Lahden ammattikorkeakoulu
Lahti University of Applied Sciences

FACIAL ANIMATION OF GAME CHARACTERS

LAHTI UNIVERSITY OF APPLIED SCIENCES
Faculty of Technology
Media Technology
Visualization Engineering
Bachelor’s Thesis
Spring 2015
Kalle Wallin

Lahti University of Applied Sciences
Degree Programme in Media Technology

WALLIN, KALLE: Facial animation of game characters

Bachelor's Thesis in visualization engineering, 41 pages

Spring 2015

ABSTRACT

Facial animation in games has increased significantly in the past ten years. This is why the thesis introduces the basic technology in facial animation. The thesis only covers the basic tools and techniques used to create facial animation of game characters. The software used during this thesis were Autodesk's 3Ds Max and Mudbox, and Substance Painter by Allegoritmic. The basic tools for creating game assets were explored.

First the thesis goes through the basics of modeling 3D objects for games. Then it deals with rigging technology and finally it presents emotional animation and its creation. The case deals with most of the techniques mentioned in the thesis. The goal of the thesis was to show the basic technologies and techniques to animate the faces of game characters and to create a prototype for a game character's face for further development.

Key words: animation, modeling, 3D, rigging, game asset

Lahden ammattikorkeakoulu
Mediatekniikan koulutusohjelma

WALLIN, KALLE: Pelihahmon kasvojen animaatio

Mediatekniikan opinnäytetyö, 41 sivua

Kevät 2015

TIIVISTELMÄ

Kasvoanimaatio peleissä on yleistynyt merkittävästi viimeisen kymmenen vuoden aikana. Siksi tämä opinnäytetyö esittelee perusteet pelihahmon kasvoanimaatiosta. Opinnäytetyö sisältää ainoastaan yleisimmät työkalut ja tekniikat, joita käytetään pelihahmon kasvoanimaation luomiseen. Opinnäytetyön aikana käytössä olleet ohjelmat olivat Autodeskin 3Ds Max ja Mudbox sekä Allegoritmicin julkaisema Substance Painter. Näistä ohjelmista tullaan selvittämään yleisimmät työkalut, joita käytetään peleissä käytettyjen mallien luomiseen.

Opinnäytetyö käsittelee ensimmäisenä 3D-pelimallinnukseen liittyviä perustaitoja. Seuraava osio käsittelee riggaus tekniikoita ja viimeisessä osiossa havainnollistetaan tunneperäisten seikkojen animaatiota ja sen luontiin käytettäviä tekniikoita. Projekti-osiossa käytössä on suuri osa teoriaosuudessa esitellyistä tekniikoista. Opinnäytetyön tarkoituksena oli näyttää yleisimmät tekniikat pelihahmon kasvoanimaatiosta ja luoda tietokonepelissä käytettävän hahmon jatkokehitystä varten prototyyppi pelihahmon kasvoista.

Avainsanat: animaatio, mallinnus, 3D, riggaus, pelissä käytettävä luomus

TERMS

Ambient Occlusion	A shading technique used to calculate ambient lighting and shadows.
Baking	Drawing information based on the high polygonal 3D mesh on a bitmap
Digital sculpting	Using computer software to push, pull and manipulate a digital object like it was made in real-life clay.
Diffuse map	A base color bit map for a 3D object.
Normal map	A bitmap containing height information of the 3D object.
Rig	A “skeleton” used to animate a 3D object.
Shader	A container for all of the texture maps.
UVW mapping	Laying out the surface of the 3D object onto a 2D bitmap.

TABLE OF CONTENTS

1 INTRODUCTION

The best known landscape for human beings is the human face. People have the ability to read even the slightest changes in facial expressions as it is something we are born with. Most experts believe that the most common and fundamental facial expressions such as expressions of fear, joy, surprise and sadness have remained as they are for thousands of years. That may be one of the many reasons why artists have been trying to capture facial expressions very accurately, as it can have a decisive impact on the effect of a picture. (Faigin 1990.) The same applies with emotions and expressions in games. A game with stunningly realistic expressions gives a player chance to have a deeper connection with the game.

3D low poly characters made their appearance in the 1990s when low poly characters were needed due the low computer power. Technology has improved significantly and today game characters can contain thousands of polygons. The name has still remained in use, as the normal and displacement maps are baked from the high poly models. Nowadays, to achieve a balance with realism and a reasonable framerate when rendering objects in realtime, an artist must make compromises with the level of detail and texture resolution.

This thesis aims to recognize the advantages of realistic facial expressions in games to achieve memorable experiences. Working with 3D models and animating facial expressions requires a fundamental knowledge of 3D modeling and animation pipeline.

2 ANIMATION IN GAMES

The animations took a huge leap forward into 3D in the 1990s. Video games started to be made in 3D, which meant that the characters had to have many poses and move in all directions. *Super Mario 64* was one of the games made in the 1990s which was to influence the future of 3D gaming greatly. As the character moved or jumped, Mario's legs and arms would make slight movements and he was even able to do flips. From this we have come to a point where almost every video game uses character animation. (Masters 2014a.)

When making animations for games it is essential to keep in mind that the characters and environment are meant to be interactable. This means, it is not enough that the animation looks good, it has to look good from every possible angle. (Masters 2014b.) A good example of game animation is idle stance (when the character is doing nothing and standing still). The player would consider it boring if there was no subtle movement, like breathing in and out while moving slightly from side to side. A majority of games today are using body mechanics, so for the animator it is crucial to spend a lot of time studying body movement in different situations. (Landgraf 2012.)

2.1 Evolution

As the computer and game engines are progressing people expect more and more animation to be implemented. For example, the first reload animation in games was implemented in Medal of Honor in 1999, but now if a first person shooter lacks reload animation, that is just absurd. During the 2000s there were two generations of gaming consoles where more complex animations were introduced in each game. (Masters 2014a.)

Today players want to have more control over the character and the main character needs to feel as real as possible. By creating more empathy for the character, gamers become more involved in the game. A good animator is someone who understands the body mechanics, physics and weight, but also goes beyond a solid walk cycle (Landgraf 2012.). The studios want to see someone breathe life into the characters. This has led to a situation where more animators are needed to create a huge amount of different animations for every situation. The studios are using more and more close-up facial animation, and subtle character traits that you would see in animated movies, are now being put into video games.

As the technology goes forward and more and more graphics can be rendered in real-time, game developers are pushing the limits to reach more realistic experiences. A good animation makes the game character feel more like real, living and breathing people. If the game is focusing on realistic graphics and body mechanics, it needs at least the same level of realistic facial animation to keep the player empathizing with the character.

2.2 Game animation today

Today game animations have reached the level where they can almost be compared with fully animated feature films. Also some of the techniques from film making have been implemented in the game industry. An example is the use of motion capture. During this generation of gaming consoles, animation in video games has increased tenfold. The animators can use more compex rigs, giving them more control over the character. Animation has clearly come a long way from where it started in the 1990s. Then we would not have seen dynamic hair simulation in real-time, but now in a game like *Tomb Raider: Definitive Edition* we can see impressive dynamic hair without sacrificing the game experience (Image 1). From games like *Middle-earth: Shadow of Mordor* one can really see the effort and time that has been spent on the animations. There are numerous unique attack and dodge moves that fluidly transition into a different animation. The best of all, there are no glitches or hitches and the movement is as fluid as one expects when watching a movie. It can also help a game achieve more realism, as some games have implemented different walk and facial animations if the player is hurt. (Masters 2014a.)

IMAGE 1. Tomb Raider: Definitive Edition. (USgamer 2014)

2.3 Software

There are numerous programs available for 3D animation. Some of the most commonly known programs are Autodesk’s Maya and 3Ds Max, Maxon’s Cinema 4D, and Blender by The Blender Foundation. Many of these programs are specialized in a specific category, for example digital sculpting and each has some specialities that the other programs do not have. For example Autodesk’s Maya has Maya Live’s motion tracking tools. Mudbox by Autodesk has been focused on digital sculpting, but it also includes tools for retopologizing and 3D painting.

There are also different applications for texturing. A few examples are Substance Painter by allegoritmic, Mari by The Foundry, and BodyPaint 3D by Maxon. Texturing programs today are using the PBR system, which means physically based rendering. In Image 2 you can see the interface and lighting in Substance Painter. This allows users to view the changes made in reflections and specularity as they paint. It is also possible to rotate the lighting in the software. Programs that have become more popular today are those that can be used to generate maps needed for the textures.

For preparing the model for texturing there are also programs that are focusing on using the retopologize tools. This basically means that the software allows the user to reduce the number of polygons and adjust the edgeflow of the model. This is a crucial part of creating game art, as it affects the deformations of the model greatly. A good topology also allows artists to create clean uvw maps for the actual texturing.

IMAGE 2. Interface of 3D painting software.

3 MODELLING A CHARACTER FOR GAMES

There are many ways to build 3D models, but the most commonly used way is so called polygonal modeling. This means that the model is created by moving polygons, edges or vertices in 3D space. A vast majority of 3D models today are built as textured polygonal models, as this is the fastest for the computer to handle. Polygonal 3D models are categorized as high polygonal and low polygonal models depending on the density of the polygonal mesh. (Wisegeek 2014.)

The low poly models are used in games to save computer performance and production speed, but the most important question when modeling a character is to know where and how it is going to be used. The character should always be approached as a simple base-mesh and the modeler should try using subdivision surfaces to smooth out the model and see it in higher detail. Detail should be added only when it is truly needed.

Creating a character for a game requires knowledge and expertise in many other fields than just modeling. A good character artist should also understand texturing, animating and know how to create UVs properly. The most important tool for artists is the game engine, as it defines the shading properties for the final textures. (Masters 2015.)

3.1 Low poly modeling

The polycount as a term is a bit misleading, because usually the modern hardware is built to render triangles. The “polycount” of a game character varies quite a lot and it needs to be targeted for the platform you are going to use. For example with mobile devices the polycount should be between 300 and 1500 whereas for desktops the ideal range varies between 1500 and 4000 polygons (Unity Documentation 2014). Usually artists try to keep the model in four-sided polygons (quads) as long as possible to make it easier to weight a skinned model to its bones when animating.

The big question for starting game developers is: “How many polygons should be used?” As was mentioned before, it depends greatly on the platform and the use of the model. This is why the answer is impossible to determine. Artists and developers have to make compromises in order to keep the game both good looking and smooth. In most cases the amount of detail needed can be achieved with normal maps, which is why artists should always try to avoid adding too much detail in the model itself. The most common use of normal maps are for objects close to the surface, like a sock or jewelry. During the modeling process it is also essential to retain the model seamsless and use as few added elements as possible, especially if the model is going to be animated. This is because the separate elements can cause problems in rigging. (Ward 2011.)

3.1.1 Topology

When modeling low poly models it is crucial to keep the topology clean. Topology in modeling means the ability to respond correctly to the grid deformations such as skin stretching, squashing, twisting etc. The idea is to lay out the edgeloops in such a way that they follow the contours of the muscles and other key forms (Williamson 2012.). This allows the polygons to deform correctly and follow the natural muscle movement. Topology becomes more and more important when using fewer polygons, in other words it is the most essential when making content for games. Frequently the best way to insert isoparms in the face are those that closely resemble the layout of muscles in a real human face (Logue 2015.).

When determing the topology for the head, the main edgeloops are the same. You can see a reference for the topology in Image 3. There is one edgeloop for the forehead and the jawline, which defines the shape of the skull. This also leaves room for wrinkles to form on the forehead when expressing surprise. The second common edgeloop runs through the nosebridge and below the mouth. With this edgeloop it is easier to define nose wrinkles and chin shape. Artists also divide edgeloops around eyes in two parts, one edgeloop just around the eyeball and the second edgeloop that reaches the bottom parts of the brows. The mouth usually needs a small edgeloop just around the lips to get more control over the deformation of the lips when animating speech or facial expressions.

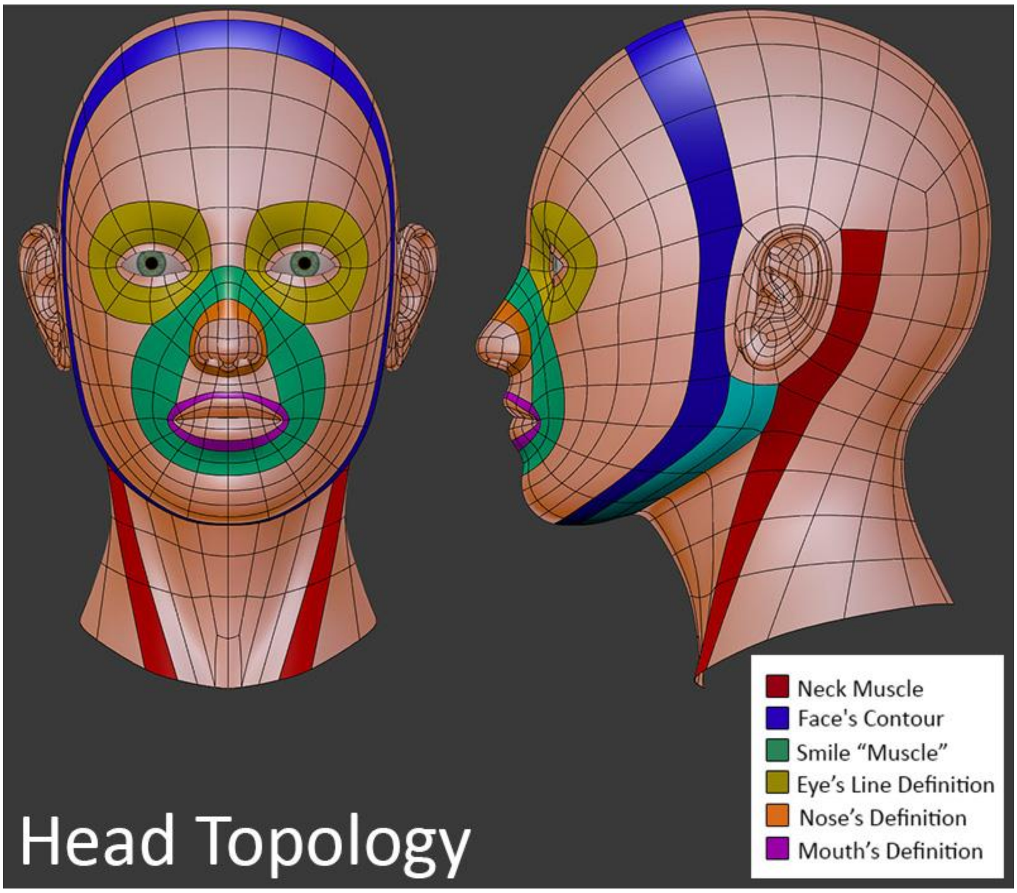

IMAGE 3. Example of topology of the face. (Blenderartists 2014)

3.1.2 Workflow with characters

Workflow with 3D modeling is basically always up to the artist. Many artists find that the best way to start modeling a character is to have a clean base mesh. Many programs have their own base meshes to start with, but as the base mesh should follow the main edgeflow of the character it is sometimes needed to create new ones.

The most widely used workflow with game characters is to make a base mesh, and start sculpting a very detailed high poly version of the character. Then all the detail in the character’s clothes or equipment are usually modeled to the character. This phase is called hard surface modeling. When modeling hard surfaces for the character it is essential to keep in mind how the character is going to move and how the surface is going to react. A good example of a character like this is a medieval knight, where the armor must react to the movement of the character, see Image 4.

IMAGE 4. 3D character of a knight. (Turbosquid 2013)

After the high poly character and other models are ready, the high poly models are used with retopologize tools to create the low poly version. The low poly version’s polycount is around 1000-4000 polygons. Sometimes the created hard surfaces cannot be retopologized, so they need to be created again as low poly versions.

3.2 Textures

In 3D, texture mapping means adding graphics to a polygon object. Textures are the best way for artists to keep the game quality high. Texturing for games requires a high level of detail because where there is low resolution geometry, the textures must be able to hide it (Masters 2014c.). A good game character uses many texture maps to achieve a detailed look without the geometry. One of the major uses of textures in games is to present the needed detail in the character that would otherwise require a lot of very small polygons to make, if it was modeled (Logue 2015.). Repeating textures are very popular, especially in games. This is because a small texture map can be used multiple times. See Image 5.

The most important thing when making textures for games is to keep the image size in powers of two. Most game engines do not accept texture files that have different dimensions. Game engines also use shaders that define how the light is reflected or absorbed or if the surface is translucent. (Masters 2014c.) Game engines today are using physically based shading, which means the reflections and shadows are based on the light system. This feature enables artists to use much more realistic surfaces in their models.

Texturing almost always requires that the model has been UV mapped. This means that the surface of the 3D model has been presented in flat 2D bitmap. The basic idea behind UV mapping is that the flat 2D image can then be wrapped around the 3D object without stretching or distorting. Artists often use different style of maps to ensure that the UVs do not stretch. Multiple objects can share one UV mapping layout to save file size, which is convenient especially when making games for mobile platforms. (Chang 2006.)

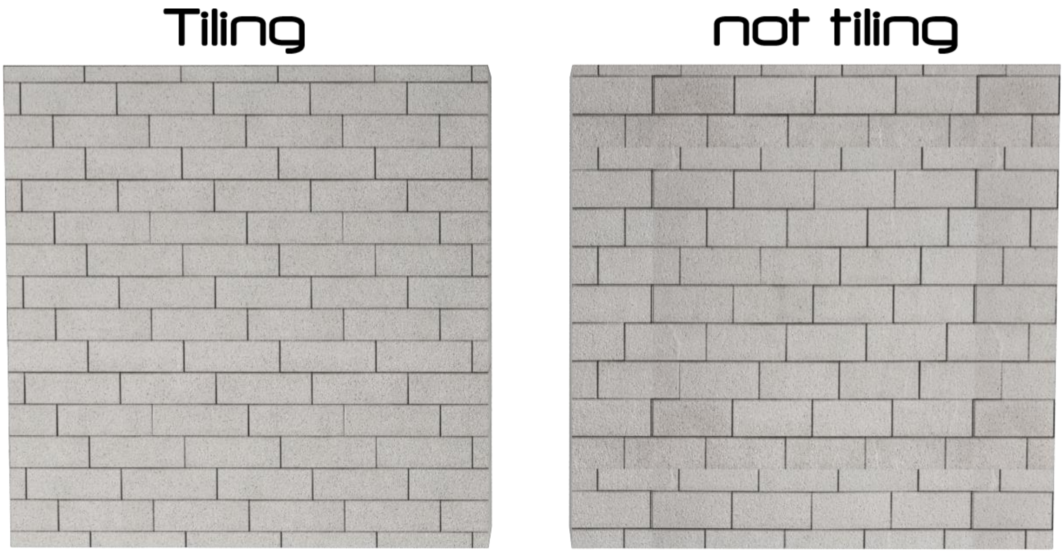

IMAGE 5. Comparison of tiling texture and non-tiling texture.

3.2.1 Shaders

As textures are created, they are combined in an object called a shader. Each texture should be applied to its proper channel on the shader, so it will display how the textures affect the model. The shaders often have adjustable parameters for the values of each texture, for example a normal map slot usually has adjustable values for the strength of the bumpiness. The reason shaders are used is because they grant their user an easy way to control the parameters. The artists can control how the model interacts with light, or if the model needs to include opacity. A typical shader in a game engine today has separate channels for diffuse, specular map, normal map, height map, occlusion map and emission map (Unity Documentation 2015.). These features in the shader allow artists to start using physically based shading to create illusion of real time light reflections. Every software has its standard shaders to use with the models. For example the new standard shader in Unity 5 is a powerful shader, and it is quite possible that every material used in a game is done with that one shader.The maps for Unity’s standard shader can be seen in Image 6 below.

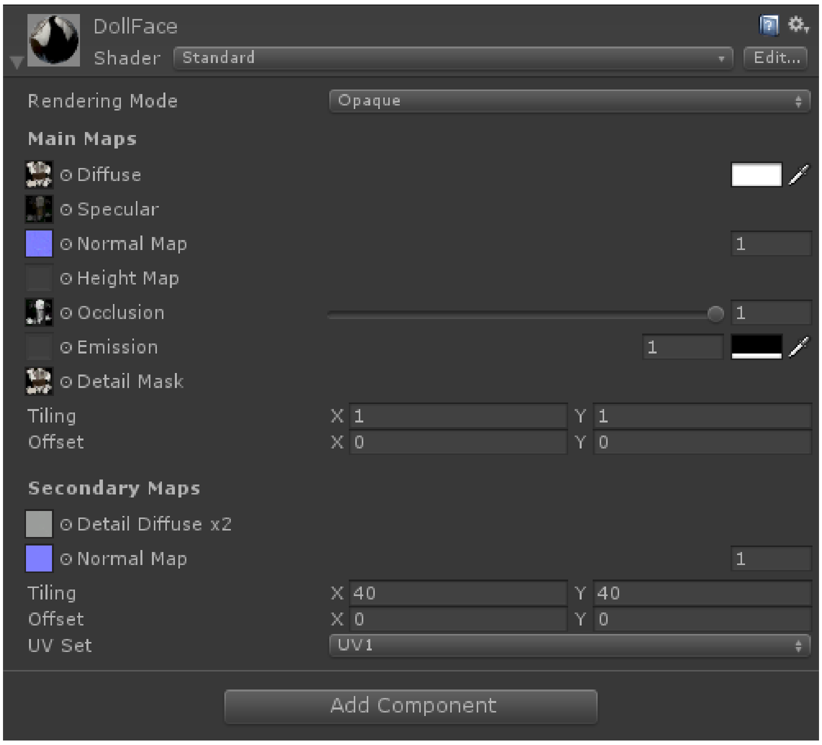

IMAGE 6. The standard shader in Unity 5. (Thecreativechris 2014)

3.2.2 Diffuse map

Diffuse is the property of a shader that defines the color of the object. It is based on the UV map of the model. Diffuse maps should not have any directional lighting included, but they can contain generic ambient occlusion to create better illusion of shadows.

Ambient occlusion is possible to bake from software like 3Ds Max. Baking an ambient occlusion map means that the computer calculates the parts where a shadow is cast, and renders it as black, and everything else is white or different shades of gray. The baking of ambient occlusion gives you a darker surface around embossed detail and cracks. This map is often multiplied with the original diffuse map.

3.2.3 Normal map

Normal maps can be considered as height maps. The idea is that the pixels of the normal map store a vector that describes the surface slope of the original high poly model. These vectors are divided in red green and blue. The red value in a normal map is the highest on negative x, the green value is the highest on negative y and the blue value is the highest on positive z. An example of a normal map can be seen in Image 7. The same image shows the effect on the surface.

Normal maps are used to create detail, or at least an illusion of high detail. They are baked from the high poly and used in the low poly models. Detail can be simulated in a very believable way by modeling a good high resolution mesh and baking the details to a normal map. Baking normal maps means that the height information can be pre-computed using the high poly and low poly models. The computer calculates the differences in height information and then renders the result. The problem with working with normal maps is that the extrusions on the high poly model should be sloped to get the best results.

Sometimes it is necessary to create normal maps from a photograph or texture. This can be done with NVidia Tools NormalMap Filter in Photoshop, or a 3rd party application such as Bitmap2Material or Crazybump. There are also programs for texture painting that enables painting displacement or normal maps with the use of stencils that contain a bump map.

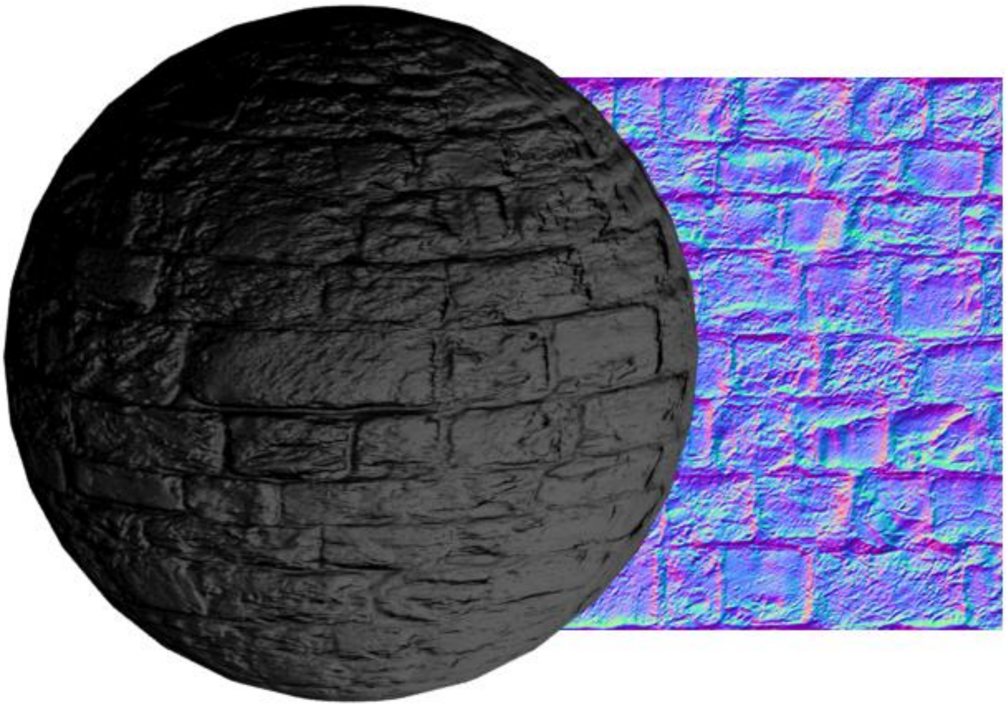

IMAGE 7. Normal map and its result.

3.2.4 Specular map

Specular maps are used to define the shininess and highlight colour of an object. The programs read the bitmap in black and white colour. The higher value of a pixel, the shinier the surface will appear in-game. For example surfaces like polished chrome would have a light specular map, while surfaces like dry stone tend to have very dark specular maps. See

Image 8, where the character's armour has some specularity through the use of a specular map. The most common way of using specular maps in a game is to make something stand out more when light hits it. That also makes the surfaces look more realistic (Polycount 2014.). Today game engines support physically based shading, which affects specular maps a lot. Shiny surfaces are lit by real-time raytracing, which enables more and more realistic lightning in the game.

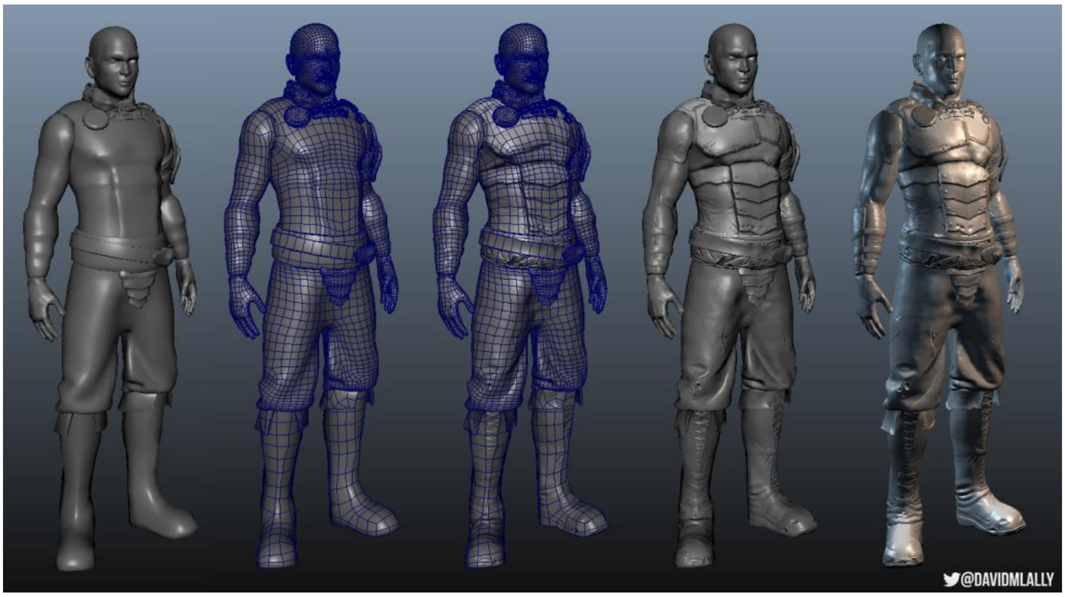

IMAGE 8. Rightmost lowpoly using specular map. (Davidmlally 2013)

3.3 Character principles

One on the most important aspects when working with games and especially game art is the audience. For game studios it is very important to know in what sort of environment the character will appear in a game. Many game series have come up with their own style, and players expect the studios to stick with the original style. Often these characteristics can be found exaggerated, but exaggerated features will also help the audience to indentify the character's key qualities. (CreativeBloq 2013.)

Many games have given their main character an interesting personality, such as Kratos in *God of War* series. The personality can be seen clearly from the character’s emotional expressions, as demonstrated in Image 9. In this series the character’s personality is also expressed the way it has been modeled and textured. Santa Monica Studios has also made a huge impact by facial expression and the character’s whole range of emotions. Kratos is described as an angry and frustrated personality, which can be easily seen in the game.

IMAGE 9. Kratos from God of War series. (Comicvine 2014)

For a game studio it is most important to make the character distinctive. It is a well known fact that there are already hundreds of similar creations, but with a few decisive unique features the studio can make the character famous. For instance the first Tomb Raider game starring Lara Croft came out in 1996 published by Eidos Interactive. The character can be listed in the top 10 most famous game characters and there were also two movies produced about the adventures of Lara Croft. Later this year a new Tomb Raider game is going to be published for Xbox. (Tombraider.com 2015.)

4 RIG

In 3D animation, bones are objects that move in 3D space. The complete skeleton that is bound to the mesh is called a rig. They have a certain user defined influence on the vertices of the character. Bones are built in a series, so that they form a hierarchy. The vertices of the 3D mesh are weighted unto each bone, so that each bone has a different percentile of influence of each vertex. The 3D mesh acts like human skin and bones control the movement. The bones are used to create a complete skeleton with the desired bone structure. (Slick 2015.)

Animators use the rig to bend and twist the character into a desired pose. A pose represents how the character is positioned. It can be described the same way as a statue is posed. Depending on how the rig is used, it can be very simple or very complex. Facial rigging is often separate from the main motion controls. For facial rigging the traditional joint/bone structure is inefficient and very difficult to use. Animators may prefer to use morph targets or blend shapes, as they are often a more efficient solution.

Placement of the skeleton is the easiest part of the rigging process. The joints are placed exactly where they would be in real life. The hardest part is to make the planning for the motion. It is better to build the rig to be flexible; the more motion it can perform the better the animation will be. Knowing the main poses can help when creating the rig, to prevent it from breaking. An example of a character rig can be seen in Image 10. (Ehrenhaus 2014.)

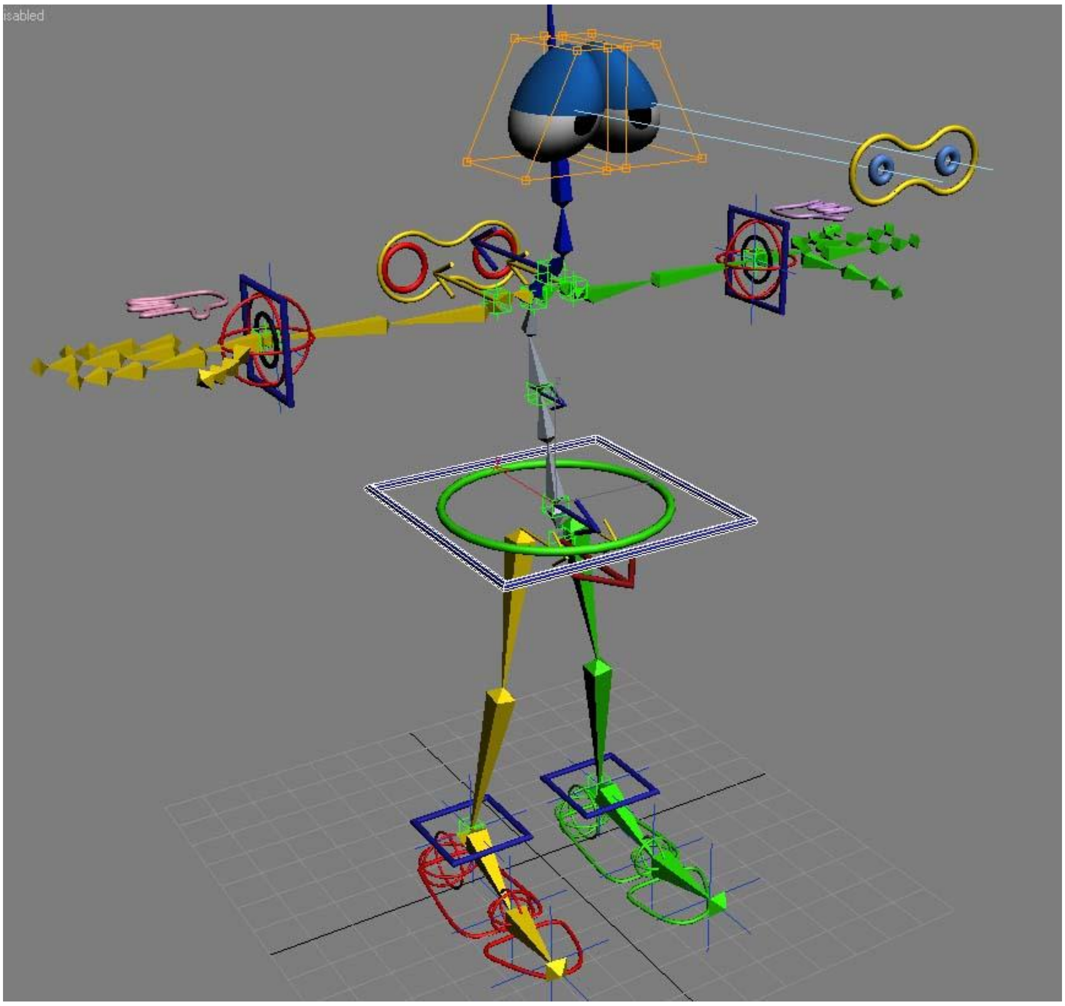

Image 10. One example of a character rig. (Treddi 2014)

4.1 Forward kinematics

When animating with forward kinematics (FK), each joint and its rotational position is specified. Basically this means that each movement needs to be very carefully planned to get good looking results. Forward kinematics works with linked objects with the principle that moving a root bone also moves all of its children. Children in this case means bone objects that have a parent in a hierarchy. Animating a human arm with forward kinematics animation is started from the shoulder, which affects on the elbow, wrist and hand. FK animation has some advantages compared to inverse kinematics (IK, see chapter 4.2), for example when animating

Kraken’s tentacle (Autodesk 2011). In Image 11 it is demonstrated that with IK it would be impossible to create the tentacle, as it would penetrate itself, whereas with FK it is fairly easy. This way the tentacle is animated by rotating each of the joints starting from the chain root that connects to the body of the Kraken. (Bousquet 2006, 8-9.)

Forward kinematics is basically used if something needs to be controlled from top to bottom. It is useful when animating a character’s fingers, but it is also very time consuming. Forward kinematics may be time consuming and it needs thorough planning, but there are advantages, which is why forward kinematics is very popular when making basic animations. Adding forward kinematics chain in a hierarchy of 3D meshes will automaticly result in a linked system of bones, so nothing else needs to be done in order to use FK.

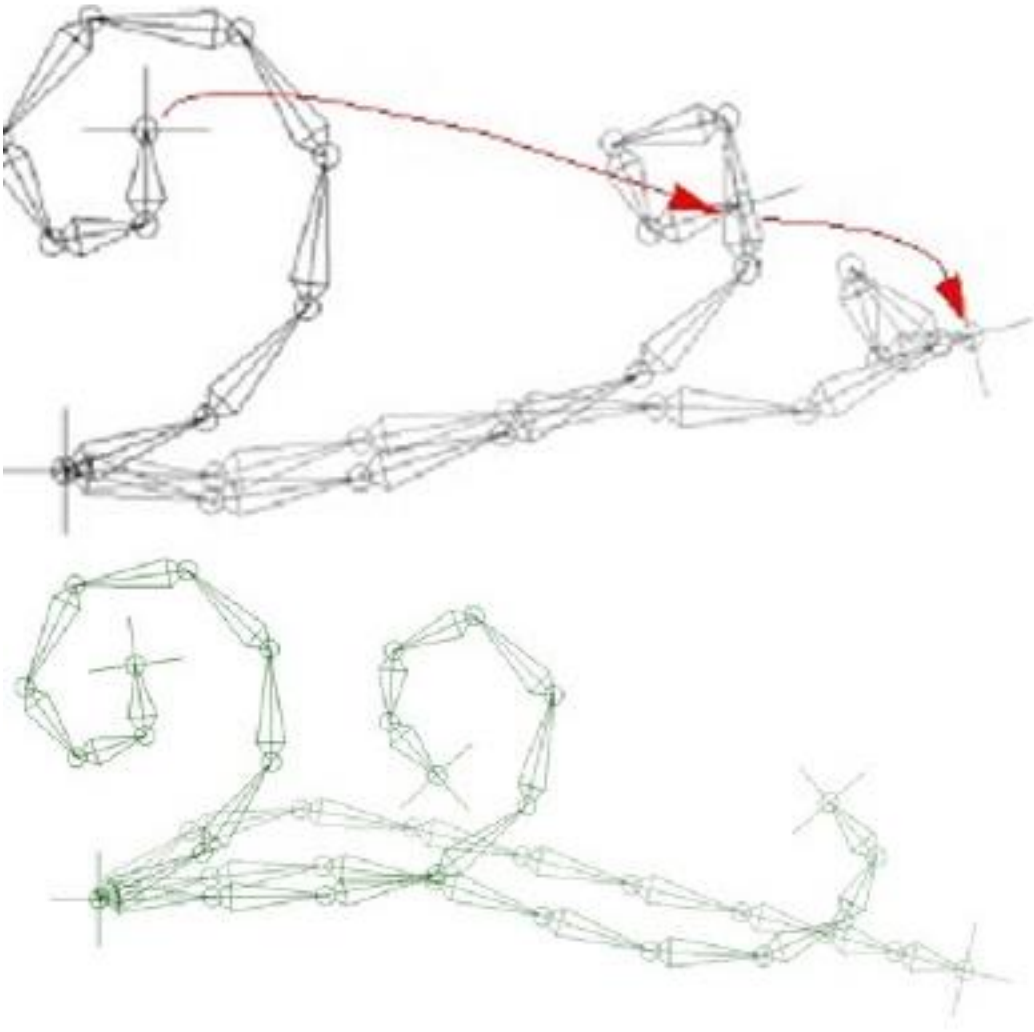

IMAGE 11. On top the tentacle with IK, below with FK. (Autodesk 2011)

4.2 Inverse kinematics

Inverse kinematics, also known as IK, is the reverse process of forward kinematics. For example a human hand is controlled by the hand, not the shoulder. By moving the hand, the joints above it on the hierarchy are automatically interpolated by the software (Slick 2015.). It is often used for rigging a character’s arms and legs, as it is much easier to calculate the distance by using the hand of the character than start moving joints one by one from the shoulder. In Image 12, the character’s legs are using an IK chain. The character’s hip has been lowered, and the control point is located under the foot, so the knees are bending as a result. Generally this saves time as there is no need to make the animation joint-by-joint. (Bousquet 2006, 8-9.)

IMAGE12. How inverse kinematics functions.

Most of the software includes several different IK options. Planning the IK chain is very important, as inverse kinematics needs to be set up by selecting the start and end bones and then enabling the IK. There are many solvers available for the IK chain. Generally artists use history independent solver (HI), as it is the most versatile of the IK solvers. Other solvers can be really useful when animating something specific, for instance a snake. A Snake's rig would be easiest to create with the simplest of the solvers, spline IK.

4.3 CAT

The character animation toolkit has many presets for bone systems. It is a plugin, but since the 2011 version it has already been implemented in Autodesk 3Ds Max. There is a possibility to create your own rig, or modify one of the presets to meet your needs. It contains many presets such as human, ape, dragon and alien. An example of an alien rig in CAT can be seen in Image 13. The controls of the character animation toolkit are flexible and presets are easy to manipulate.

CAT works like normal forward or inverse kinematics. It offers tools to use basic rigging controls, skinning, and options to work with muscle deformation and jiggle effect for the flesh. It is possible to export the animation straight to game engines like Unity.

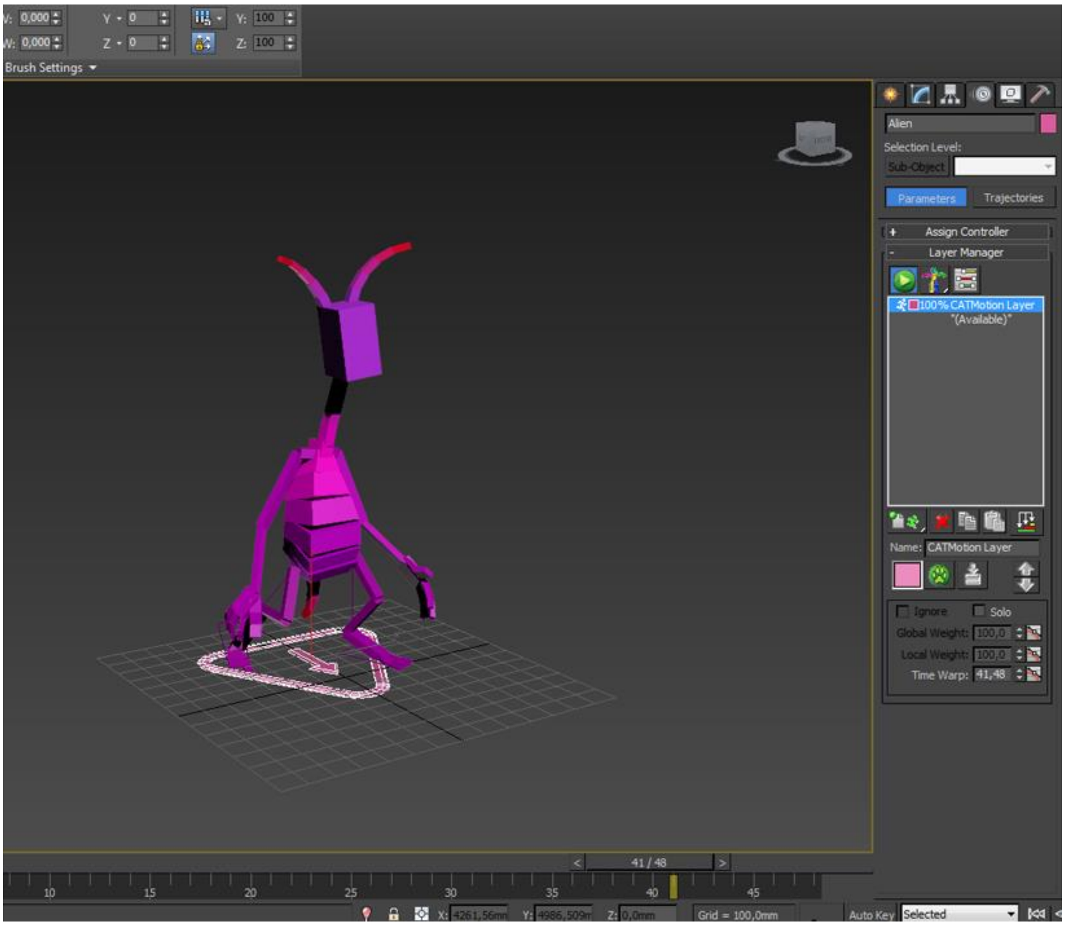

IMAGE 13. Alien rig in CAT.

4.4 Morph Targets

Working with facial expressions or fabrics and models that do not function properly with bone systems, artists often use morph targets. Morph targets use vertex point animation and software interpolation to create frames in a particular animation sequence. A morph target is a duplicate of the original model, but with a different state. The Computer then interpolates the gradual variations between the starting state and the ending state. In many ways it resembles the shape tween animation in Flash. There are many ways to control the morph target. For instance, it is possible to turn down the amount of the morph so a smile would look like a small frown. (Luc-Sanders 2014.)

Morph targets are often used with the speech of the character. Animating speech can be one of the most difficult tasks in the animation process. Character artists often use phoneme shapes to animate the movement of the mouth. Then the specific morph targets are timed with the audio track to create a natural-looking mouth movement. There are twelve different morph targets shown in Image 14.

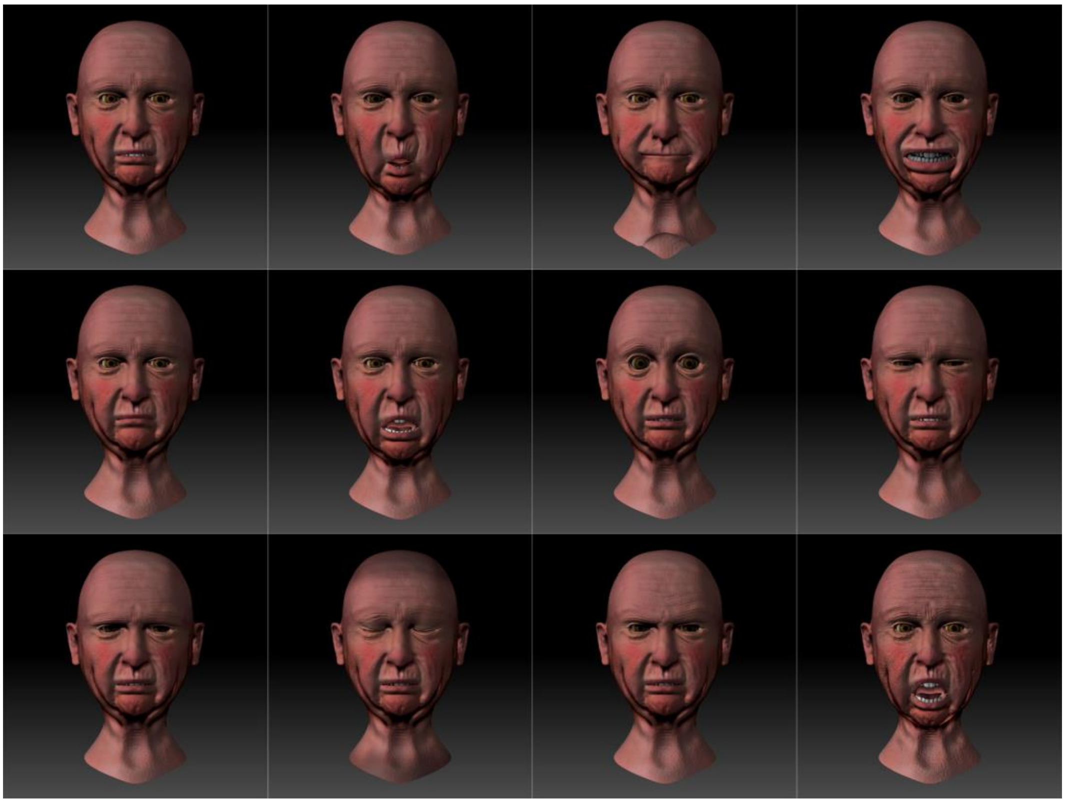

Image 14: A collection of morph targets. (Aoostergetel 2010)

The problem with morph targets is that artists are forced to make manual manipulation for various vertex points to have different states for the animation. This is extremely time-consuming and requires some modifications after the interpolation between different states. As morph targets need separate models, bone systems work with the existing model. (Luc-Sanders 2013.)

4.5 Controls

Artists use more and more controllers called rig nodes to control certain parts of a rig. When the character has ready rig controls, the animation will be much easier to work with. A good rig has more controls than it would necessarily need to get the character moving. Animators want to use simple control systems to affect the whole main body of the character. The extra controls are often used to exaggerate the motions. A good example of a facial rig controller can be seen in Image 15.

Animations use muscle controls that affect the skinning of the mesh. The skinning is considered to be fairly easy with low poly characters as there are fewer vertices to cause problems with the model. Certain areas need to appear flexible and fleshy, so artists use muscle/volume controls to stretch some areas while other areas are shrinking. A good example of areas that need muscle controls is human arms and face.

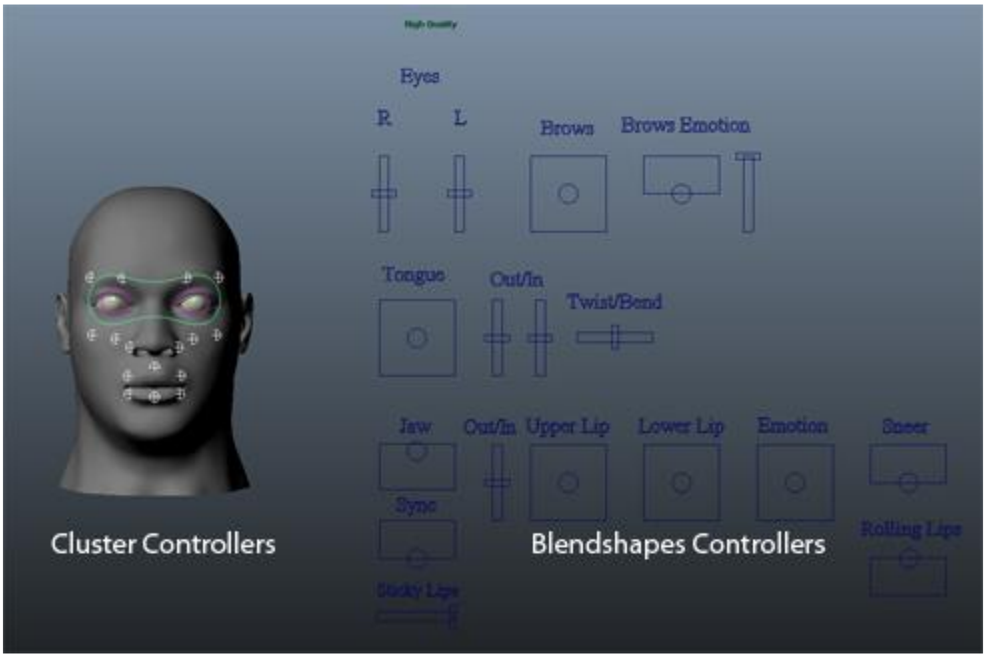

IMAGE 15. Example of a facial rig. (Ahlborg 2010)

5 EMOTIONAL ANIMATION

People have universal emotional expressions that are used to show a certain emotion. The most common are disgust, anger, fear, sadness, happiness and surprise. These six can be called microexpressions, which can occur really fast, as fast as 1/15 or 1/25 of a second. Posture and gesture can have a decisive effect on the character's expressions. Six microexpressions can be seen in Image 16. People have been communicating using the same facial expressions throughout the entire human evolution. The human skull is what makes our heads different, but still all skulls are basically the same. The proportions of the skull are nearly always the same. (Faigin 1990, 14-15.)

> *Think of the face as being like the key solo instrument in a symphony orchestra. In a concerto, the soloist can carry the melody, as can the full orchestra along with the soloist.*
>
> *–Gary Faigin*

Image 16. Six microexpressions. (Hillwoltan 2012)

In animation, facial expressions are a way to make the character feel more like a real person. Animators needs to learn about muscles of the face and how they interact with one another. The human face is composed of lots of little muscles that surround each eye and radiate out from the mouth. To create correct bone influences for the animation rig, an animator needs to know where the muscles originate, and where they are inserted. (Maraffi 2003.)

Emotional animation is often used together with body language to boost the effect of an emotion. Facial expressions should be enough to convey the character’s state of mind, and should work well enough to communicate through pantomime i.e. without sound. This is why artists tend to exaggerate the emotions. Even though the facial expression might be exaggerated, the characters also tend to express the same emotion through body language. Artists should also highlight the movement of the body, for example when feeling pain or caught by surprise. Body language is a way to express the weight or force. The technique used to represent the force in animation is called the line of action. Examples of line of action can be seen in Image 17. Line of action can also be used in facial expressions to give momentum for the emotional expression.

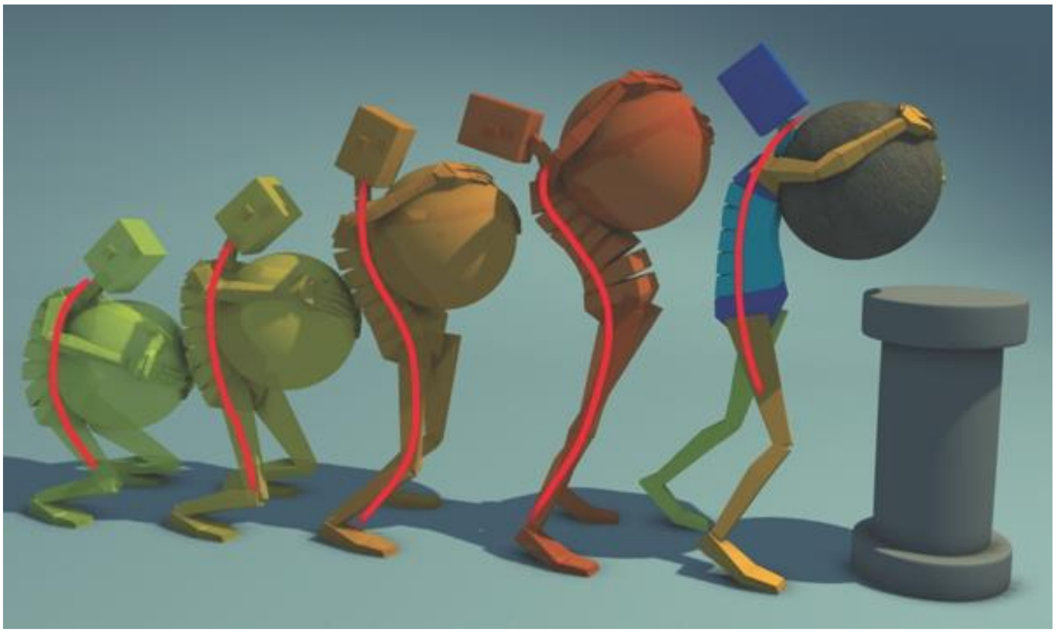

Image 17: Line of Action. (Mori 2014)

5.1 Facial animation

Just like in real life, communication involves both verbal and nonverbal forms to make sure our message is heard. From birth to death our face links us to our friends and family. It is important that we understand these subtle signals as a larger part of the communication process. This is why we rarely misread people closest to us. (Faigin 1990.) In games facial expressions express the mood of the scene and the character so the players can feel the immersion of the game world.

Facial animation can be considered the hardest part of creating the illusion of a real-life character. People are so familiar with the human face and they have been reading microexpressions their entire lifetime, so basic expressions need to appear realistic. In games it is also quite hard to read small and subtle movements from afar. This is why facial expressions and body language is often exaggerated.

Game characters from certain series have become very much alive for the players and the game studios are trying to implement as realistic animations as they possibly can in their games. Some of the best-liked facial animations in games have been in L.A. Noire, Half Life 2 and God of War. Among others these game studios have taken the facial animation process from the film industry and are using motion capture to record the facial movement as closely as possible.

5.2 Lip sync

Lip synchronization plays a huge part in a believable facial animation. If people see a character speaking, they expect to see normal movement of the lips. In animation, exaggeration should be used everywhere, even in lip synchronization. Lip sync often happens very quickly, for example players might see the talking character just for a few seconds. Players need the exaggeration to happen, so they can read the facial expressions properly.

For an animator it is crucial to realize that not every syllable needs to be animated. Often it is enough to create a start and end pose for a word and use a blender over the middle-part. If every syllable were animated the mouth would be closing more than it needs to. It is also important to use different poses when the character is whispering or when it is shouting. The best way to catch realistic looking lip synchronization is to film yourself to get reference material.

When there is too much offset going on in an animation, it looks very disturbing. However, if there is no offset and the audio starts playing just as the jaw is opening, it will feel like the animations are slightly ahead of the audio sound. Offsetting becomes important when using the closed mouth shapes, such as B – M or P. The important thing about closed mouth shapes is that they cannot be blended. The mouth needs to close to seem realistic and players can read the expression properly. Today many game studios use motion capture to animate realistic looking lip sync. (Wikibooks 2014.)

5.3 Motion capture

Motion capture has also come a long way since its early stages and it is now used with almost every video game. Motion capture means recording the movements of real people or animals and mapping the data onto the character. (Masters 2014d.) It is a powerful tool to add realism on the cutscenes and gameplay. For game development however, motion capture is often too realistic, as games are a way to escape from reality.

Motion capture with games nearly always needs to be tweaked afterwards. The character will move just as the actor moved, and for most of the games, artists need to change the poses to get the best performance. Often animators use animation where both motion capture and keyframe animation is combined, which is commonly referred to as hybrid animation. (Masters 2014d.)

To use motion capture with games requires precise planning and directing. A good example of a game that succeeded in using believable motion capture is L.A. Noire (see Image 18). The motion capture process itself is very expensive and time consuming so the key question when creating animations for a game is, whether it suits the visual style, game engine and, most of all, budget and schedule. By watching and learning from what other game studios have done, it can be seen that the actor and game character resemble each other quite a lot. This makes the directing much easier. (Gamasutra 2000.)

The technology is taking huge leaps right now, and there are solutions for smaller studios to start using motion capture as a tool to add realism in their games. For instance software like Face Plus only needs a webcam to get started. There are also real-time capture technologies, which will some day be used in games, but for now they offer intriguing possibilities for TV production only. (CreativeBloq 2015.)

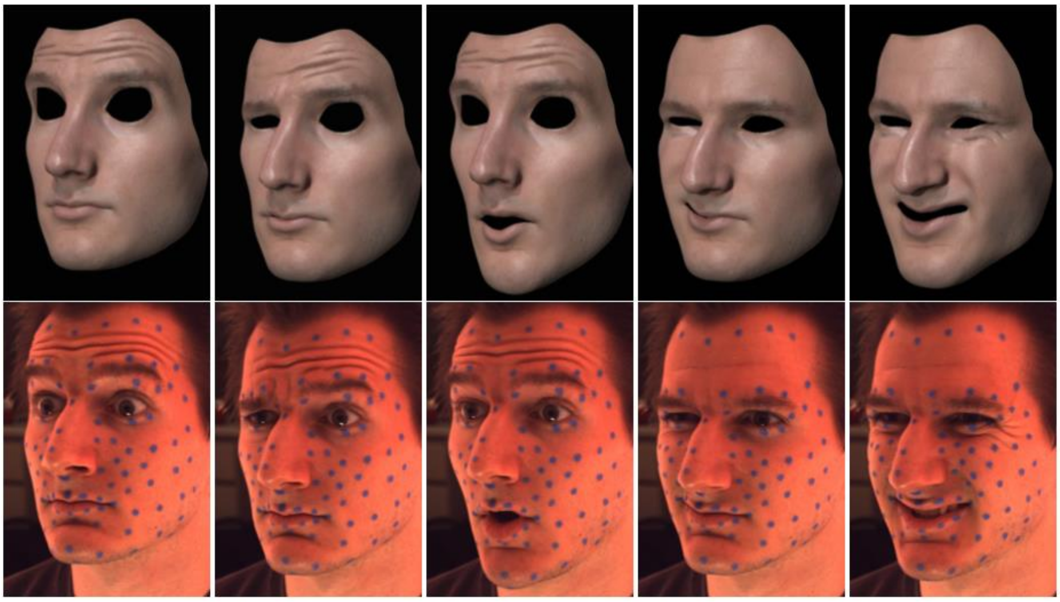

IMAGE 18. Motion capture for a character. (Bielefeld University 2014)

6 CASE

This case is focusing on animating the human face with keyframe animation. The techniques mentioned earlier in the thesis will be used for this model to investigate subtle changes in facial expressions. One objective is also to prove the key points of the structure of the model and texturing choices used for this particular 3D model.

Modeling and animation of the objects will be done in 3Ds Max 2015 and Mudbox 2015. The texturing will be done in Substance Painter. The bone system and skinning will also be created in 3Ds Max, as it offers all the needed tools to cover the basics of facial animation for game characters.

6.1 Planning

During the project the significance of studying the basics became clear. The first part of studying the facial animations was the muscles of the human face. A book called *Artist's complete guide to facial expression* by Gary Faigin was very informative and clear, showing how and why the expressions should look the way they do. Reference videos played a huge part in the studying process. The easiest way of getting reference is to act the expressions yourself, but it can be useful to gather more reference for variation. The planning can contain a lot of searching for reference 3D models. An example of a reference model can be seen in Image 19.

After studying how the muscles move, it was time to start planning the animation. During the planning phase, an exposure sheet was used to time the animations correctly. An exposure sheet is a paper where the animator can see the instructions and timings of the animation. This saves time especially if the animator is someone else than the person who is making the exposure sheet.

The sex of the character is also very important. Males and females have the same facial expressions but there are features that make a female face look more feminine. This also has an effect on the topology of the model of the character. Male characters have more square features in the outlines where female characters have more rounded curves. There are also differences in the nose ridge and the eyes as a male skull has a more rectangular base for the eyes.

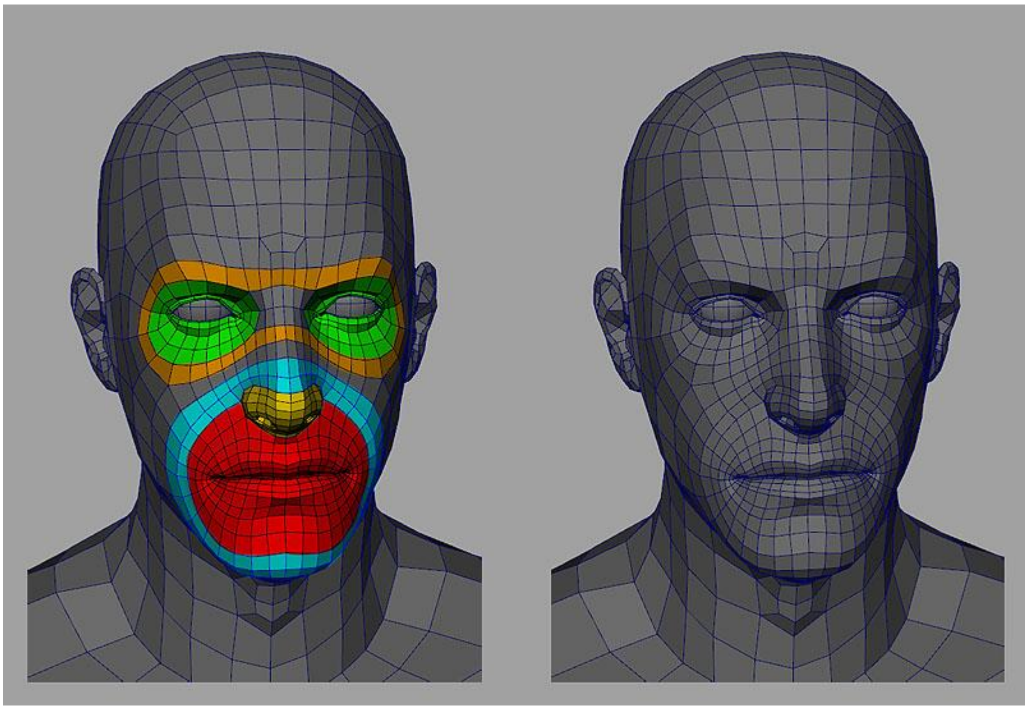

IMAGE 19. Example of a planning reference. (Tom Parker 2011)

6.2 Modeling

The basemesh for this project was made by Digitaltutors. For this project it was manipulated first in Mudbox to create the highpoly version, to get all the needed texture maps. The design itself is very simple, the head consists of five separate 3D meshes: The head, left and right eye, and lower teeth and upper teeth. For modeling the detail for the surface a stencil was used. A stencil is a black and white image, similar to heightmap, which allows artists to quickly create detail to the surface.The highpoly model itself is not very detailed; it has some guidelines for the hairflow and eyebrows and a little displacement in the skin area. The idea to keep the high poly fairly simple was a choice that was made, as the texturing will be done in Substance Painter, which offers tools to create more height-mapping.

IMAGE 20. The highpoly version of the face.

The topology follows around the natural muscle lines, to keep the topology clean for deformation during the animation. The model is made from quads, which means polygons that consist of four sides. It will ensure cleaner topology, and it also looks cleaner, so it is better considering possible teamwork with a model. Some parts cause difficulties with the edgeflow, which is why some parts of the model have a triangle instead of a quad. In the end, this model is going to be used in a Unity project, which means that the quads will be converted into triangles anyway. This particular model is designed to be used in a PC game as the polycount of the low poly version is fairly high.

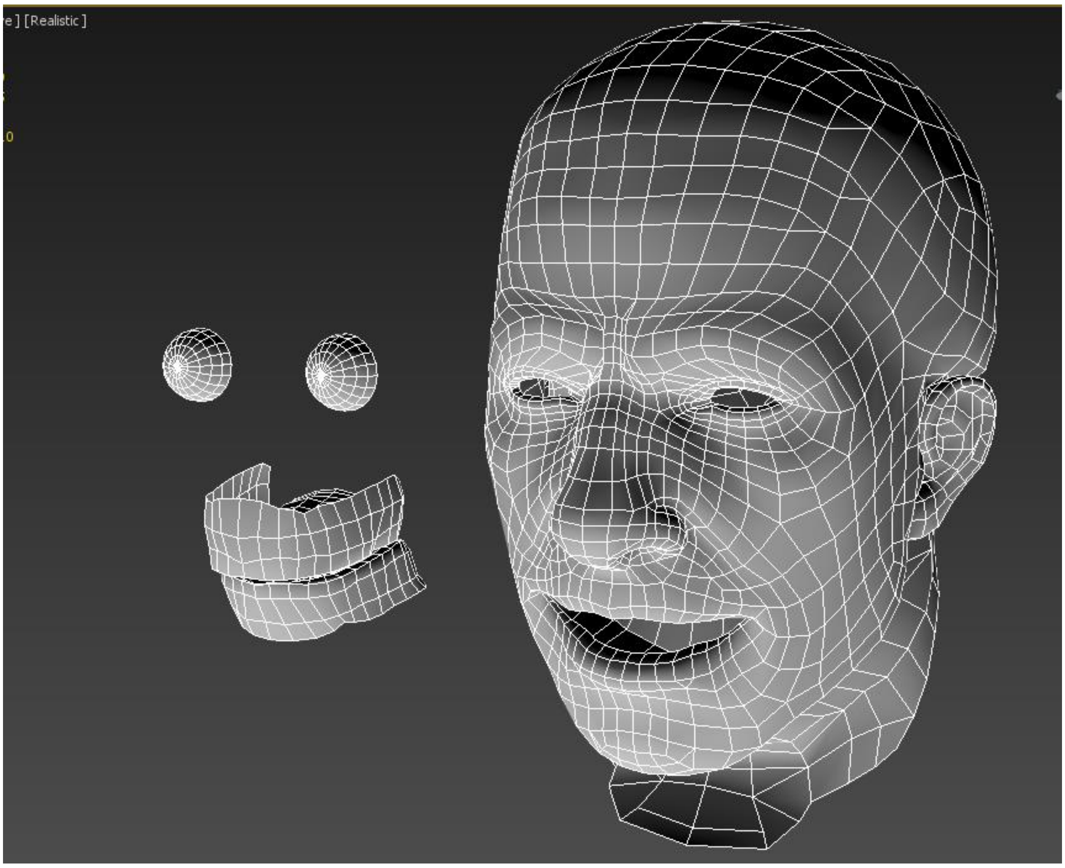

IMAGE 21. The topology of the case model.

6.2.1 UVW mapping

This project includes a part where the texture coordinate map is created. The UVW mapping is done with the unwrap UVW modifier in 3Ds Max. Mapping can be one of the most difficult parts of the 3D modeling process. It is considered a tedious part of creating textures, but it is also very necessary. The unwrapping part with organic models is the most probable phase to cause problems. The biggest issues tend to happen with the holes where eyes should be. Image 22 shows that the right part of the face has texture coordinates, but the left one does not. This is because the face is symmetrical and the UVW map of the left side was mirrored on top of the right one to save space on the map.

This was later found useless saving as the project does not include any more objects that would need mapping, so the UVW map was mirrored back to its original space.

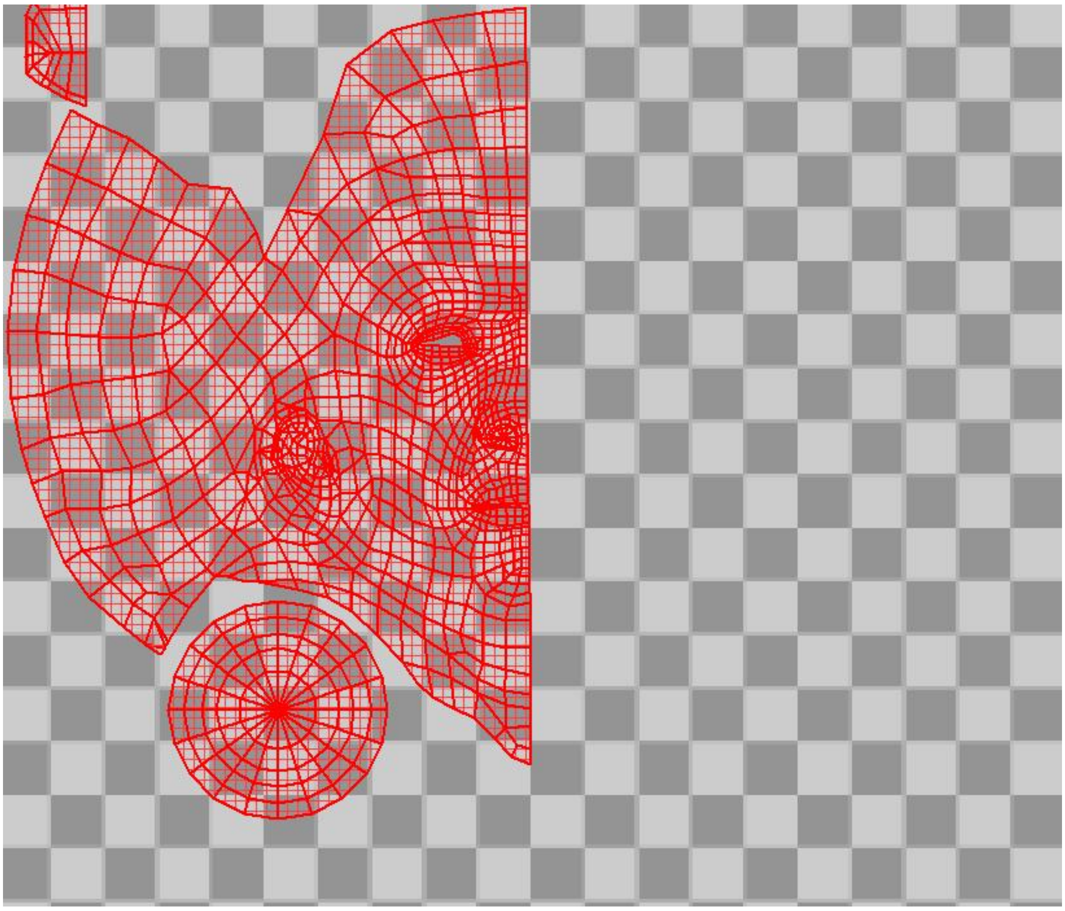

IMAGE 22. UVW map of the model.

6.2.2 Texturing

This part consists of extracting texture maps from Mudbox and 3D painting in Substance Painter. With Mudbox 2015 by Autodesk, it is possible to extract texture maps for your model. In this project normal maps and ambient occlusion maps were extracted. Normal mapping in Mudbox works between different subdivision levels. A new subdivision level will increase the amount of polygons in the scene but often three or more subdivision levels are needed to create sufficient geometry for fluid digital sculpting. Mudbox then calculates the height information variation between these two subdivision levels and creates a chosen texture map. An example of options during extraction can be seen in Image 23 below.

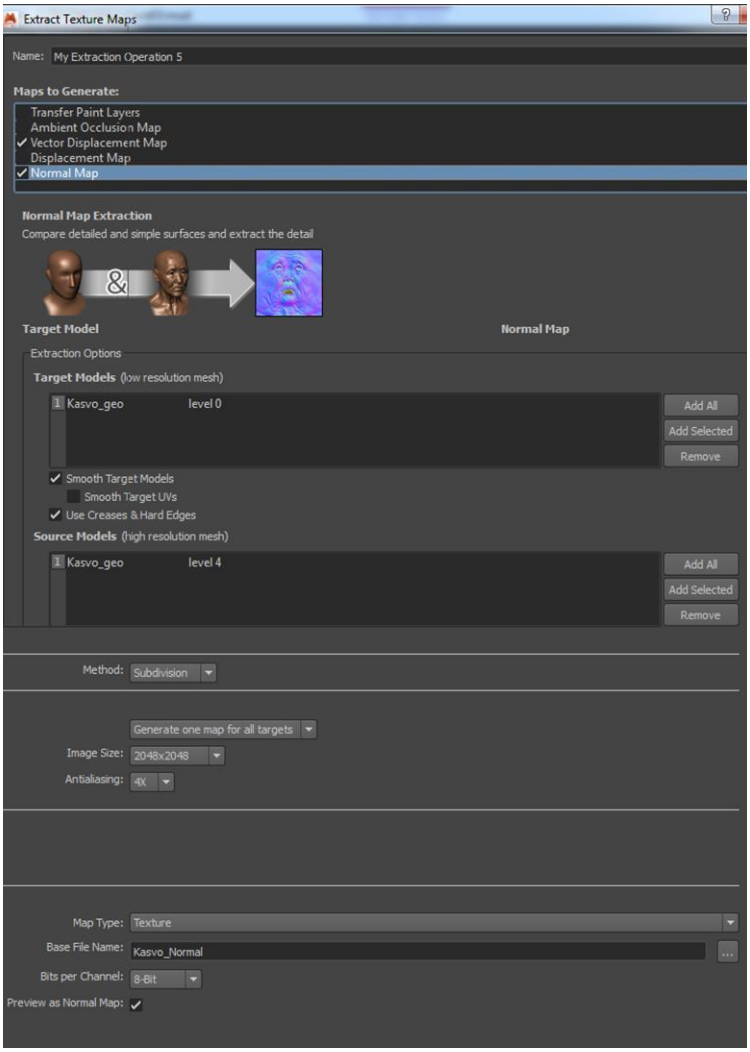

IMAGE 23. Extracting texture maps in Mudbox.

The 3D painting was done in Substance Painter, as it supports physically based shading, which works with the Unity 5 game engine. Physically based shading means that simultaneously at least four different texture channels are used. The default settings use channels for base color (diffuse), height, roughness and metallic. