3-D animation is you basically have to take into account the 360° view.”

A big difference between 2-D animation and 3-D animation is you basically have to take into account the 360° view. Whereas in 2-D animation you are locked into a plane so you can cheat or skip things, in 3-D animation you pretty much have to check it from all sides [XZ].

Checking from all sides is especially relevant for people working with sequences of animation for a game when the player is able to view figures from all sides:

If you are animating for film you do it in one camera view. That's what people see. But in the game a person can put the camera anywhere. It has to be all round. So basically when I animate I use the 4 viewports, front, back, and top view. That is why I have an extra monitor. I use all these cameras to see how things look. But in a film you don't have to, you just work on the one thing that the audience is going to see. But in the game everything can be seen everywhere [Navis Binu].

Thinking within the space of 3-D not only occurs in a 360° view but also through the more fixed view of a camera. For animators this can be a pleasure:

That's what I love about animation especially...usually you animate to a camera, so you have got your shot and if it looks right it is right. It might not be all right at a different angle, or it might look a bit funky, but if it looks right in camera that's it, that's right [KMcD].

“That's what I love about animation especially...usually you animate to a camera.”

Where 3-D animation often calls to mind three-dimensional figures and objects, descriptions of working with the user interface (UI) bring different kinds of configurations of space and spatial thinking into the foreground. The graph editor used by animators provides insight to an alternative way of working with space. James Farrington describes his working set-up on Maya's UI as a camera view, graph editor and a perspective view, allowing him to look at both the dimensional model and its movement as a set of curves:

Whereas, what happens now is that you think in all angles around the object. So for instance, the set-up I like on my screen is to have the camera view, what's going to be seen, and underneath that I have my graph editor, so I can see all the curves, and then I have the big perspective view that I'm constantly moving around, and the model to see what it looks like from different views [James Farrington].

Focusing entirely on the graph editor, Maya user and teacher Paul Hilton comments on the fact that even though the graph editor shows what's happening in 3-D space, it does so in a very flattened and linear way.

Learning how to think through and work with this very different configuration of space is part of animating a scene:

The graph editor...you're seeing the actions happening in three-dimensional space, and elements moving around in three-dimensional space, but illustrated in a very flat linear way. Well, it couldn't be any flatter. And interpreting that, after a while you can see an issue in a piece of animation, and you know what you're looking for in the function curves [Paul Hilton].

“You're seeing the actions happening in three-dimensional space, and elements moving around in three-dimensional space, but illustrated in a very flat linear way. Well, it couldn't be any flatter.”

Three main points emerge from these answers: thinking in 3-D space requires animators to be aware of all of the angles; creating models for games and animations have different constraints; and, creating movement in 3-D space also involves seeing movement as flattened curves, something that might be unexpected to people unfamiliar with animating.

Engaging with the Space of 3-D

As well as thinking about the different constraints of shot composition in games and animation, and the abstracted space of a graph editor, several people spoke about how they engaged with the modelling and animating space of Maya's UI. Working with models involves an engagement with space, in the sense of being able to go into and out of the detail of the model. As Barry Sheridan says: **“I still like having something 3-D in front of me and being able to spin it around, I still like all that stuff. I still like the immediacy of being able to manipulate something seemingly floating in front of you.”** For other users of the software, modelling doesn't feel as immediate:

“I think you have to absorb yourself with the scenario, the performance, and the acting. The physical actions that happen inside a space, you need to be a bit more in tune with it.”

I think sometimes perhaps when I'm animating I feel a bit more connected with the space because you have to be aware of space. Spatial awareness I suppose really. I get absolutely absorbed in what I'm doing. I love when I'm actually working [on a building a model], producing stuff. It's brilliant. But I guess I still feel a bit like I'm looking through a window. You've got an array of buttons in front of you, and a window that you're looking at, and what you're dealing with. It's similar to one of those things where you put those big gloves on and you're working in a glass cabinet with volatile substances...it is like that. Looking through the plate glass [Paul Hilton].

Having made this comparison to a fume hood, Paul goes onto to compare his experience of modelling with animating a scene: **“I think you have to absorb yourself with the scenario, the performance, and the acting. The physical actions that happen inside a space, you need to be a bit more in tune with it. It is not a conscious decision, I don't consciously make a separation. I guess it's just a feeling.”**

Summary

From these various examples, it seems as though people do think differently about space when using 3-D animation software. But this is due to software changing how space is made, rather than involving a fundamentally distinct conceptualization of space. The basis of this claim relies primarily on comparisons with other animation techniques, in particular drawn animation. Being aware of the three-dimensions of objects and figures in time and space is clearly central to working with 3-D animation software. The UI also offers different kinds of engagement with space, from the sense of immediacy when either modelling or animating, to a more abstracted awareness of movements depicted as curves. And finally, despite all the freedom, nothing comes for free in 3-D. Gravity, friction, and points of contact all have to be made, they don't just happen. Because of this, it's tempting to suggest that people who create animations have an enhanced understanding of the physicality of the world, even if they do tweak and twist it to fit an imaginary one.

timing

“time is your life. time is everything [Hunter Grant].”

To make things move in ways that are interesting, timing is everything. This section shows the ways animators think about timing, poses and performance when working with Maya.

Overview of replies

- timing brings animation alive
- poses are everything
- *you* have to tell the computer how to join things up

Details

With the principles of animation in mind, timing, or the speed of an action, gives meaning to movement. This meaning can be physical or emotional, conveying details about actions or the feelings behind those actions. In response to my question about whether Maya makes a difference to timing, animators say in various ways that the art of timing lies in two things: the pose and the arcs of movement. Maya’s algorithms interpolate from key frame to key frame to create the in-betweens. The work of the animator is to often work an organic feel into movements, or to create the kind of movements appropriate to the content of scene.

The responses discussed in more detail below focus on several different elements of timing: getting timing to feel right, the importance of pose and thinking about ‘performance.’

...we study the physicality of movements and so in a sense we are bound to reality if we want to create a believable performance. However, what makes animation great is the ability to push reality [Aaron Chan].

The computer doesn't understand the timing. We have to pose everything. And the poses are everything [Navis Binu].

“You are not, you know, creating these images in your mind, you are using some other tool set, a graph or boxes or squares or controllers, different things.”

