# 3d Cel Shaded Rendering and 2d Integration

An innovations report by Dean Wright

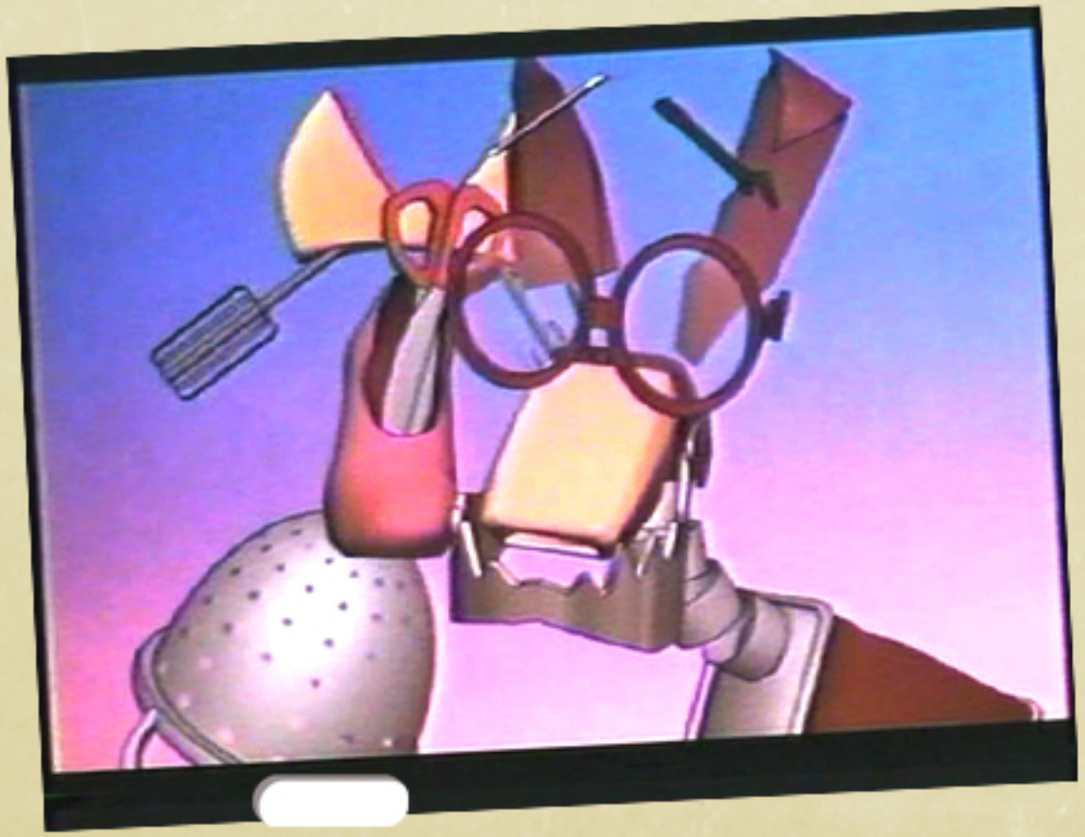
# Abstract

This project and report is a guide to creating toon shaders for 3D animation and how to integrate traditional hand drawn animation with CG rendered images. It will tackle not only the visual but some technical aspects of bringing the mediums together. This is done using research into previous techniques used in the film industry to produce similar results and through first hand practice. Information has been attained through research papers, websites and people in the CG industry. The resultant shaders provide 3D renders that appear to be potentially hand drawn and sit well alongside 2D characters and environments.

# Contents

# Introduction...1

Presented to you, is a report into the techniques of attempting to integrate 3D rendered images alongside traditionally hand drawn 2D animation. The primary drive for this research is to incorporate it into the major project short film 'Sky-caper'. In addition to this, as 3D has established a strong foothold in animation, 2D has been disregarded by some as a dated medium. However, both mediums have their merits and downfalls, so by bringing the two together, you are able to take advantage of what they both have to offer. Although there have been remarkable developments in improving the flexibiltiy of 3D animation, there are still limits to the way in which objects can be deformed to create smooth flowing movements. Unlike 2D where its a simple matter of moving your line.



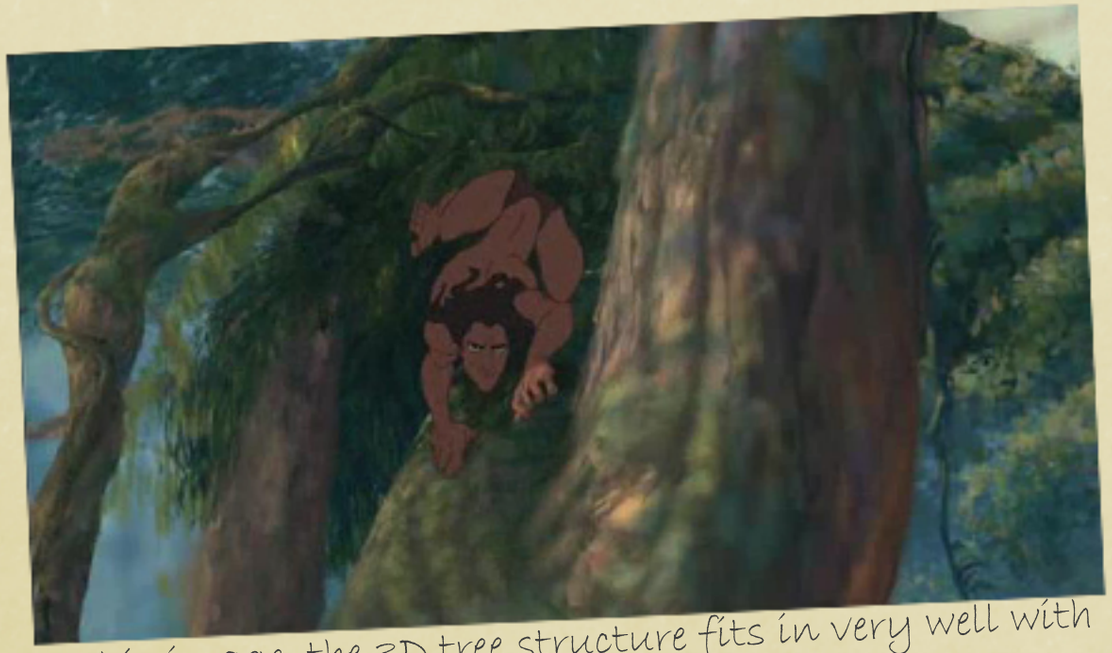
A still from Who Framed Roger Rabbit. Notice the excellent integration of the 2D character, aided by very detailed colour work to similate the real life lighting.


On the other side of the coin, this downfall is also 3D's greatest strength. The ability to reproduce a perfect rigid surface, or produce similar artefacts on a huge scale is an extremely difficult and laborious task for hand drawn animators. But is much simpler to produce using the processing power available of a machine. Integrating two mediums isn't a new idea of course. 2D has been alongside live action on many an occasion, possibly the most famous of which is 'Who Framed Rogger Rabbit' (above). One of the earliest attempts of producing apparently 2D images using 3D techniques is 'Oilspot and Lipstick' (left). Tad Gielow and MJ Turner developed the shaders with Walt Disney Pictures and Pixar back in 1987. The lines and colour were produced separately and composited together, they also created a plotter that could draw the lines produced by the computer allowing the images to be integrated into the traditional Ink and Paint process.

# Introduction...2

This ink and plotter technique was used on a number of Disney films such as 'The great Mouse Detective' and 'The Little Mermaid'. However, 3D integration into 2D made a huge leap forward in the animated film 'The Lion King'. The famous scene with the stampeding Wildebeests may not have been possible were it not for the skill of the 3D artists. A number of the close-up Wildebeests would have been traditionally animated, but the sheer numbers required for the stampede meant they had to be reproduced in 3D. As you can see in the image to the right, the huge number of toon shaded animals integrate very well with their traditional 2D environment. This boosted confidence in the technique leading to its use in a huge number of films such as Hercules and Mulan.

Since then computers have integrated themselves in much more pronounced ways. The two mediums work side by side and have merely led to more creativity from the artists who's imagination is simply given more freedom with the use of 3D. A perfect example is the 'Deep Canvas' technology used in the 1999 Disney movie, 'Tarzan'. In the image to the left, the very complex foliage and tree structure is 3D generated and mapped out, along with the camera movement around the environment. However the 2D artists can paint onto the 3D models, just as they would a normal 2D canvas. This means very complicated environment moves can be planned and run through, then the animator can simply overlay the character to suit the movement. This would simply have not been possible without the aid of 3D tools.



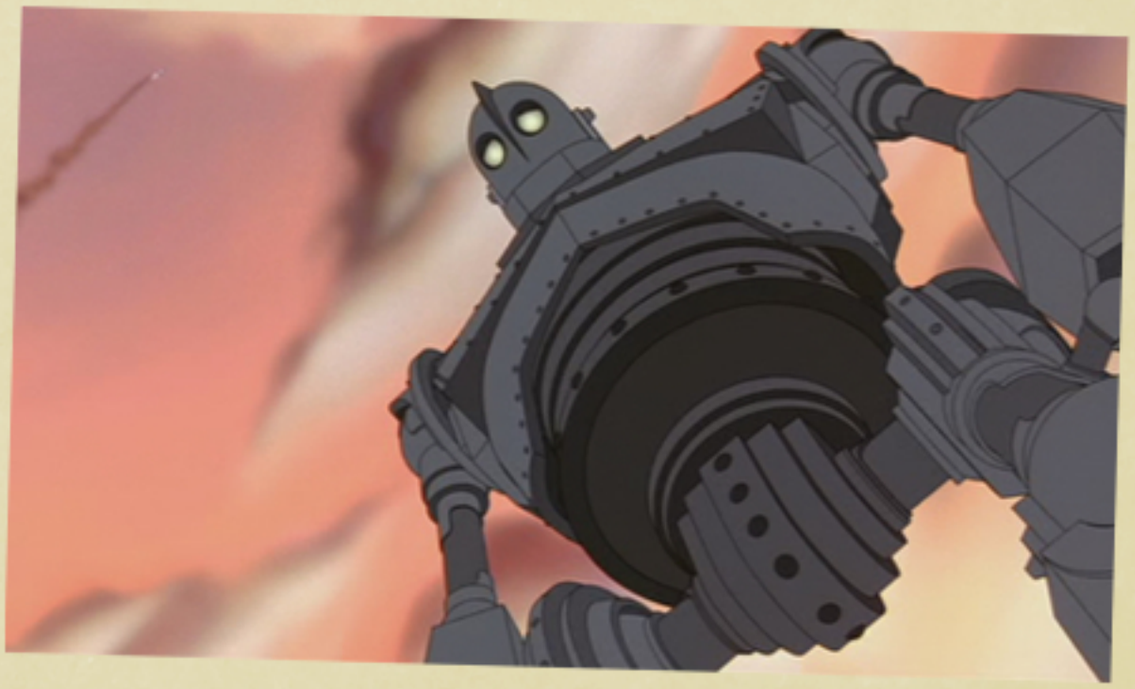
In this image, the 3D tree structure fits in very well with the 2D foliage thanks to being able to paint onto the 3D models.

# Introduction...3


Of course there are many different styles that 2D animation has traditionally been painted in, which results in many different 3D shaders being developed to mimic those styles. The simple, flat colour shading, is what was required of the film 'Sky-caper', an example of which can be seen in the image above from the 1999 Warner Brothers Animation, 'The Iron Giant'. This motion picture is the primary inspiration for the shaders investigated throughout this report for a number of reasons. Using this style, the work load on the 2D animation artist is considerably reduced as there is no tonal work involved and it is also simpler to implement in 3D as no lighting is needed in the scene. This makes the 3D and 2D easier to match and helps reduce any continuity issues. In addition to this, from a purely aesthetic point of view, it is the aim of the film to relish its 2D style and so adding more detail or shade work could merely detract from the animation.

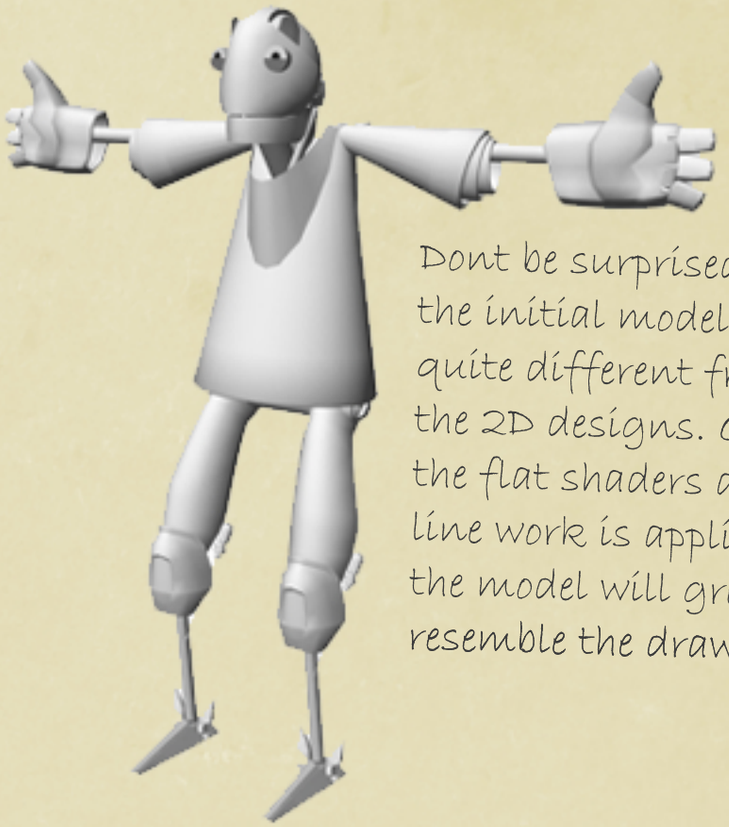
Computers are spreading their influence into many other stages of 2D production, particularly in the storyboarding process (Brad Bird; 1998). Using After Effects, scenes can be set-up and even basic animation directed to lay out and decide camera moves for each shot. "While staging is no substitute for story, I felt then, as I do now, that the camera is an unseen character, the eyes of the audience" (Bird; 1998)

As digital practices work alongside traditional methods more and more, the emphasis of this investigation will be into attaining seamless integration of the two mediums, as well as good quality of line from the 3D renders. Other issues such as reflections, how to deal with cross-medium character interactions and other details will also be covered briefly.

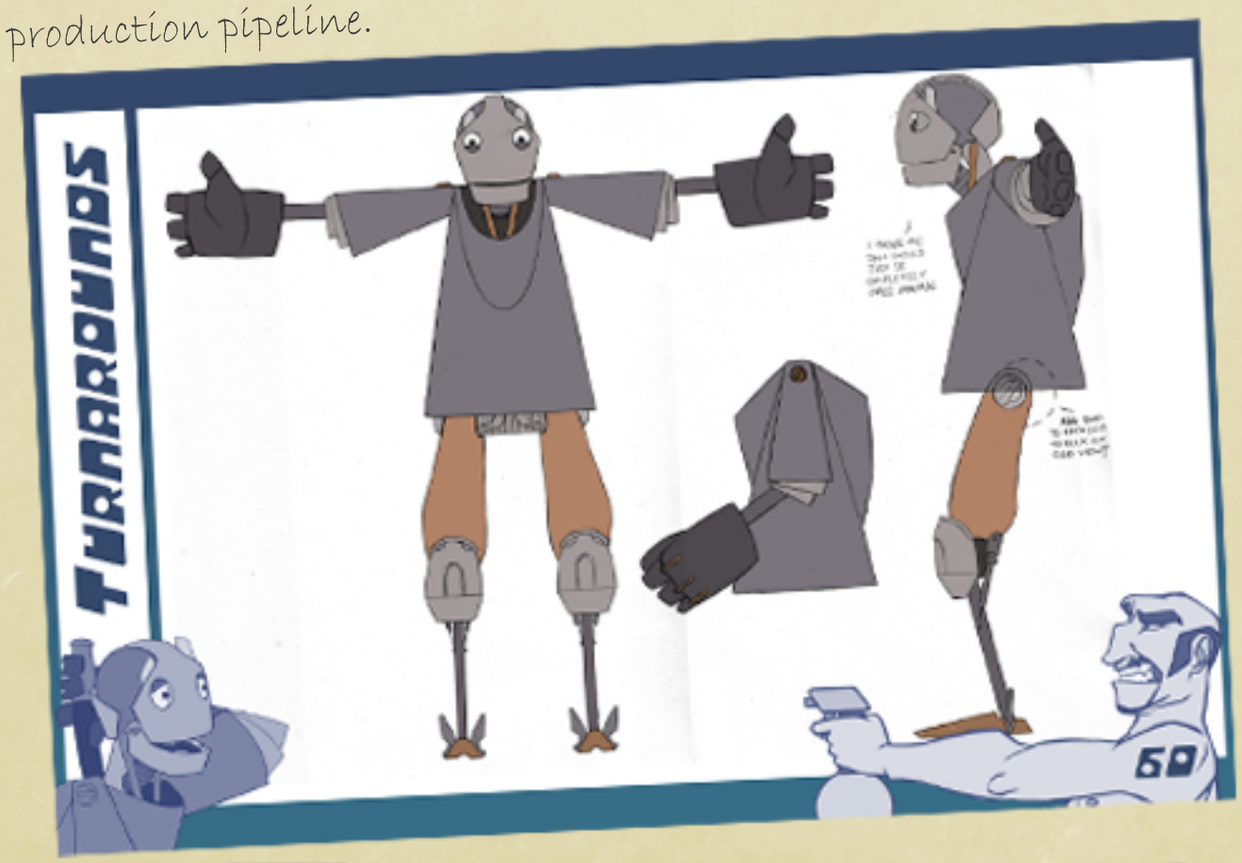


# Toon Shader Development...1

An important lesson learnt while developing these shaders was that by taking the 3D element into account at the design and modelling stage, you can greatly reduce the amount of work needed at the shading stage. This is particularly important when dealing with the lines. Needless to say, having a good design that takes into account how things can move in 3D space is a necessity. Although of course it is important to not let the 3D element get in the way of artistic style, so its about attaining the right balance. From the original 2D designs below, the 3D model to the right was derived. Since that initial render, a few changes were made to help accommodate the shading work. So when embarking on a similar project, it is recommended to carry out some rough tests to see how the lines are working on your model as you build each part of the character, to save any headaches further down the production pipeline.



Dont be surprised if the initial model looks quite different from the 2D designs. Once the flat shaders and line work is applied, the model will greater resemble the drawing.

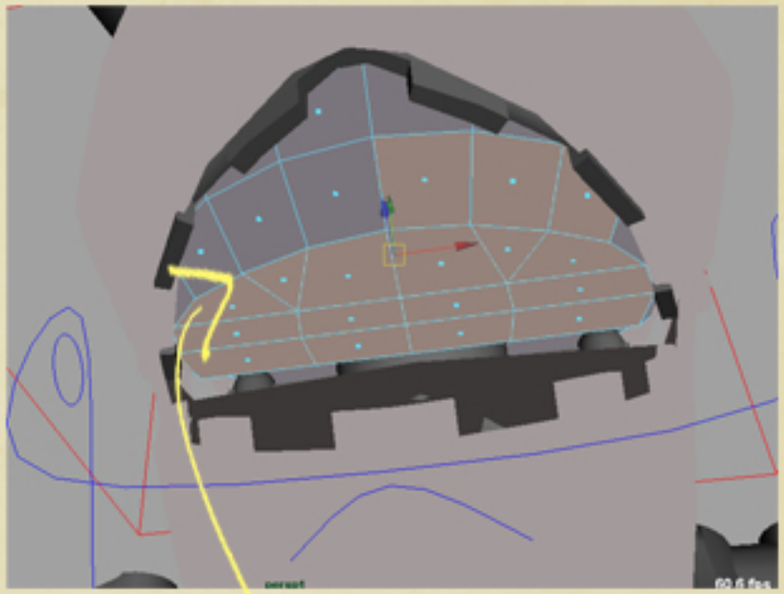



The software used to model and texture the character was Maya and its built in toon shader. It proved very versatile at producing the shade and lines needed for the entire character. Important modelling points worth mentioning when using mayas toon shaders are;

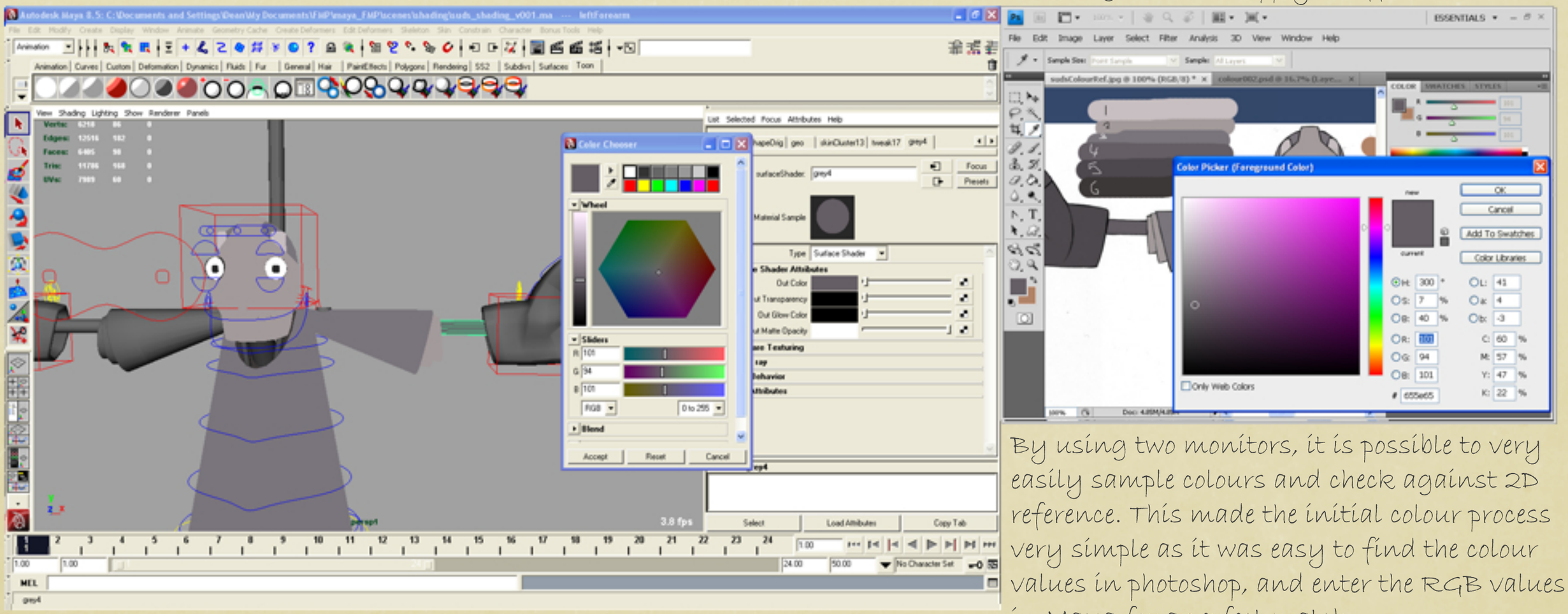
- It is not possible to colour the back-faces of polygons different from the front. So for any poly's that need to be seen on both sides but different shades, a wafer thin extrude must be made so there is another polygon layer on which to apply a new texture.
-The line is drawn very accurately when reaching a border edge, so its worth leaving the edges of collars, sleeves, etc as single polygon borders.
-Dont be scared to add edge loops to soften areas where you do not want a line to be drawn, or remove edge loops to harden edges to encourage a line to be drawn.

# Toon Shader Development...2

The initial flat colour shading process is very simple. As no lighting will be needed and no changes in tone will happen over a single mesh, it is simply a matter of picking the right colours and assigning them to the right areas of the mesh. The major advantage of the entire toon rendering process is it can all be seen in the viewport interactively. There is no need to render each time you make a change to see the results. It is very easy though to lose track of the colours and even lose the shape of the mesh as all you can see are flat surfaces. It is therefore important to name and number your shaders so you can easily relate to them in the 2D drawing. The importance of naming all your shaders clearly is particularly important when it comes to creating the lines, as you soon have dozens to get lost in.

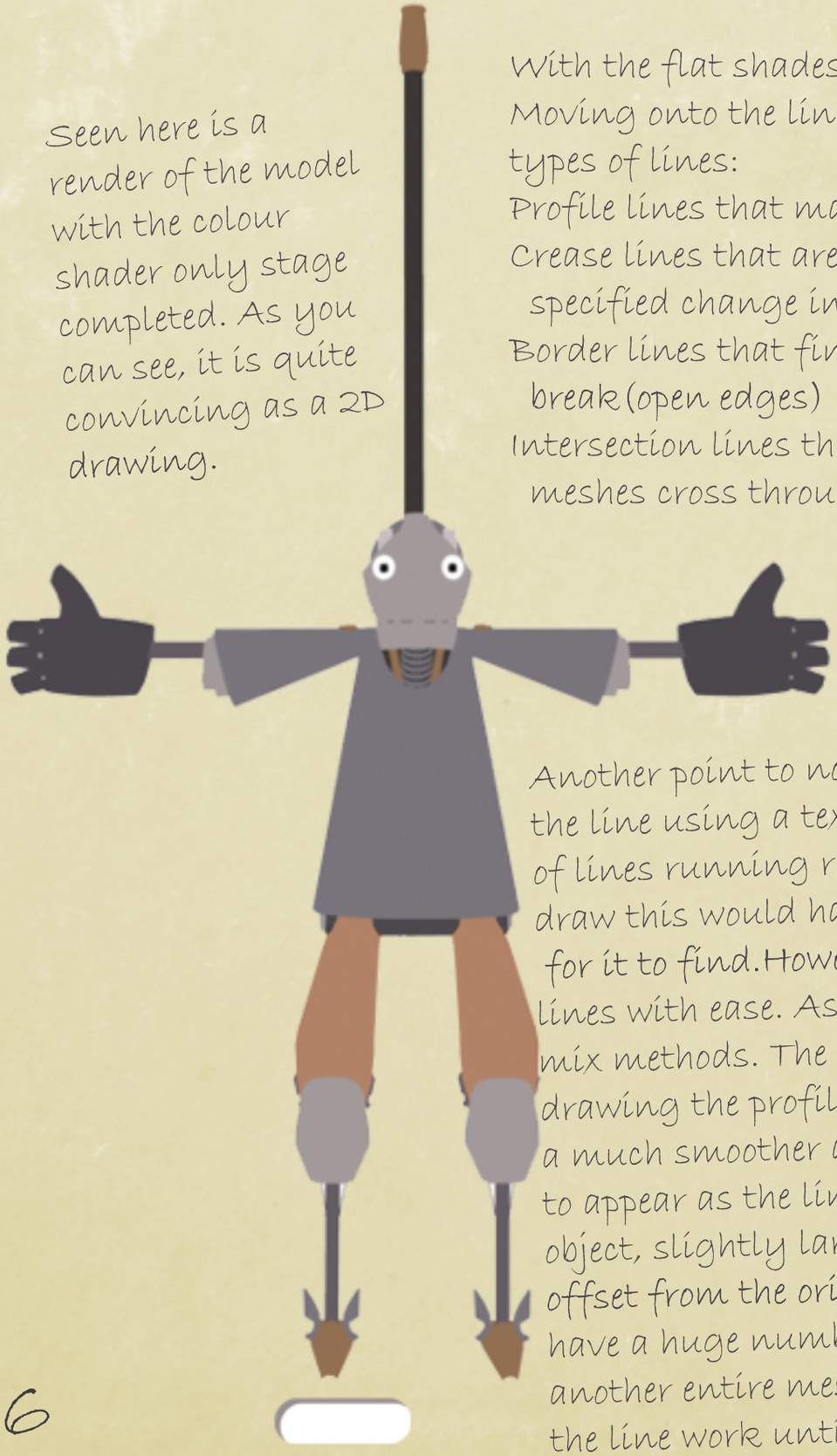


Individual faces can be selected on a single mesh to apply a different colour.




By using two monitors, it is possible to very easily sample colours and check against 2D reference. This made the initial colour process very simple as it was easy to find the colour values in photoshop, and enter the RGB values in Maya for a perfect match.

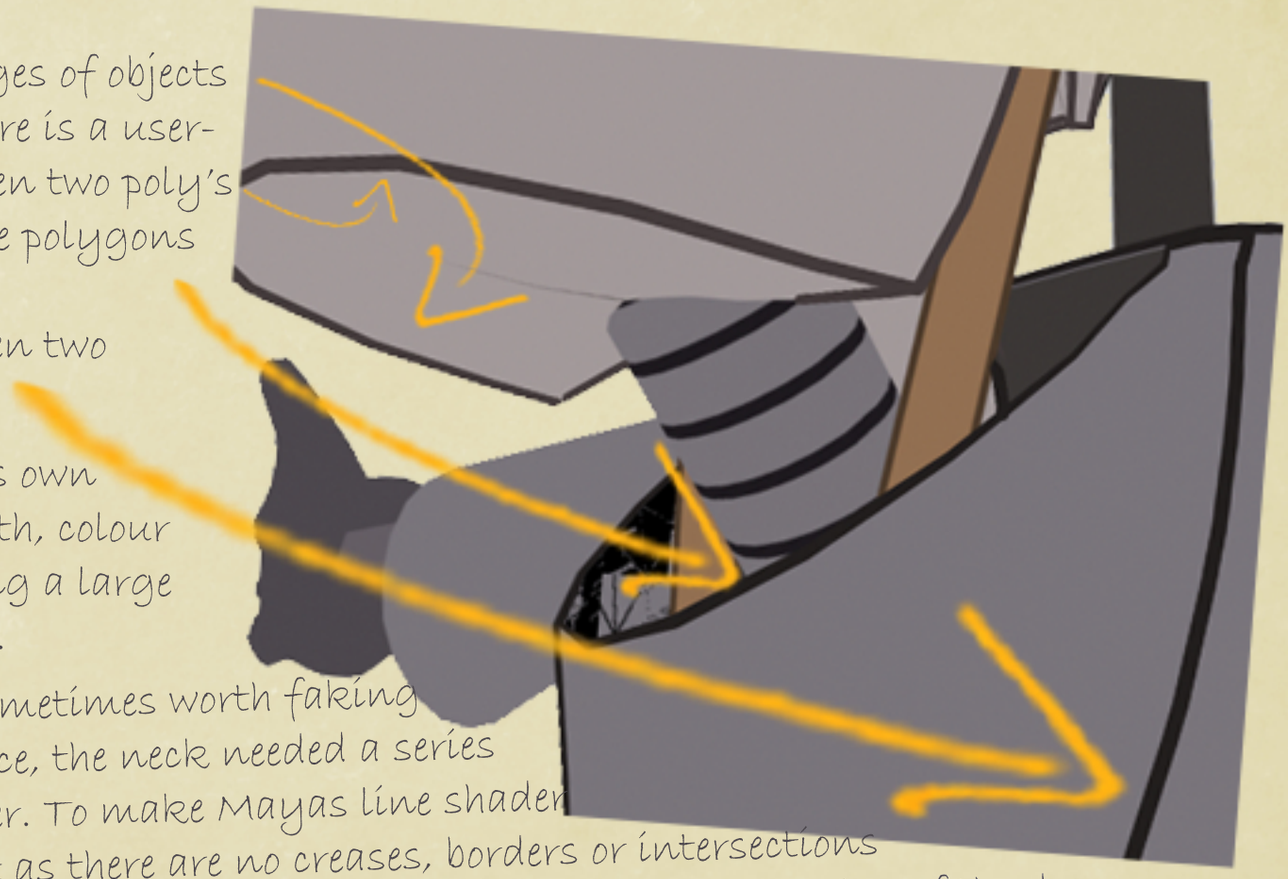
# Toon Shader Development...3



Seen here is a render of the model with the colour shader only stage completed. As you can see, it is quite convincing as a 2D drawing.

With the flat shades all assigned we now have a clear impression of the characters 2D appearance. Moving onto the line shading, there is a substantial more amount of work to be done. There are 4 types of lines:

Profile lines that map around the edges of objects
Crease lines that are drawn when there is a user-specified change in the angle between two poly's
Border lines that find the point where polygons break (open edges)
Intersection lines that are drawn when two meshes cross through each other.

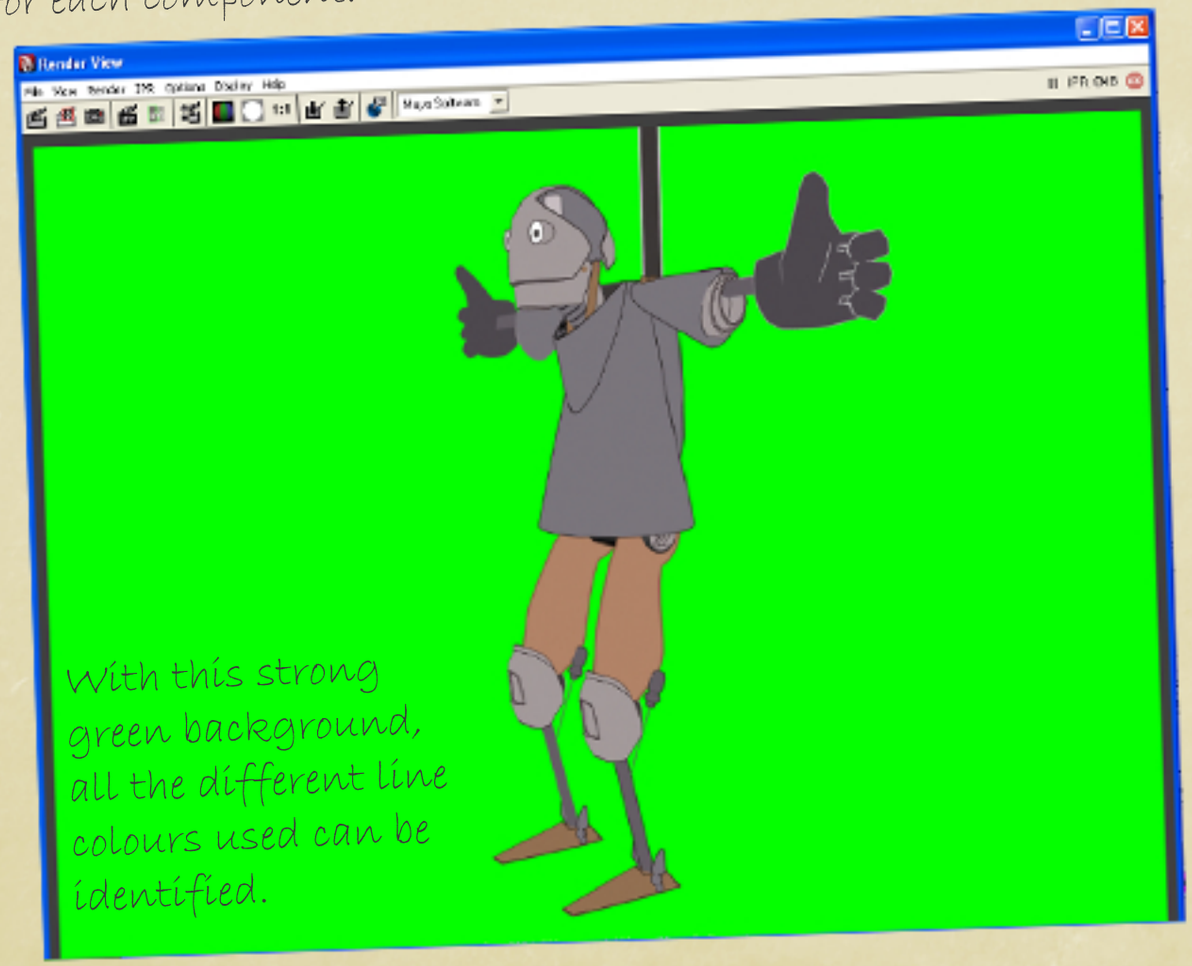


Each line has its own settings for width, colour and so on, giving a large range of control.

Another point to note is that it is sometimes worth faking the line using a texture. For instance, the neck needed a series of lines running round the cylinder. To make Mayas line shader draw this would have been difficult as there are no creases, borders or intersections for it to find.However, using a simple cylindrical projection created the impression of the drawn lines with ease. As long as the colours match, and it looks correct when rendered, then its fine to mix methods. The same method was used on the knees and backpack. There are two options for drawing the profile lines, 'offset mesh' and 'paint effects'. On all occasions the offset mesh provided a much smoother and accurate result. Unlike paint effects which finds the edges and creates a mesh to appear as the lines, the offset mesh method creates a duplicate of the mesh that surrounds the object, slightly larger than the original. Then whenever an edge is detected, the mesh is displayed, offset from the original object. The disadvantage of this method however is that you very quickly have a huge number of polygons in your scene as every time you create a line shader, you create another entire mesh. It is therefore recommended to use render layers when animating, to remove all the line work until render time.

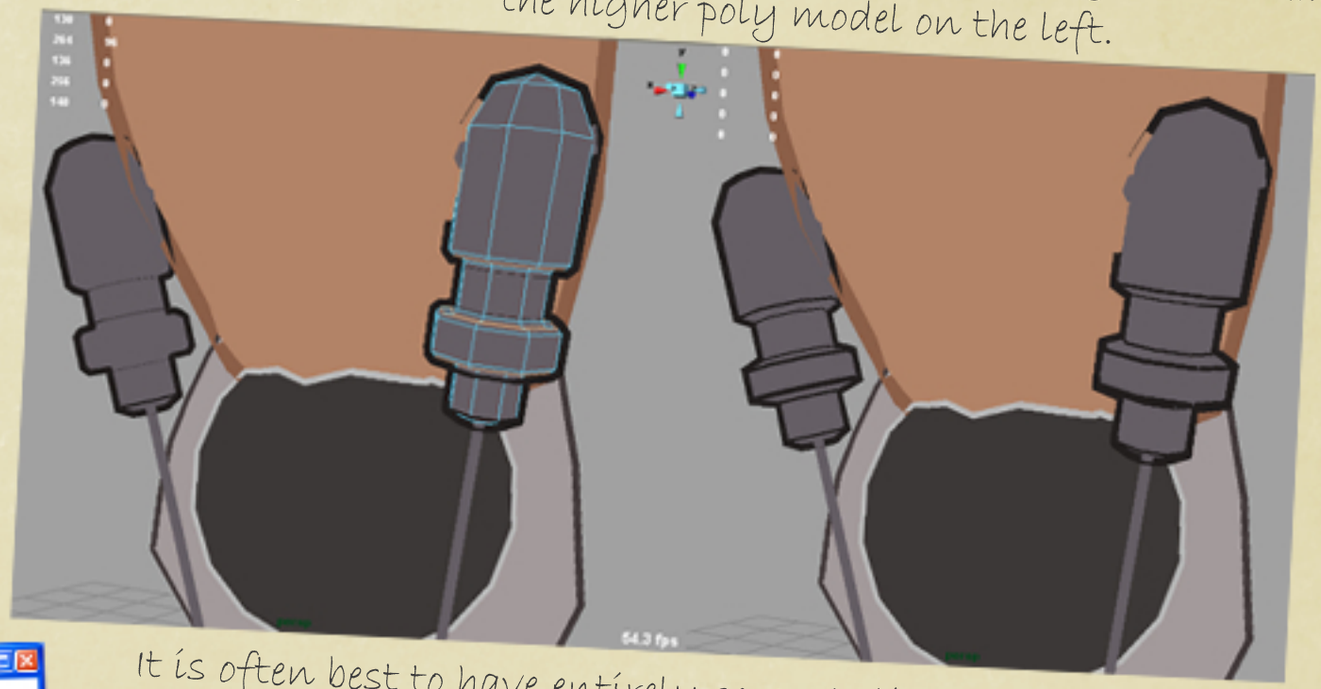
# Toon Shader Development...4

It's worth mentioning at this point that it is best to not just depend on the viewport rendering of the line work as, although useful, it is not always entirely accurate. When rendering, it is best to do so against a colour that strongly contrasts the shaders and line so that all the shades can be distinguished. Almost every component of your object is likely to need its own line shader as the width of line displayed is interpreted differently depending on the object it is assigned to. So to keep a consistent line thickness, it must be approximated by eye for each component.




With this strong green background, all the different line colours used can be identified.

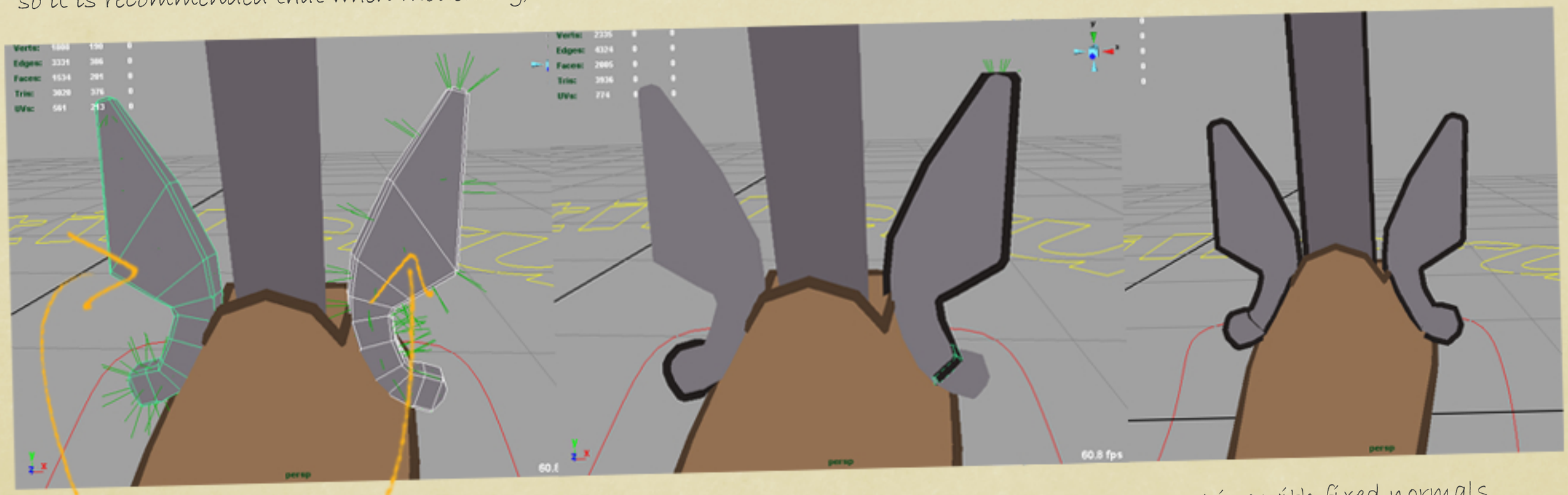
This image shows that the lower polygon model on the right draws stronger lines than the higher poly model on the left.




It is often best to have entirely separate line shaders for each objects crease lines and intersection lines as well. This is because you may be able to use the same profile line shader on multiple objects, but you may wish one of those objects to show creases in the mesh more than the other. Above is an example of an issue stated previously where sometimes a lower resolution mesh can provide better results. On the left is a high res version and you can see the crease lines within the piston do not show up very clearly. By removing the high-lit edges, the creases become much more distinct, as seen in the image on the right. The reason for keeping the intersection lines separate is because both intersecting meshes need the same shader assigned to them to work. But if they need different colour profile/crease lines, sharing the same shader is not an option. So another line shader must be created and assigned to both objects purely to deal with the intersections. This can be seen above where an orange object intersects with a dark grey object but a white intersection line is needed.

# Toon Shader Development...5

The biggest potential problem with using Mayas toon shader is with mirrored objects or objects with incorrect normals. In the image on the bottom left you can see the mesh for the wing-nuts on the feet of the character with the green lines displaying the polygon normals. They should all point out, but due to the way the object was modelled, some of the normals are facing into the object instead of out. This results in the line not drawing at all as the offset mesh created is within the object. This can be seen in the centre image. The simple way to solve this problem is to go through face by face and use the reverse normals tool. This should fix this problem as can be seen on the far right image. However, This method does not always work if you have already rigged the character, as freezing the transforms and the skinning process will over-ride this fix. This problem will cause the line shader to display over the entire object, not just the edges. If this problem should occur, the only way to work around this seems to be to export the skin weights, unbind the mesh from the skin, mirror the object, freeze the transforms, assign the shader and re-mirror it, leaving the scale in the minus. Then you are able to re-bind the skin and import the weights. This is quite an arduous task, so it is recommended that when modelling, use a basic line shader to check it all works before rigging.




Incorrect Normals

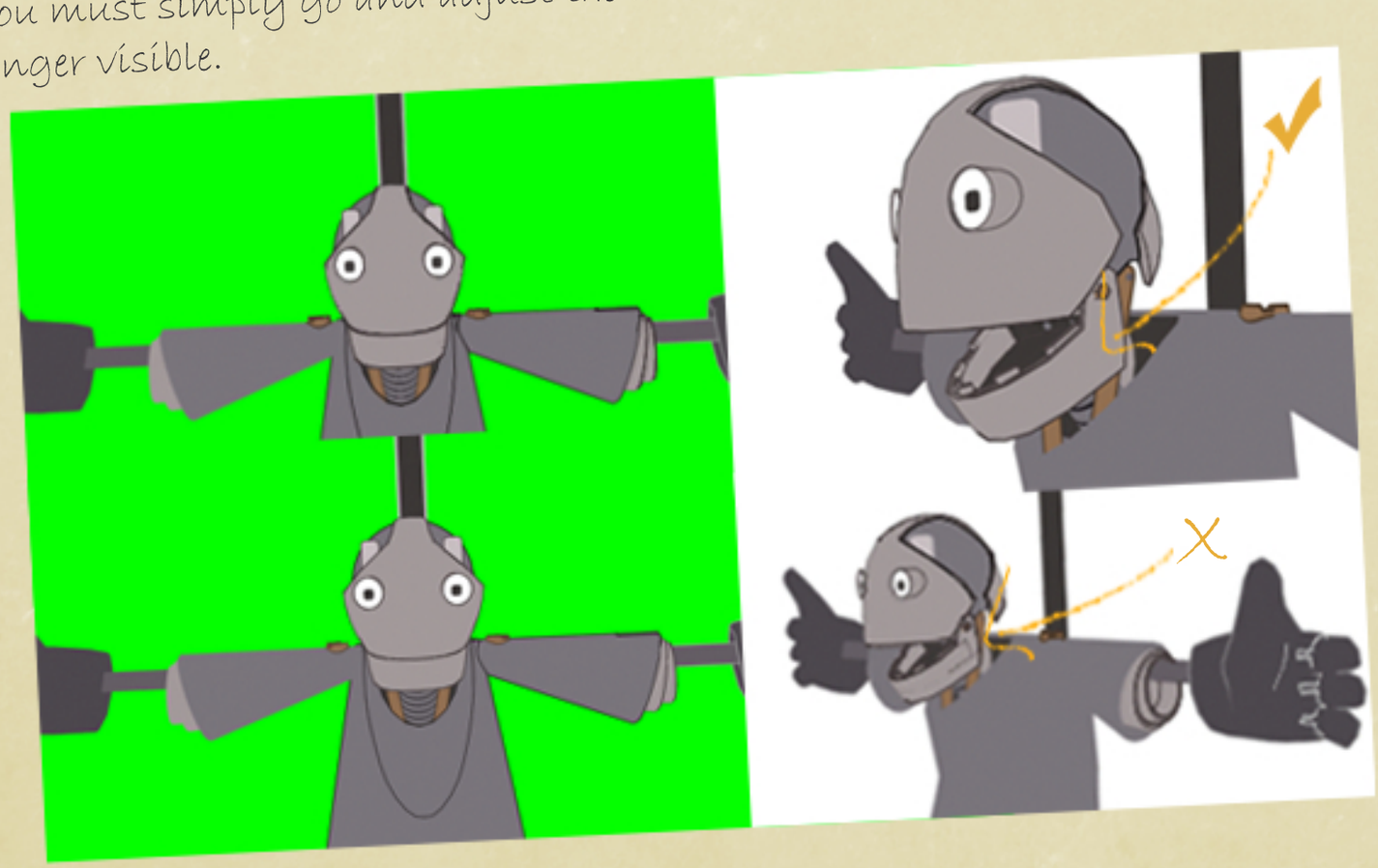
Correct Normals

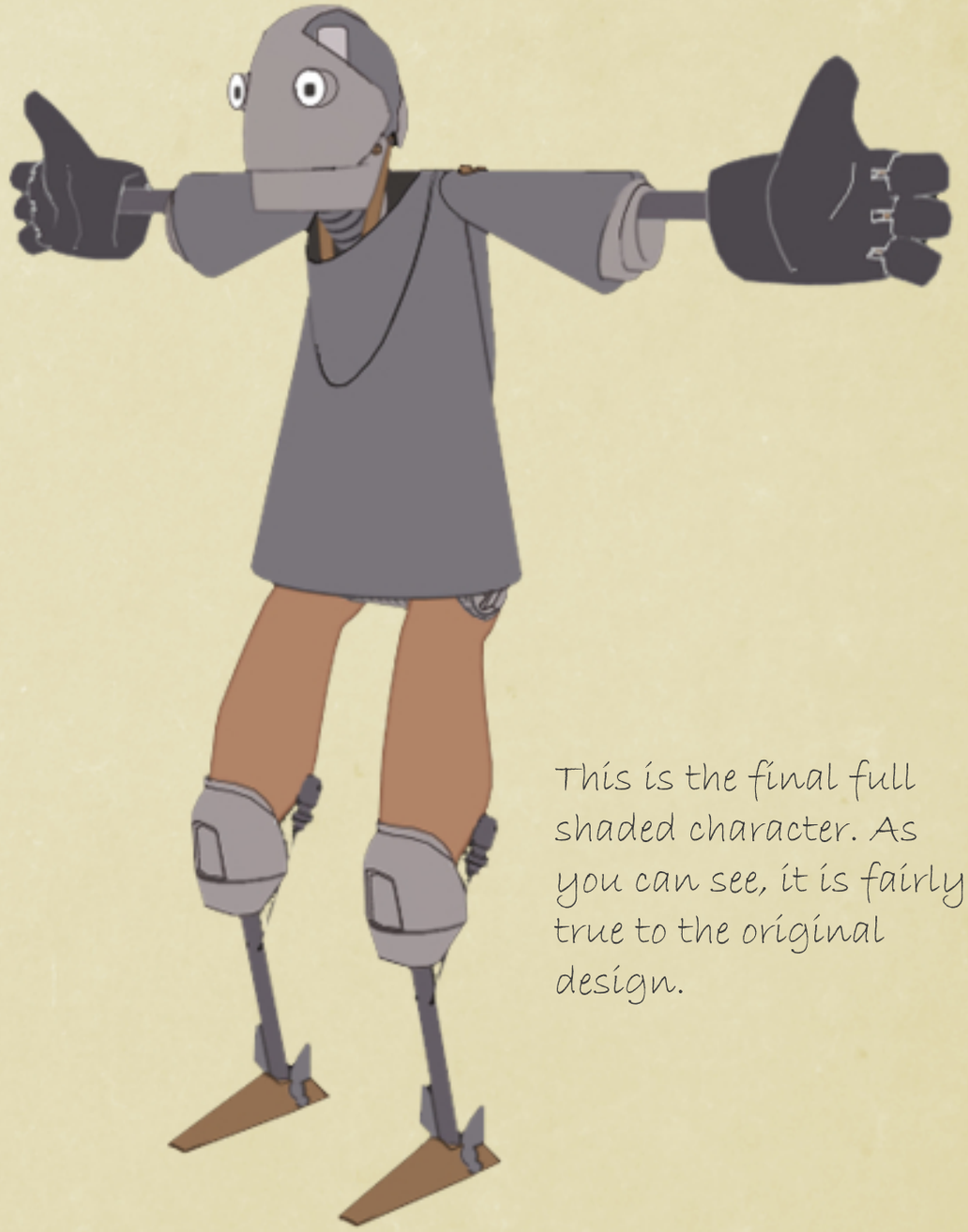
Broken line calculations

Line with fixed normals

# Toon Shader Development...6

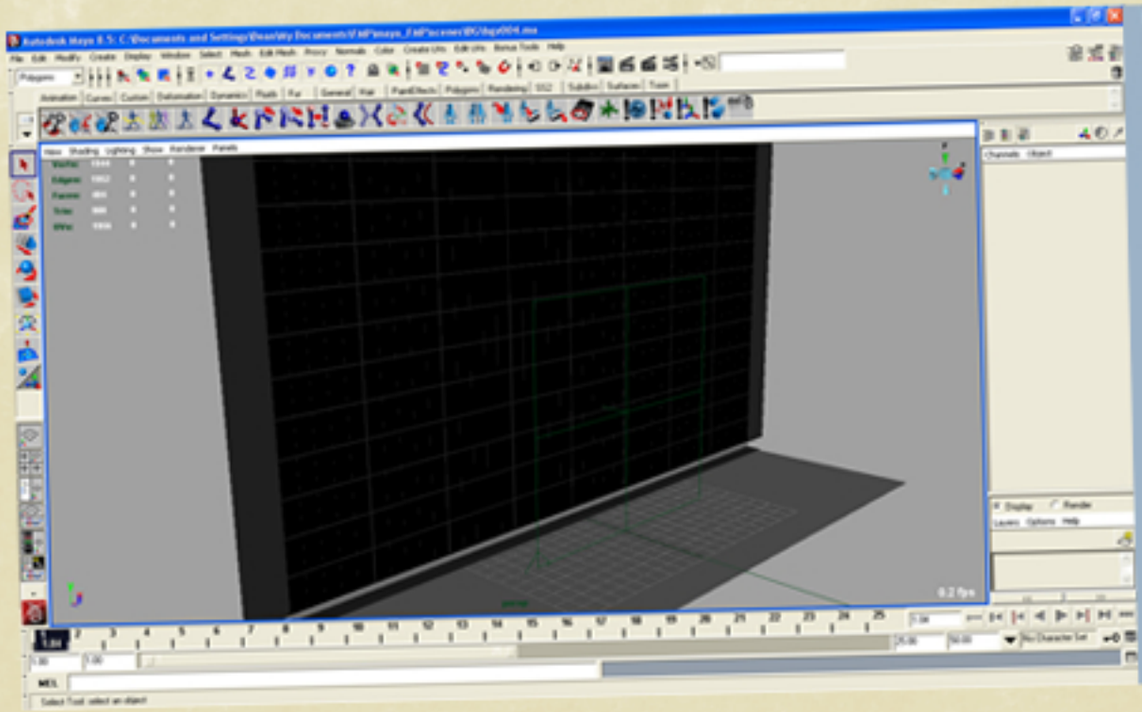
As you come to a close with your line work, its important to check how your renders appear at different distances from the camera and at different rendering resolutions. Due to the way the shader calculates the line, the best results are achieved with large resolution renders. An example of this can be seen below in the green images where the top image has a very rough outline with many artefacts in certain areas. Whereas the lower image, rendered at a much higher resolution, has a much smoother line with very few jagged areas. The camera distance issue is demonstrated in the white images where in the top image, the camera is close to the subject and it renders accurately. But in the lower white image, where the camera was placed further away from the subject, an internal line has broken through the mesh of the jaw. To solve this, you must simply go and adjust the internal mesh or line shader so it is no longer visible.

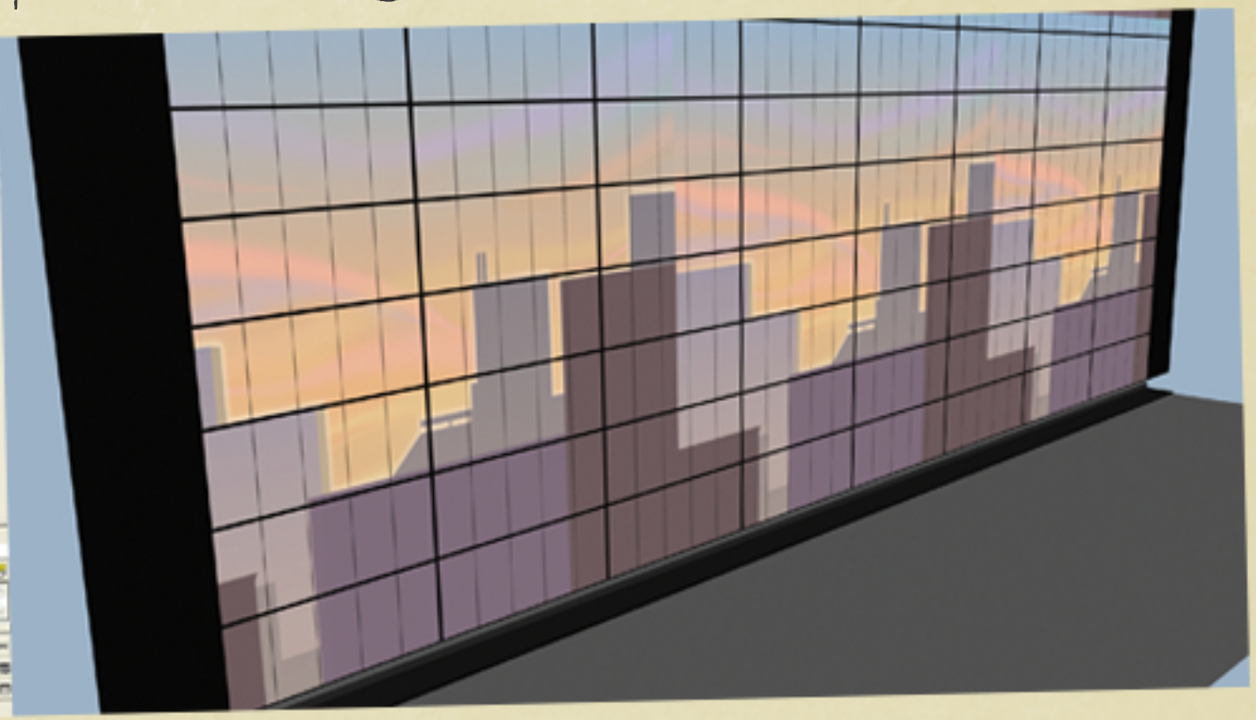


This is the final full shaded character. As you can see, it is fairly true to the original design.

# Background and Compositing...1





With the characters shaders set up, we can move on to how the 2D and 3D elements will be integrated together. The 2D animation will result in a sequence of images with alpha channels where all but the 2D drawing has been masked. Using the alpha channel produced by Maya in the rendering process we can mimic the 2D output so we have two sets of images which can be composited in a very similar fashion. Keeping the background separate from both the 2D and 3D characters gives us a good level of control as we can layer, colour correct and composite each element as we see fit. The test project is set against a large sky-scraper and so reflections are obviously an issue. Reflecting in 3D is a matter of pushing a few buttons, however 2D reflections would need to be completely redrawn for every frame, which is not an option (Unless the reflective object is at an angle perpendicular to the character, so the image can simply be flipped to mimic a reflection). So it was decided that any reflections should be done in compositing and to ensure the 2D and 3D matched, the same method would be used for both. The environment is based entirely around the windows, so as you can see above the 3D model is very simple. In addition to this, to speed up render times, instead of actually reflecting the 2D environment drawing, it is simply projected onto the windows. This would not work should there be any large camera movements as you would expect there to be problems with parallax. However, the film only shifts around the scene using cuts and if there should be any continuity issues, the projection can simply be shifted slightly.

# Background and Compositing...2

Now we have the background and foreground ready to render we can begin setting up the scenes for the various passes. Shown to the right is an example shot and its corresponding alpha channel. Unfortunately, there is a problem with the semi-transparent "Miracle Windows" screen. Maya's software shader is unable to detect the semi-transparency in the alpha and grey it out accordingly, instead we get a solid white alpha as if the object completely occludes the background. This is an issue as we are compositing the enviroment in behind this layer, so the background needs to slighlty show through the screen, but this completly white alpha would result in us just seeing the grey screen, as in the top image. One solution would be to use a Lambert material, instead of a surface shader, and then remove the diffuse value so it is not affected by lighting. Then, when using normal mapping of a PNG texture file, you do get the correct alpha. However, when using a line shader, the profiles lines are calculated after all other materials, so the lines that are behind the screen, such as on Suds' legs and the back of the basket, are drawn as if they are on top of the screen. This results in very odd renders where the shading of objects is behind, but the lines that define them are infront. The solution is to simply render any semi-transparent layers seperately. To do this, you need to assign a "Use Background" shader to every other object in the scene, so when anything goes infront of the semi-transparent layer, it is occluded correctly. Then, overlay the solid rendered version in comp' and lower its opacity so objects behind it show through. This method also gives us greater control in compositing. It can also reduce render times as, in this shot for instance, the basket does not move, so a single frame can be rendered for the screen. So now we have three render passes, a background, a foreground with the semi-transparent objects hidden and a foreground with just the semi-transparent objects displayed.

# Background and Compositing...3


With the potential layers decided upon, it's time to develop the background fully so we can move on to a full compositing test. Above to the left you can see the final environment image. Initially this image was drawn very precise and with straight edges, but as this is to be projected onto windows, to add to the glass effect the image was warped slightly. This results in the projection appearing to bend over the glass as if it were due to imperfections in the glass reflecting the environment. Above to the right you can see the test render we will use to put together the comp. In this instance, the program 'Shake' will be used, however any good compositing software such as 'Nuke' or 'After Effects' would be fine. An important note to make is at this moment in time, the 2D animation for the project under development has not been fully produced. The animation has been drawn, but not yet cleaned up, scanned and coloured. In light of this, a still image will be used as a place holder for the character Ivan, allowing his compositing network to be developed ready for when the animation is completed. The whole point of the compositing stage is to help integrate the characters convincingly into the same environment, so ideally we should be able to use near identical networks to composite the 2D and 3D characters.

# Background and Compositing...4

Environment

Suds 3D Pass

Alpha derived reflection


Suds screen

Ivan

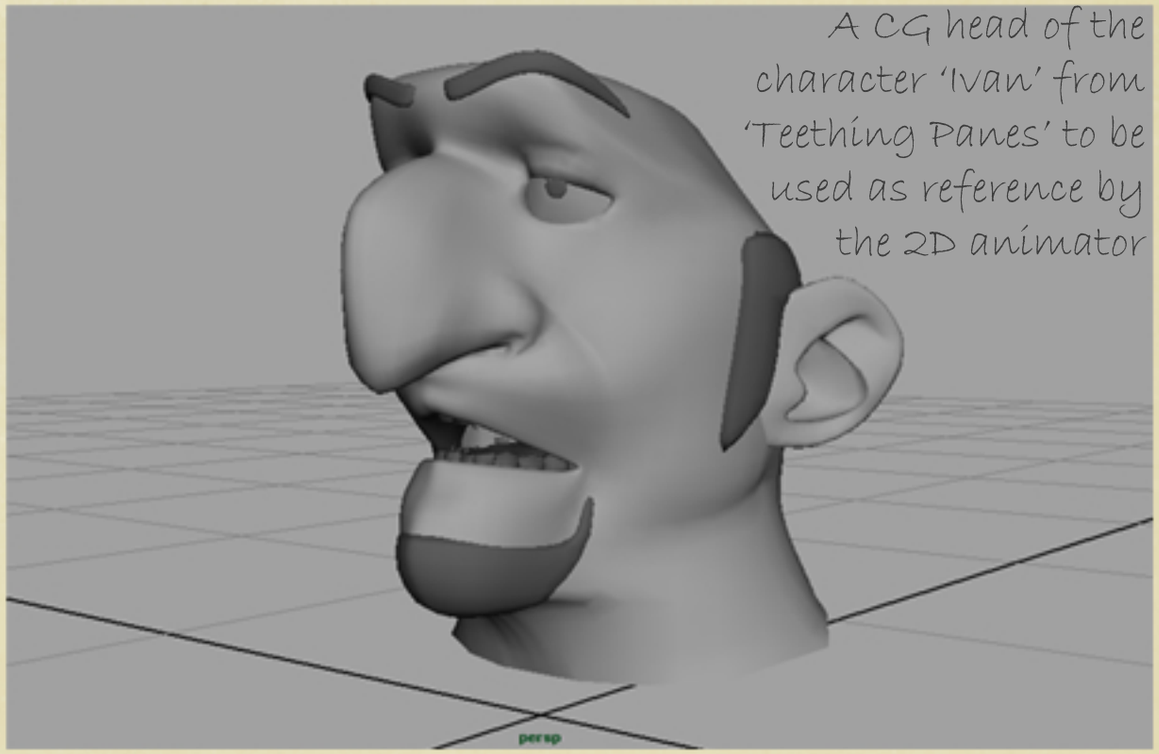


Alpha derived reflection


Final Composite

# Further notes...1

Throughout the development of any project many unexpected issues will crop up that require some fast problem solving. This section will cover a few of those issues as well as some other tips picked up through research on other individuals work in this field. One issue that initially came up was what frame rate to work at. The 2D animator is taught to work on 2's where each drawing holds for two frames of a video running at 24 frames per second, switching to 1's for fast action with 24 drawings for each second. Within the UK, PAL is an accepted standard and most CG animation is geared towards that format with 25 frames a second. This could cause a big problem, but it can be worked around using options inherent in most 3D packages. The CG animator is able to animate at the 25 frames a second they are comfortable with, and then simply change the frame rate just before render time. This keeps the second duration the same, but simply renders at different intervals within that second. This results in CG renders at 24fps, for when the cg character is alone on screen. However, when both characters are on screen at once, it is important that if the 2D animation is on 2's, so running at 12fps, the CG animation must drop every second frame to also run at 12fps. Otherwise, the two characters would not integrate well as the CG will run a lot smoother than the 2D. Again, as most UK players are geared towards 25 fps, having a final movie at 24fps could cause problems. Two ways of getting round this is to use a good converter which can re-encode the video to 25fps with no loss of quality. Or, take the 24fps images, run them at 25fps and slightly speed up the audio to match. This is obviously not ideal as the timing will be slightly different from intended, however it is a straight forward solution to an otherwise




A CG head of the character 'Ivan' from 'Teething Panes' to be used as reference by the 2D animator

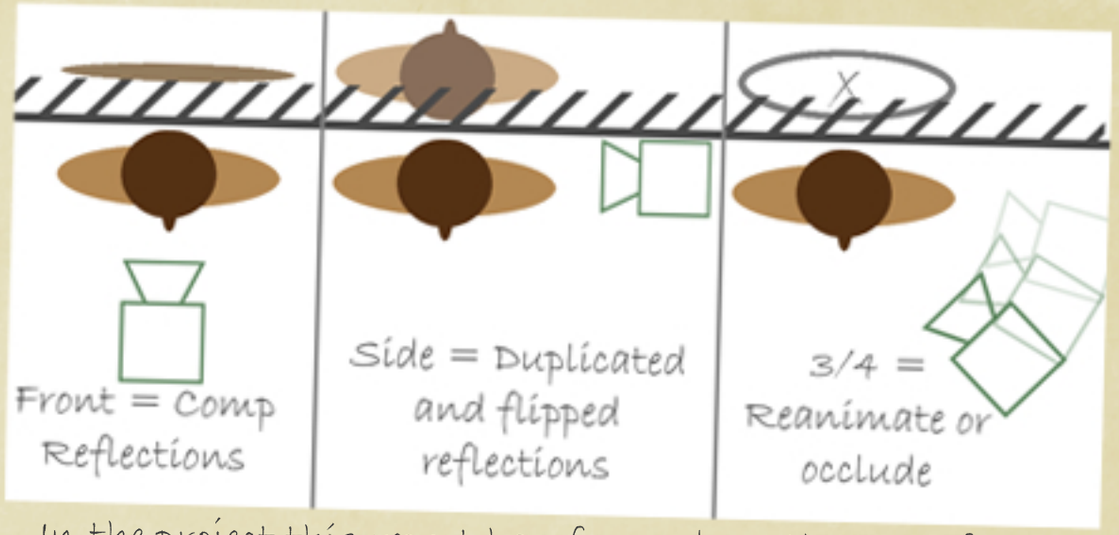
Working closely with a 2D animator you get to see many of the advantages and disadvantages of each medium. One of the biggest problems a 2D animator is constantly up against is retaining volume and maintaining shape. After hearing about this problem and reading in the paper '2D/3D Hybrid Animation on Spirit' (Doug Cooper; Dreamworks), they used 3D models as reference for the 2D animators.

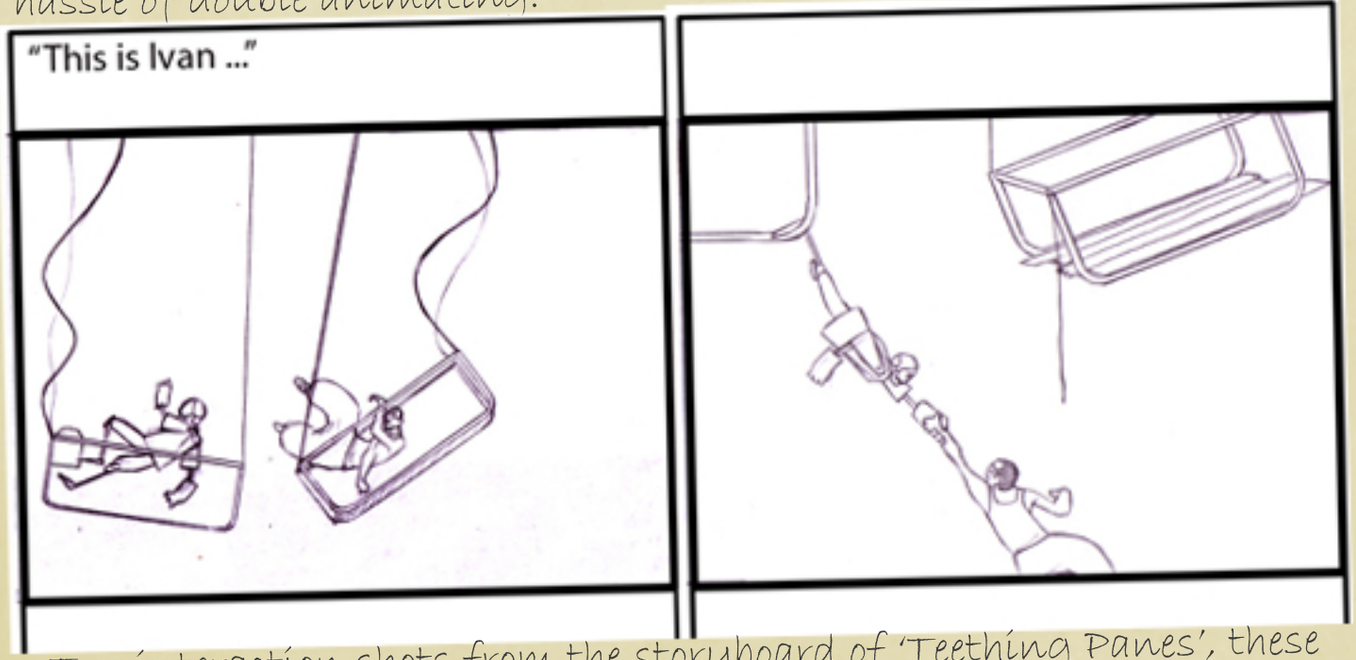
"The animators did not rotoscope the CG renders, but instead used it solely for scale and perspective reference."

As a result, to help the 2D animator visualise the character Ivan from all angles, the model seen above was put together. This allows the 2D animator to see the heads shape from where-ever he chooses and has been a great help in almost every shot so far.

# Further Notes...2

As mentioned at various points throughout this report, tackling the reflection issue has been a constant battle. It is important to make sure there is evidence of it, without having to force the 2D artist to reanimate the sequence. This is why using the alpha channel to create a false flat reflection in compositing was a basic, but reasonable method to use for frontal camera angles. For completely perpendicular camera angles, the character can simply be flipped and colour corrected to create an accurate reflection, but it is the 3/4 camera angles that cause the real trouble. Obviously, those shots could just be avoided by never placing the camera in that situation, but it certainly adds more variety to the film making if those shots are not completely ruled out. If animating twice is not an option, the best road to go down is to hide the area where the reflection should be. Either using other objects, or in the instance of this project, soap suds and water from the characters window cleaning can occlude the reflection allowing more freedom of cinematography, without the hassle of double animating.




Two interaction shots from the storyboard of 'Teething Pains', these shots must be very well planned for the animation to match.

In the project this report has focused on, there are four shots that not only require the 2D and 3D characters to be on screen together, but they also have to physically interact with each other. The two characters never exist in the same scene until compositing so how to make sure they match was an interesting problem. Assuming that one would have to be animated first, it makes sense for the CG artists to take the lead on these shots due to the level of control they have over correcting and tweaking animation. It was also found that using a place-holder for the 2D character, some very simple animation could be done in 3D for the 2D artist to use as a character guide. Rigid structures moving around the scene and rotating are very difficult to animate in 2D as unlike characters which can change shape, rigid objects must retain volume and form. It was found that animating these objects in 3D first would ease the work-load on the 2D animator considerably. With these CG placeholders, the scenes can then be rendered and printed out for the 2D artist to trace over and overlay his own animation. This ensures that come the compositing stage, the two characters will match at all times.

# Further Notes...3

A factor seemingly overlooked so far are shadows. Seeing as it was a style choice to stick with flat shading for the project 'Sky-caper' the issue of dealing with different tones has not been dealt with in this report. However, Mayas toon shader is capable of providing some interesting results and there are many other toon shaders available that provide a very different look. One worth mentioning is the X-Toon shader (X-toon; Pascal Barla) which rather than just using a 1D shader to vary tone, uses a 2D shader which uses the tone variation and a lambert function to create layers of varying shade detail. Examples of which can be seen below. It also adds some nice depth of field opacity control, similar to the way 2D deals with distant objects. On the same topic, this report has not covered shadows cast onto the scene by the characters. It was chosen that for the project 'Teething Panes', these shadows were not necessary as the character would be in a cleaning basket and the main integration factor would be the reflections in the windows. Traditionally when integrating 3D into 2D, the shadows would be drawn by hand to make sure they cast over the 2D environment correctly. Although the paper 'Shadows for Cel Animation' (Lena Petrovic) looks at some interesting techniques in producing accurate shadow mattes by replicating enviroment and lighting conditions in 3D space.

Many different applications of the X-Toon shader can be seen here

# Conclusion and Evaluation

This has been a guide on the production of toon shaders for 2D integration and the recommended steps to take that could make the process develop smoothly. The processes discussed have been derived from first hand experience and it is hoped they will save others from making the same errors along the way. Creating the shaders is fairly straight forward, but the use of the line and pre-planning at the design and modelling stage played an integral role in the success of the final renders. If I were to develop 2D shaders for another model, I have certainly learned to test all the line work at an early stage, to prevent much larger problems occuring after rigging or having to carry out fixative modelling.

The shaders developed are versatile and create a fairly convincing 2D image for most angles and character movements. Although there were problems in further animation when certain objects intersect with other parts of the character they were not initially set up to collide with. This would require some further refinement in the line shader to ensure all possible eventualities are accounted for. At the compositing stage, the Shake scripts developed to overlay and integrate the two types of animation are re-usable and animate-able so they will work on multiple shots and allow some animation of the layers. Although the reflection solutions offered via the use of compositing are successful in their individual application, to convincingly reflect in 2D from any angle would still require a complete redraw of the character.

# Conclusion and Evaluation

Near the closing stages of the investigation I discovered an interesting point about the movie 'Iron Giant'. The technicians wrote a program that would cause the line of the character to wobble slightly, adding to the imperfect hand drawn appearance. As a further development to the line shaders I built, the next step could be to write my own program to create the same effect. There are other line effects that I could develop in the future, such as creating slight artifacts to mimic pencil lead as well as line width variation. Ironically, these are all issues that alot of traditional 2D animation wishes to get rid of!

Although not relevant to the project seen throughout this report, I have been in contact with David Lanier who helped develop the X-Toon shader mentioned previously. And example of this shaders application can be seen below from the film "Ryu" (Andy Alesik). The shader is currently only available in beta and I have a copy that I intend to carry out some tests with.


Overall I would say the final outcome has almost completely met my initial goals. Many lessons have been learnt and interesting challenges put forward while trying to bring these two mediums together. It is hoped that others can use this report as a stepping stone in developing their own shaders in their projects, as the gap between 2D and 3D animation slowly shrinks.

# References


PAPER:

Doug Cooper, 2002, '2D/3D Hybrid Animation on "Spirit", Dreamworks animation' - Sourced through ACM portal.
Lena Petrovic, Brian Fujito, Lance Williams, Adam Finkelstien, 'Shadows for Cel Animation', Princeton University, Disney Feature Animation. - Sourced through ACM portal.
Pascal Barla, Joelle Thollot, Lee Markosian, 'X-Toon: An Extended Toon Shader', Artis Gravir/Imag Inria, University of Michigan. - Sourced through ACM portal.
Brad Bird, 'Director and After Effects: Storyboarding Innovations on The Iron Giant', Animation World Magazine, 1998.

FILM:

Who Framed Roger Rabbit, 1988, [Movie], Directed by Robert Zemeckis, USA, Amblin Entertainment.
The Lion King, 1994, [Animated Movie], DIrected by Rogers Allers and Rob Minkoff, USA, Walk Disney Feature Animation
Tarzan, 1999, [Animated Movie], Directed by Chris Buck and Kevin Lima, USA, Edgar Rice Burroughs Inc and Walt Disney Feature Animation.
The Iron Giant, [Animated Movie], 1999, Directed by Brad Bird, USA, Warner Brothers Animation.
Oilspot and Lipstick, 1987, [Animated Short], Tad Gielow and MJ Turner. USA, Walt Disney Pictures.

WEBSITE:

Non-Photorealistic, Painterly and 'Toon Rendering an annotated survey of online resources - http://www.red3d.com/cwr/npr/ - Craig Reynolds - 2003
Toon shader 1.0.4, http://www.davidlanier3d.com/index.php?lang=english&option=com_content&task=view&id=35&Itemid=63, David Lanier, 2009
Constructing a Cartoon Shader for MAYA - http://www.vrvis.at/vr/cgr4/cartoonshader/index.html - Anton L. Fuhrmann - This page was last edited: February 5, 2004

IMAGE:

Lion King Stampede - http://www.lionkingpride.net/images/screencaptures/stampede.jpg - TLK Pride Backup© 2006
Oilspot and lipstick - http://bp3.blogger.com/_dVGtulJZVFk/R6Z1WrFD09I/AAAAAAAAAE0/Uek8LR54qyo/s1600-h/oilspot_lipstick.jpg - March 24th, 2008 - drawn2gether
Ryu X-Toon image - http://www.davidlanier3d.com/images/stories/toonshader/dl3d_toonshader_ryu004.jpg - 2008 - Andy Alesik - www.alesik.de