Timing, pose, performance

In answer to the question of whether Maya changed the way she thought about timing, one person recalled her experience of learning to animate using Maya. She talks about not creating the images in her mind but on the screen by working with different toolsets. Being able to realise the movements she wanted involved gaining an understanding of how the software makes movement, how it generates the in-betweens:

I definitely recall that beginning with 3-D animation in school, it definitely adjusted my perception of timing for animation. Because this software translates the in-betweens for you and you need to look at different tools within the software, such as graph editor, or something like that. You are not, you know, creating these images in your mind, you are using some other tool set, a graph or boxes or squares or controllers, different things. So it did have an effect on where to place keys, where to place breakdowns for animation. And I do remember that translated very differently. So from an animation perspective, I guess, yeah, it changes the way you think about animation [Liz Skaggs].

The interplay of user and software is central, with animators commenting that they only have to specify certain areas of the animation, such as key poses, while Maya ‘does the rest.’ In filling out the details of this interplay more fully, people talk about the process of timing by explaining how they animate pose-to-pose or in step frames. In the following, James Farrington talks about his preference for blocking out a figure’s movements in simple translations to get a sense of timing, and then adding levels of movement until the figure comes to life:

**“It will suddenly
come to life.”**

I personally like working in levels of curves, working to the controls of something. So you know, a block will start off with basically just translating it around the screen, and seeing the timing of how long it will spend in each place. And then working through the character and the levels of how it will move, layering it until you're working at it from a global positioning, right down to fingers and links and tiny detail. I tend to find that what happens with working in this way, is that it doesn't look very good for a long time, and then it's almost like there's a point where you've got enough rotations and translations keyed in there, so that it will suddenly come to life and then you go through and can put lots of detail in [James Farrington].

In talking about blocking by using step frames rather than translations, XZ comments on the mismatch between what she imagined the animation would look like and how it appeared when splined, and having to get the timing she saw in her head back into the animation:

“The computer will just join the dots that you tell it to...”

Well, I mean there are a lot of times when you have planned an animation and it looks great at the blocking phase. Then, when you spline it and look at it, it has just lost something. You have to work to get that back in...Well, this used to happen a lot when I was blocking by doing step frames. You get really snappy timing and it looks great. I think mentally you are kind of filling in the blend but then when you actually spline it and everything is moving through time, when the computer is doing all the in-betweens and stuff, then you see that there are these problems, like you needed a breakdown there or something. So you have got to work to get that back, the timing that you saw before [XZ].

The two ways of animating described by James and XZ, though quite different approaches to timing, reveal the tactic of working and re-working the sequence until it comes right. XZ draws particular attention to the fact that computer is doing the in-betweening. Animation involves getting the right timing out of the software, working to get accelerations and decelerations combining with rotations, translations and/or scaling in any of the moving elements of a sequence. As Martyn Gutteridge comments: **“And obviously the computer is essentially doing all the in-betweening for you. And as an animator you have to control it, you have to tell it. The computer will just join the dots that you tell it to, that's exactly what the graph editor is basically. That's you telling it how to join up the dots.”**

The word timing sounds straightforwardly linear, a timeline showing movements across time. It is more accurate to say timings, since each scene consists of a whole series of timings, from the global movement of a whole body crossing space, to the local movements of individual limbs, figures, or eyebrows. Rather than a single line, the graph editor can look like spaghetti, a mass of wires running from one connection to another, timelines for every moving element of a pose.

The subtleties and details of timing in animation come from poses, and the kinds of poses possible in any piece of animation vary with the genre of the work. Visual effects work embedded in live-action scenes constrains stylistic choices to matching the look and movements of the live-action: **“If you are matching to a live-action actor, you still have to make his digital double as real as you can get it. Hopefully it fools the audiences. And that is where the real becomes involved... [Hunter Grant].”** Hunter goes onto to say the following about poses:

...whereas if you started to create the stuff that we do here, most time we go for a more real look, but it is not photoreal because none of what we create can exist. You know, yes we do make photoreal humans, but even those humans are not fully real. We might pull some of their proportions, or make sure that their faces are longer than normal. There is usually an extra something. Whether it is the movement, the texture, or the lighting, how it is all comped together, or the overall look of the piece [Hunter Grant].

Thinking about timing and poses goes hand in hand with the idea of performance, the performance of figures and other digital objects in the scene. Some animators talk about thinking their way into the scene to get the best out of a performance: **“I think you have to absorb yourself with the scenario, the performance, and the acting [Paul Hilton].”** As Paul continues, absorbing oneself into a performance can contradict other ways of working. A number of interviewees comment on the usefulness of the redo option on Maya: **“But I think it [redo] makes you lazy as well. Because you can be like, ‘let’s map something out really quickly and get a sense of timing.’ Which is fine, but I**

“There is usually an extra something. Whether it is the movement, the texture, or the lighting, how it is all comped together, or the overall look of the piece.”

sometimes worry that by doing that you are losing the connection with the performance. You have to live the performance, and let that just come out naturally in the buttons you are pushing, and the way that you're pushing them [Paul Hilton].”

“You have to live the performance, and let that just come out naturally in the buttons you are pushing, and the way that you're pushing them.”

Another interesting contrast lies between working with motion-capture data and key frame animation, and the following comments again reveal the idea of animators thinking into the performance:

When you are working with mocap data, I think it is not as intimate a kind of thing. You are correcting someone else's performance. But I think in the times when I have done my own [animations], like the key frame animation that I do, and I plan it and everything, I mean you have acted it out so in that sense when you look at the screen and what is happening you are kind of mentally reversing that to yourself. Whereas if you weren't connected to it that way, if you hadn't acted it out or it wasn't your performance, I mean you are pretty much looking at it the way it is on the screen. I mean you kind of put yourself in there when it is your performance, I think [XZ].

Though not couched in terms of performance, data visualization too offers the similar idea of interacting entities revealing themselves to the animator as they work through a scene. Below, both Drew Berry and XX (who work in different data visualization studios) talk about animating molecular interactions within a cell, which is based on data from a range of different sources. Drew remarks that animation software allow different forms of data to be brought together, including verbal descriptions, and 3-D molecular models, if they are suitable:

Animation software is now so sophisticated you can bring in all these forms of data and it will reveal itself to you. And in the way things are, objects in space, the way they can possibly interact, how they articulate and come together. Yes, it gives you a sense of the nature of what that thing is like. And it's really a stage for you to bring all these actors together and see what happens [Drew Berry].

Drew's reference to a stage resonates with the previous discussion of performance. In molecular visualizations it is also the case that from knowing how things interact, the equivalent of the timings, poses and performances of proteins and other cellular entities can be designed to act according to the narrative demands of the piece of work:

They get a lot of information from a large variety of sources in order to get information about where it is in the cell, how it moves around in the cell, what other proteins it interacts with, when it interacts with them, etc. And so basically, with the animation software you can create this sort of visual model of how you think it is moving, who it is interacting with, how they interact and what are the ramifications of this interaction [XX].

“And so basically, with the animation software you can create this sort of visual model of how you think it is moving, who it is interacting with, how they interact and what are the ramifications of this interaction.”

Thinking about time and software mostly revolves around finding ways of creating the physical and emotional meanings of a story or visualization through timing, poses and also performance. Interestingly, several animators gesture to the idea that the timing and interactions between entities emerge, revealed in the process of working a scene. It comes to be through the interaction with software.

Other time-based observations

“I mean we have to watch how many triangles we use, how many materials we use, what size textures are...”

Timing isn't the only way of thinking about time and animation. In terms of workflow efficiency and usability, the weight of data has an impact on time. Mostly, this is linked to real-time engines where data weight has to be low enough to allow rendering to take place in real-time: **“When you are outputting graphics for real-time, there are constraints, and the pipeline. There are working principles that are not found in other areas, so not like what cinema or TV have to deal with. I mean we have to watch how many triangles we use, how many materials we use, what size textures are...Barry Sheridan.”**

These restrictions do not operate in the same way in pre-rendered high resolution animations used in visual effects work, animation features, shorts or game cinematics: **“I guess if you're talking to a game person, there's a lot more precision because they have to optimize everything really quite thoroughly. Whereas when you work in visual effects, I'm quite lucky I can throw polygons at the problem. They have to be really, really clever at what they are doing, whereas I kind of don't [XY].”** But even so, data weight is an issue when animating a heavy scene, as scrubbing the timeline becomes slower and means that animators have to think around the problem in a different way. The following comments illustrate two animator's thoughts on this problem:

I suppose if there are limitations it is to do with speed. So if you have to finish a shot, and there is no way of taking things out of it to lighten the load of the scene, you've got to just work with what you've got. The most frustrating thing is something called scrubbing to timeline to actually play through. The ideal is for the animator to have that real-time, literally scrub scrub scrub and see how something is moving. But quite often you don't have that because there is so much going on in the scene that you are working on. That's probably the most common frustration. Because it does slow you down and it genuinely decreases the overall quality of your work because you don't have that malleable aspect that you usually have [Ben Wiggs].

**“And in some ways
you have to work
smarter.”**

If you're working with something that is geometrically intensive, you have to work smart. An artist should rely more on visualization to be efficient, making sure that every move they make is potentially a good one. Now, as software gets better and computers get faster, things become easier, but it also gives the artist the ability to put more detail into their work. It can be a never-ending cycle [Aaron Chan].

Summary

Timing in Maya draws on two sets of skills. The ability to makes things move in ways that are expressive, meaningful and also interesting, slow or snappy depending on the tone of the sequence, is essential for any animator regardless of the technology or technique being used to animate. When working with software such as Maya, the creative ability to animate well relies on also knowing how to control the software. As animator Ben Wiggs says: **“In that sense its like any art, you have your medium, but your medium can't do anything without your direction. It is such a wonderful tool, just as long as you keep it in check. Don't let it rule you.”**

who does the doing?

who does the doing?

John Lasseter has famously stated that the “term CGI is a misnomer - the computer doesn't generate the images. That would be like calling traditional animation Pencil-Generated Imagery. No matter what the tool is, it requires an artist to create art.” My question, ‘who does the doing of animation’ sought to tease out ideas about the ways in which automation influences animation. The responses that follow move quickly away from the anticipated answer (people saying *they* control the animation), towards the insight that using software is a process of working with and against the defaults of the software, a point that has already been encountered in relation to Section 2’s discussion of timing.

Overview of replies

- software is a tool and under the control of the user
- working with software involves making it do what you want by stopping it doing what it wants

Details

Without exception, everyone (modellers, animators, riggers) replied that they are in control of the software used in the creation of an animation. A number used the word tool as part of their description of Maya. The top two answers in the box below make clear that the user is in

“It sets something different in your brain.”

control. In contrast, the bottom two answers begin to complicate the answer, expanding out from the idea that a user is in control, to the sense that working well with Maya is a consequence of knowing what Maya can do, how it does things, and also being able to think in the ‘Maya way.’

It does what it does, what it's told and only that [Paul Hilton].

You make it do what you want it to do, so it is a tool. I mean if you didn't actually work with it, then you would be letting Maya animate for you [XZ].

I think I am definitely in control, in the way that I know what Maya can do. But I'm not trying to force Maya to do what I want do [Jacky Ke Jiang].

You can do anything, but you can't think of it in the same way as the real world. You have to think of it in the Maya way [Ady Holt].

While the notion that software is a tool for animators is not surprising, the more extended answers reveal insights into how people work with the software, often described as a process of working against the automation of the software, its ‘out-of-the-boxness.’

Describing her experience of learning to use Maya, rigger Liz Skaggs described the process as being **“like animating with boxing gloves.”** As she gained greater experience, she commented of the software that: **“it sets something different in your brain.”** These turns of phrase draw attention to the ways in which users of Maya work with the parameters of software, something that is drawn out more expansively in the following discussions of working against the software. These responses mostly refer to poses and movement as opposed to modelling.

Of poses and movement, users indicate that they operate within the envelope of the software's functionality, working productively and creatively with the opportunities offered, often problem solving around constraints. The points raised expose the idea that software automation doesn't have a 'feel' for what will make any piece of animation good. Software doesn't recognize the nuances of the principles of animation, the ways in which arcs of movement conjure a gesture, or how easing brings liveliness to moving objects. Instead, software works by plotting out a straight line between A and B according to a set of conditions. It is the animator who works and reworks that line to get the quality of movement that she or he seeking.

"I think it is fighting against the CG realm, it's about keeping it alive."

Commenting more broadly about working with Maya in this way, animator Ben Wiggs says of the software:

It all needs a large amount of artist input. Again I think it is fighting against the CG realm, it's about keeping it alive. In the end it's just binary code, and binary code wants to be uniform, monochrome, flat, simple basic. And I'm happy for that as that's what makes the software work so well. But, of course, what you're trying to create is more organic, it's alive, it's springy...I see that from an animator's view. Maya is such a wonderful tool that it allows me to do my job, but if you leave it to its own devices it will churn out very computer generated looking images. So it's important to keep it alive [Ben Wiggs].

While talking in a more detailed way about poses and movements, generalist and tutor Paul Hilton too talks about how good work comes from being exacting in ensuring that the poses are hit in the way that suits the feel of the animation and not the efficient calculation of the software:

“You know, however good the software is, it's not going to be up to doing it properly.”

Sometimes you go in and you set a number of poses, for a character, and then you look at the animation. And you think, well it's kind of hitting those poses, but I need to see that pose longer, or it's going in or out of that pose, and that's just purely because it is interpolating between those frames. It is doing what it does out-of-the-box between frames. You need to then go in and work on your breakdowns and in-between's or the function curves, to re-time the animation and to get it to do what you want it to do. It is breaking down the out-of-the-box-ness [Paul Hilton].

Paul goes onto to say that animating skill includes being self-critical enough to say that something won't do, so not allowing the software to dictate how an animation comes out. As he puts it: **“The fight can sometimes be just trying to manage those curves, get them to do exactly what you want them to do. Micromanaging the function curves.”**

Re-iterating this idea Mike Kilminster remarks: **“In my earlier stuff, it [the software] gave you a temptation to think you are done when you weren't. Because Maya will automatically in-between everything for you, there's a temptation to think ‘well it moves between there, it's done.’ But it's not. You know, however good the software is, it's not going to be up to doing it properly.”**

A different perspective emerges in relation to data visualization. Where software is used to realise the depiction of sets of data that have been generated from various kinds of instrument-based surveys, **“the thing that drives our visualizations is the data** [Alex Kekesi].” The remarks below flag up an issue that is true for all users of animation software, not just data visualization labs.

Along with the constant pressure of deadlines, a project is very often loaded with a need to achieve a particular look or effect, provide an explanation of a concept or object, or coherently move the narrative forward. In the following, Alex describes how the interplay of constraints has an impact on the doing of an animation for data visualizations:

The visualizer helps guide it and shape it to best tell a story that is accessible to the viewer, but the data drives everything we do. In fact, sometimes the data may not support our story. Maybe the research finding is apparent under certain forms of analysis that are not easily visualized, in which case a chart or graph may be better suited for the visualization, than direct visualization of the data. A good example of this is the change in Arctic Sea Ice minimums when compared from year to year. When seen annually, the Arctic sea ice minimum change shape significantly from year to year. If all that is shown is each year's result chronologically, it is difficult for the viewer to discern the shrinking change in area. If played very fast, one might be able to see it, but it is not immediately intuitive, and might take several iterations to fully see. However, if a graph of the area can be overlaid while the sea ice data plays underneath at a reasonable speed, the downward sloping trend in the graph emphasizes the decrease in area over the time period. This is an example of how the data drives the animation to finally arrive at an intuitive story [Alex Kekesi].

Summary

Even though a number of users describe it as a tool, they also say that working with software involves not just using the software but working with and against its automated processes. Being occupied in creating an animation requires iteration after iteration, tweak after tweak, until the movement is as good as it can be in given time and budget constraints.

“The visualizer helps guide it and shape it to best tell a story that is accessible to the viewer, but the data drives everything we do.”

negotiations

negotiations

Talking about timing and pose and the ‘doing of animation’ reveals that people working with Maya are often negotiating with the software. By negotiating I mean that modellers, animators and riggers are working with and against the settings of the software. Moving away from the more familiar terrain of space and time, I asked Maya users if they experienced a sense of ‘creative tension’ when working with Maya. In using the word tension I meant a mismatch between the capabilities of the software and what the user sought to achieve. When naming one of the best things about Maya, many people stated that Maya makes it possible to create anything you can imagine (See, for instance, Section 1 and the discussion on Building Space). This suggests that users do not negotiate but put the software to use. When paying closer attention to the detailed descriptions of how people work with Maya, it becomes clear that users are at times battling and tussling with the software in order to get what they want out of it. The skill of working with software lies not only in being able to time and pose, but to find solutions and work-arounds for any barriers or problems thrown up in the process.

Overview

- we need to remove the computer from computer generated animation
- you learn to anticipate what the software is going to do ‘wrong’
- one of the best things about Maya is its openness and scripting

Details

“You are forced to do certain things a certain way.”

The narrative that emerges in relation to my question about ‘creative tensions’ is mostly a positive one, with modellers, animators and riggers talking about how learning the software also involves getting to grips with ‘how it thinks’. The different solutions for negotiating with the software mentioned below demonstrate how different toolsets create different sets of tensions for the users to manage.

...it's always those things that are too perfect that will unhinge everything. It is a tension in the software that people have tried to jump, people have tried to fill, get rid of the uncanniness of 3-D visuals, I would say that's the thing I'm constantly trying to do [Barry Sheridan].

You push it too far and then it is pops. You push it too little and then it is squishy. And you are finding this with any animation, it is a constant battle when using software that does things for you [Liz Skaggs].

All the companies develop a plug-in that you can use. Or they [Autodesk] will have developed it themselves and released it in the next edition [Marty Gutteridge].

The best thing about Maya, is everybody's using it. So if you have any problems you can find the solution to the problem within seconds on Youtube just by typing in a few terms [Paul Hilton].

Getting to grips with how software works

An obvious example of getting to grips with how software works is when its functionality requires users to do things in particular ways. Some animators have the sense that they are forced to do things a certain way, which for some is problematic because it takes away a degree of creative freedom. Pushing this idea further, but in terms of it being a positive, or something with which to negotiate, Liz Skaggs comments:

So I think the same thing with Maya, this software presents problems and people find creative ways to hide those problems, make this problem into an asset, you know. If you look at schools like Gobelins you know Pixar has created a very specific style for 3-D animation that works well with 3-D animation, that overcomes a lot of problems, but it is also challenging from a creative perspective because it is something new that people have never seen before [Liz Skaggs].

“It depends on how deep you want go down the rabbit hole.”

The comments made by Liz draw attention to a user having to find ways of handling how software influences the practices of modelling or animating. Learning to use software not only involves developing the skill to creatively use the toolsets, but to also get a feel for the software, to be able to anticipate its default approach to a task and intervene:

In Maya definitely there are lots of defaults. But default does not look quite as good as what you can ultimately do. You can change all the default settings and make it better. But, I mean, eventually you just kind of get used to it. You know where the defaults are, and if you want to get there you will change those defaults. It just becomes routine work to change defaults, at least for me. A lot of people, they don't change it and it will just look like a very generic, everyone look. I've seen painters, physical painters, they actually mix their own paint, you know. They are actually chemists. Wow, so they can make special paint for their canvas. This is the same thing. It depends on how deep you want go down the rabbit hole [Jacky Ke Jiang].

The three examples show how working with software well requires not only the skill of being able to build, move or rig digital entities, but also knowing how and why the toolsets work in the ways that they do.

Ways of negotiating

Software, in the end, does what it's told to do, and nothing else. Doing what it's told, as opposed to what an animator or modeller imagined in their head, leads one modeller to comment of Maya: **"It is almost like a non-compliant colleague. You've got a vision in your head, then you go "for god's sake, not like that!"** It is what I was saying earlier, I seem to do a lot of work getting away from the perfect. Most of the effort I would say in CGI or animation, it's always taking away the computerness, it is always removing the CG from the CG [Barry Sheridan]." Animator's too find that sometimes the interpolations of in-betweening generate unexpected glitches in their timing:

There are timing things too, like the tangents and the way the curves work. Sometimes you just have to phutz with it, you start with maybe all your keys linear, flat or stepped or whatever. They are all the same tangent, and then when you spline it you realise that there are these mini accelerations where there shouldn't have been, and you just have to change the tangents to actually match the motion you intended [XZ].

"You've got a vision in your head, then you know you go "for god's sake, not like that!"

These first two examples show how users work with or around the protocols of the software, or manage their tangents in ways that they hadn't expected. This sense of negotiation comes across in character rigging too, as Liz Skaggs comments: **"You find yourself in this constant battle. You want this character to really have the inertia, but everything is kind of muddy."**

At other times, the reasons for a Maya users' tension with the software are to do with a mismatch between the conventional limits of a toolset and the look required. For instance, Liz gives a description of her skeleton rig breaking because of the mismatch between the pushed pose defining the physicality of a moment of action and the limits of movements within the design of a rig:

I worked on Star Wars [animation] and this is cinematic animation, so there is of course a lot of running, a lot of fighting, a lot of fast action. You know when characters are moving slower in Maya, emoting and speaking with some minor gestures with hands, this is very simple. But when a character is like, phew, phew phew, jumping from wall to ceiling to God knows where, that was really difficult. My rig was breaking all the time. You want to create a pose so that the arm is further than it should be but you of course can't break the skeleton. So this is like telling Peter Paul Reubens that he has to stick to anatomy that is real. And you can imagine how frustrating [Liz Skaggs].

These comments on rigging draw attention to anatomical accuracy and the scope for interpretation and artistic freedom. In the discussions of timing and poses, users commented on their ability to push poses and exaggerate a figure's dimension when it did not need to exactly match human physicality [Section 2 and discussion on timing and poses]. By contrast to projects with more in-built leeway for interpretation, data visualizations aim for greater accuracy. As Alex Kekesi puts it: **“Afterall, our visualizations tell a story about scientific results. Since the scientific result is the core of our story, we are bound to tell that story as accurately and true to the data as possible.”** Even so, there is some scope for interpretation in visualizations:

“So this is like telling Peter Paul Reubens that he has to stick to anatomy that is real. And you can imagine how frustrating.”

Color choices, camera moves, and how data might be represented are all creative decisions made by the visualizer. For example, let's say we're working with a flow field. There are a number of ways that flow fields can be visualized. You could have fixed points with flag-like streamers that flow in the direction of the field, or you can release particles that float along the field, or you could use a morphing-like texture that changes with the field. And in addition to that, you might represent the rate of flow with color, or particle densities, or speed. So, even though there is data driving everything we visualize, there is still a lot of artistic freedom in how that data is to be visualized. That's one of the most exciting parts of our job. It is our job to bridge the gap between raw scientific data and visual storytelling [Alex Kekesi].

XX introduces the interesting problem of how to convey uncertainty in visualizations. Timing and pose are based on suggesting certainty in the sense of giving physical and emotional meaning within the structure of a narrative. But when aiming to visualize a concept for which data is incomplete, depicting uncertainty can be a problem as it has the potential to introduce misunderstandings:

“So, even though there is data driving everything we visualize, there is still a lot of artistic freedom in how that data is to be visualized.”

I think the thing that is the most difficult with scientific visualization is the idea of uncertainty, since usually in a hypothesis there are things that are known, experimentally, and things that are not, things that might be conjecture. And I think it is of interest to the scientists in general to be able to convey uncertainty within the animation, which is a difficult thing. So that is one thing that I think. Usually within an animation I think there is probably more conflict on the part of the scientist conveying an idea that they don't want to convey more information in the animation than they feel comfortable with. I wouldn't say that is a discomfort for me, but it is a discomfort for them [XX].

XX's point, though specifically addressing an issue in scientific visualizations, draws attention to a wider process of negotiation that goes on beyond the software. Animators, modellers, and riggers, unless they are working independently, are part of a team working to a project remit. The creative input of an individual is constrained to working within the parameters of that project.

Open software

Negotiating with software can also involve adding functionality through scripts. Maya is an open software, with two scripting languages, initially only MEL (Maya Embedded Language) and now Python too, available for writing scripts, especially for repetitive tasks, custom modelling, animation, or dynamics. Many people mention that the studios they work in have scripting specialists, or they may create their own scripts: **“I suppose that was one of the good things about Maya as well. You really can, if you're using it professionally, you can develop your own tools [Martyn Gutteridge].”**

When looking for solutions to a problem, Maya users may also look outside of their own studios, connecting with the wider Maya community: **“The best thing about Maya is everybody's using it...the biggest thing I've realised is that it's such a massive community. And it is brilliant because you can just find tools, Mel script, all sorts of different stuff for any kinds of problems you've got [Paul Hilton].”**

“You really can, if you're using it professionally, you can develop your own tools.”

The ability to script helps users with tasks, such as the repetitive and specific movements of molecules in a molecular animation, and also the makes the task of rigging speedier:

Mel Scripts I think are basically good for repetitive tasks, so with a molecular scene there is often dozens of molecules that are moving in some kind of specified way. So with MEL you don't have to do that all by hand, so I don't usually do a lot of hand based key framing, I usually try to do as much scripting as possible to save time [XX].

But with Maya you can script things. I wrote a script to make the rigging process easier. I learnt MEL scripting and I wrote a script and you can use it to automate the rigging process so we can do rigging in 2 hours or 3 hours. Previously we used to take 2 days, 3 days. That kind of scripting is very useful [Navis Binu].

“One of the things our Studio likes the most about Maya is the flexibility it provides via MEL.”

Of Maya's openness, Drew Berry remarks: **“It's a package that is just extremely open and flexible to whatever. It's not pre-canned in its toolset. The tools are not built ready to achieve a certain outcome. It's more you can devise a toolset to suit your particular need.”** The following comments sums up the flexibility that scripting can offer an animation studio with very specific requirements for data visualizations:

One of the things our Studio likes the most about Maya is the flexibility it provides via MEL. MEL gives us the ability adapt Maya for whatever unique challenge we face when it comes to visualizing highly complex NASA scientific data sets. Over the years, we have built a library of MEL scripts that allow us to efficiently ingest NASA-related data ranging from flow fields to ephemerides [Alex Kekesi].

Summary

Using software is a multifaceted process. I've used the word negotiation to draw attention to the ways in which 'using' software involves not just knowing how to apply the principles of animation to the movements of objects in time (though this is essential), but also having a good working knowledge of how the software sets its defaults and orders actions. Being able to negotiate with these features of software are part of the process of modelling, animating and rigging. Finally, using MEL scripting there are ways of getting around many problems, easing the load of repetitive tasks, and adding flexibility to the software.

technically speaking

technically speaking

This final section deals with the most abstract set of ideas since my questions were aimed at finding out more about the technical side of working with software. For people not familiar with animation software, thinking about what animators do is limited to looking at images on the screen. But behind the scenes there is much of interest in what modellers, animators and riggers have to say about the technical side of creatively working with software.

Overview of replies

- people come to know how to work things out in a certain way
- descriptions of ordered thinking reveal how users combine creativity and process
- the deeper structure of the software is apparent (though not of interest to everyone)

Details

The answers brought together in this section are in response to my question of whether there is something called the ‘language of the interface.’ Users of Maya have responded thoughtfully to this rather abstract question. Their comments move from thinking about the visual languages possible using Maya, to articulating their range of engagements with the technical side of animation software.

“So I think that Maya has, but it's the same as 3DS Max, is the same as them all.”

To a certain extent you are in control, but maybe it does dictate the way in which you do things to create an object. And then, obviously, the more time you spend in there, you kind of understand it [Ben Thomas].

I can't even contemplate how much work has gone into writing the code to make something do that. It's just incredible [Paul Hilton].

You can actually bring up a window that shows you how everything is connected. So you have to get your head round that concept and you have to get your head around the concept of numbers, numbers for everything [Barry Sheridan].

And that will become second nature, you know not to put certain things next to others in the hierarchy. But to start with it's hard to get your head round, it is really not clear what is causing the problem [Ady Holt].

Maya and visual language

Asked whether he thought Maya had a language, KmcD replied: **“It can become like a massive super language, where you can do anything you like. So I think that Maya has, but it's the same as 3DS Max, it's the same as them all.”** Many users of Maya echo KmcD in saying that the software allows them to do anything they can imagine, from photorealistic images intricately matched to live-action, to the more exaggerated poses of cartoon conventions. Taken in this way, using Maya is flexible and without limits.

Scratching at the surface reveals a slightly different view and expands on KmcD's further remark that **“everyone knows how to work something in a certain way.”** In using software people work in particular ways. As a consequence creative solutions take the ways of doing things in Maya into account. For most people this is not actually seen as either a limit or problem. One person who does use the word problem, in fact turns the problem into a positive:

I think it forces people to develop a new...just like Flash has...it became very popular because it is very fast and cheap. It forced the industry to go into an animation style that is similar to 60-70s UPA, a very limited cartoon style that they can produce very quickly in this software. And because this tool came, you know, this is similar to when paints came in tubes and now everybody paints landscapes. So I think the same thing with Maya, this software presents problems and people find creative ways to hide those problems. So to make this problem into an asset, you know [Liz Skaggs].

Another Maya user talks about becoming used to a way of thinking: **“It is like I evolved my brain muscle into the software, to work in some kind of harmony. It is a limitation but you turn the limitation into a positive** [Jacky Ke Jiang].” Both Liz and Jacky acknowledge that people find creative ways to work with the problems or limits presented by software. Good modellers, animators and riggers find ways of getting the software to do what they want.

The idea of an evolving ‘brain muscle’ that accommodates to the software underlies the following two remarks as well. Hunter Grant talks about the ways in which different software packages lead to different thought processes:

The funny thing is that, depending on the package that you use, whether it is Maya or 3D Studio Max, some of them create different thought processes. Whereas something like Character Studio back in the old 3D Studio Max days, you had to think a lot more like stop motion because you didn't have some of the tools that you have in Maya [Hunter Grant].

“The funny thing is that, depending on the package that you use, whether it is Maya or 3D Studio Max, some of them create different thought processes.”

When users get very familiar with the software they are working with, the thought processes become transparent or automatic. The different thought processes only come back into the foreground when moving between packages: **"Because you get used to one software, you practice 10 years, your muscle memory becomes permanent. It is hard to adjust to a completely different kind of flow [Jacky Ke Jiang]."** Jacky expands on being used to one software as opposed to another in comparing his experience of using Z-Brush and Maya:

"Your muscle memory becomes permanent. It is hard to adjust to a completely different kind of flow."

I started working at the Disney studio and I got to learn to Z-Brush from my talented colleagues and they showed me all the tricks. Even they say Z-Brush has the weirdest navigation system. But the result is great, Z-brush, it is a powerful software. Every time, when I use Z-Brush, I feel I am going to a Mafia meeting! You can get something you want from them, but you have to follow their own customs. It is like they have a lot of...you have to press a button, and then start to draw, and then release that button. It is just weird. Maybe for a couple of people it's very intuitive, but for me it is always...it just doesn't feel quite right. Maybe I'm from the more traditional Maya. Because, I've heard the story of people who was born with Z-Brush, they can't get used to Maya [Jacky Ke Jiang].

On a similar point, but in relation to moving between Maya and SoftImage, James Farrington talks about only needing a few days to shift his working habits between one software and another:

You kind of have to repeat doing things and repeat doing things for a long time. I found this also swapping software. There was a job where I actually had to use SoftImage again, after years of not using it. And to go back into SoftImage, it was not a problem at all in terms of the way I thought about how I want to do the work, they both do more or less the same sort of thing. But it was just down to keys, buttons and where to find things. And I thought it was going to be a nightmare, but it only took me about three days to break the habits of using Maya, and form new habits of using SoftImage. You'll be using a mouse and keyboard and you won't be aware of using it, probably like learning to touch type, those sorts of things work on a different level and your conscious mind is thinking about other things entirely different to using those controls [James Farrington].

“Probably like learning to touch type, those sorts of things work on a different level and your conscious mind is thinking about other things entirely different to using those controls.”

Using a different metaphor to memory muscle or habits, independent animator David O'Reilly talks about how software can feel embedded in a user's mind:

It is possible for Maya or any 3-D software to feel embedded in the user's mind, so that any conceivable three-dimensional phenomenon can be broken down into a chain of simple actions which eventually reconstruct it digitally. This is perhaps the most interesting side effect of becoming familiar with the software [David O'Reilly].

‘Thinking the Maya Way’:

The words memory ‘muscle,’ ‘habits,’ ‘embedded,’ all suggest that the software requires that a problem be thought through in particular ways. (This is not the same as saying that users are limited in having to think in those ways: the better users learn and exploit the particular ways of thinking and effectively combine

creativity and process). For people who do not have much insight into working with software, or are new to software, descriptions of ordered thinking reveal more about how users combine creativity and process.

“I guess it's to do with the algorithms behind it.”

Just as becoming familiar with how software works in terms of default settings is part of the expertise of using software, so is knowing how to order actions: **“Modifiers happen in a certain order, and, if you start messing around with that order, things that rely on other things happening earlier or later, may not happen [Barry Sheridan].”** Ben Thomas talks too about how working in the right order has consequences not just for one toolset, but right down the pipeline, from modeling to rigging, and further on:

I think in terms of the language, I suppose it does in a way, in the sense that the way you go through a process, the way it guides you to do something. To a certain extent you are in control, but maybe it does dictate the way in which you do things to a certain object. And then, obviously, the more time you spend in there, you kind of understand. If you're modelling that language is mirrored in rigging, in the way that it needs to have something done to make it to work. I guess it's to do with the algorithms behind it, so the order in which you need to constrain an object to another object, and what you need to select first to do something [Ben Thomas].

Barry Sheridan describes the particular necessity of paying attention to the ordering of transformation, scale and rotation when creating movement. Not getting the order right means can generate nonsense:

So you've got these three pieces of information, you've got the information controlling its up and downness, information controlling its scale, and information controlling its rotation. And you can order these in different ways. You can tell Maya to first look at movement and then look at rotation and then look at scale. And there's only one way of arranging these three factors to get the effect that you were expecting, as anything else would just give nonsense. So when you are animating you have to think in components [Barry Sheridan].

As Maya users expand further on the experience of ordering, their comments also begin to reveal a very different way of thinking about the digital entities on the screen. For viewers used to seeing things move about on screen, depictions of objects in the viewport are the most familiar feature of the user interface. As users talk about digital entities in terms of components, the correct ordering of actions and the connections between parts, the wholeness of a 3-D entity can be seen as a human view that sits alongside what might be called the computational view of objects as packets of data. Martyn Gutteridge explains his understanding of this aspect of Maya:

Maya breaks things into two sets. You've got the solid complete object, so you can view it in that mode. Or you can view it in component mode where you're seeing all the components of it and stuff. And you can go between those two, but I suppose that's like any 3-D software. It's got a lot of... It's quite a node-based system, so all the shader networks and things, and you can do quite a lot with those nodes. You can use utilities and stuff [Martyn Gutteridge].

“And there's only one way of arranging these three factors to get the effect that you were expecting, as anything else just give nonsense. So when you are animating you have to think in components.”

Ben Thomas too draws attention to Maya as a nodal-system in his comments on the difference between seeing

the object as a virtual projection in the viewport (where the user sees the 3-D object in the front of them) versus the more graphical depictions of data as names, numbers and connections:

Well, I do see it as a node based system, but I choose not to use it in that way. I like to see it with that interface. You can do things in hypergraph, where you are viewing lots of texts and names and numbers, but coming from the background that I do, I prefer to see it in front of me [in the viewport]. And I think it is a nice interface, and it's fairly easy to navigate around. From spending more and more time in it, you know where things are going to be, and where they are likely to be hidden. And solve problems [Ben Thomas].

“Well, I do see it as a node based system, but I choose not to use it in that way.”

In a node-based system, the modular components that constitute an object can be connected to form a graph. Not only does this offer a different way of depicting space, but provides another avenue for modellers, animators and riggers to manipulate the details of their project. Ady Holt says:

You can see it if you open up enough windows. You've got your viewport with your project in it, you've got the outliner, and that has a view of all of the things you've got, but it doesn't show all the connections. You need to go into the hypergraph if you really need to see what is happening. And you've got all the nodes and everything, and you think I'll just see what's connected to what, and you open it up in hypergraph, and there's like 100 things with all the lines and it's just like...[Ady Holt].

Ady's comments convey how complex node-based depictions can be. Rather than moving or rigging an

object, a user is faced with an array of connections, and the more things there are in a scene the more connections there will be. Barry Sheridan describes how these connections build up:

It means that the things, objects or animations in the computer's world, they are almost stored as compartments of information. So for example, just a simple cube consists of data saying that it's a cube, connecting to data that says what size it is, connecting to a set of data that says what material it will have on it, and that's the obvious stuff that goes on to make up the cube. But the cube itself is connected to a set of data that says what lights affect it, a set of data that says what happens to it when it gets rendered, does it cast shadows, does it receive the shadows, and so on. And so that bit of data which says what lights affect it, that piece of data is also connected to everything else in your scene, so quickly you see that all these links build up. And when you go down an animation pipeline it is even worse, because you've got connections to the rig and the rig is connected to do everything [Barry Sheridan].

“So those are all graphically depicted and it's like a network, so you have your different icons and they are connected with inputs showing that this is going to this. It reminds me of, mostly of electronics, logic gates and things.”

Martyn Gutteridge takes a slightly different direction when thinking about this same point, drawing out a visual metaphor to describe what these nodal connections look like on the user interface:

So those are all graphically depicted and it's like a network, so you have your different icons and they are connected with inputs showing that this is going to this. It reminds me of, mostly of, electronics, logic gates and things. All the different tasks that you want to achieve and, they all have their own utility, they all have their own toolbox that you can go into yourself. And that will effect what you're seeing on the screen. Pallettes and things. I suppose it's quite like any software. You've got toolsets [Martyn Gutteridge].

Talking about data and compartments gives insight into software and its language. The explanations of ordering

and connected compartments of data reveal two things. Working with software involves engaging with and understanding protocols in so far as being able to make sense of a necessary order for actions. These explanations also illustrate how software ‘sees’ shape. Not in the same 3-D way as viewers, but as compartments of data to be computed. From this perspective, the combined necessity for both creative input and also understanding enough about the software protocols is essential.

Maya and Code

“So you know, we take for granted, well I take it for granted, when I push a button that it will do something completely fantastic.”

A more literal way of answering whether or not Maya has a language is to think in terms of code. The software is coded using the object-oriented language C++. The program is open or extensible using either MEL or Python scripting as discussed in Section 4. But code, unless you programme or script, remains opaque. For many people, there is an assumption that the code will work as expected. As Paul Hilton puts it: **“So you know, we take for granted, well I take it for granted, when I push a button that it will do something completely fantastic. I know what it’s going to do, and I know how to make it do it, but the background of it is just incredible.”**

As Paul goes onto to describe, when thinking about whether Maya has language, there is a way of ‘seeing’ this code via the script editor. Each time the software carries out any computations, the details are given in the script editor:

Yes. But it is a language I don't understand. Well, okay, a small amount. I think it's the architecture, you know, what's actually running everything. If you open up the script editor and you create a box, it tells you what it's come through to get you there. "Have you any idea how hard I've worked to create this box for you..."! You've got it there in a list, and that is kind of fascinating to me, but it is something that I really don't understand at all. So I think that's the language it uses, so yes I think it does. And I tend to skim across the surface of that language, which I probably shouldn't. I should probably dig a bit deeper but it's a whole other world [Paul Hilton].

**"But it is a language
I don't understand.
Well, okay, a small
amount."**

The phrase 'under the hood' is used quite frequently to describe what goes on in this whole other world of the software. Phrases such as nodes and architecture begin to suggest a less opaque structure. When Maya was first released in 1998, interface designers George Fitzmaurice and Bill Buxton working for Alias|Wavefront published a research paper outlining the thinking behind the drop-down boxes on the UI. It is worth briefly noting what Fitzmaurice and Buxton say. They comment that while artists may not really want to know about the deep structure of the program: 'providing access and acquiring such understanding is often necessary for users to achieve their goals. Thus, we must find ways of exposing the deep structure to the user in ways that are compatible, intuitive and efficient.'¹ They go onto argue that deep structures can be exposed via dialog boxes or manipulators, and that these reveal the relationships among the components of a model or scene. The answers that have been discussed above reveal exactly this process. Maya users explain their engagement with the processes of the software in ways that show their awareness of its deeper structures.

¹ George Fitzmaurice and Bill Buxton (1998) 'Compatibility and Interaction Style in Computer Graphics.' *Computer Graphics* 32(4): 64-69: 64.

Summary

Asking people whether or not Maya has a language is an abstract thing to do. Even so, the answers have revealed very tangible examples of how users explain their thinking about and engagement with the software. In their different ways, descriptions of ordered actions using toolsets, nodes, and the conceptualization of digital entities as connected compartments of data, expose the extent to which the work of modelling, animating and rigging involves knowing how to work with the technical processes of the software, as well as having the creative skills to make animations. To be sure, not everyone expresses their engagement with software in the ways described in this section, but those that do give insight into how they understand the software to work, and that animation is an effective combination of technical awareness and creative input. Martyn Gutteridge's remarks on dialog boxes makes the point succinctly:

It is like a schematic. It's a logical view of things. So is that very technical? Yes it is. Maybe that's contrary to how an artist uses things. Possibly yes, I don't know [Martyn Gutteridge].

final words

Though the word frustration looms large, flexible and versatile also feature strongly. One of the features of Maya is its versatility. As Drew Berry puts it:

I think one of the strengths of Maya is that there are so many different ways of actually working with it. If you are a programmer you can type in code, if you are a person who likes to have icons to click on, it has that. If you're a person who wants to type in numbers it has that. It also has the marking menu system which I used to use. It was one of the main glories of Maya, where you can use a pen and swipe, and have commands that respond to your characters. That's still my favourite way of interacting. It just suits my way of thinking versus having a menu system where it drops down from the top and you go left, down, down, down. Being able to radiate outwards, and then build commands and scripts into...To me it's just a beautiful...and it's under exploited [Drew Berry].

“I think one of the strengths of Maya is that it has so many different ways of actually working with it.”

Since there are a number of ways in, talking about Maya yields a wide range of different experiences of the software. Even though many perspectives about space, timing, poses, performance can be read in the answers collected together in this document, a central thread emerging is that being able to create animations requires knowing how to make something, a character or entity, and then give that something movement in ways that are engaging, regardless of the technique used. Software is a complex device that mediates those skills, as opposed to giving those skills to someone.

One of the features of using 3-D software is that users say it allows them to do anything they can imagine. But at the same time, software only does what the people using it tell it to do. As the answers describe, working with software requires combining animation skills with a good operational knowledge of how the processes of software work. In experiencing this process, people move between remarks about software becoming transparent when they are in the zone and having to think about working with numbers:

You're in the zone. That's an awesome place to be, and that can happen when you're creating something in 3-D. Like anything that requires intense focus, if you're inspired, you can become completely engulfed. The advancement of technology does play a role in this, in a sense that as interface becomes more transparent, focusing on the craft becomes easier [Aaron Chan].

“It's just like playing an intense sport, or anything like that, you are just completely aware in that moment.”

You have to get your head around the concept of numbers, numbers for everything. I guess it's like that blob of putty, or modelling clay. If you'd been given modelling clay you would model with it without having to know its density mass size, location in space. But in Maya it's numbers. I guess that means you must think that way. See, that's when you start losing the freedom of it all, because you're defining stuff by numbers. If I make a cube, and it is asking me for a size I'm going to type in whole values, using say 5 x 5 x 5. It is never going to be 4.9 or 5.2 otherwise that is going to lead to headaches later [Barry Sheridan].

The seemingly polar opposites of immersive freedom and the constraint of precise numbers capture the range of experiences of using software: a joining together of art and technology.

“That is the thing about 3-D. It is the art side and the technical side joining up [Martyn Gutteridge].”

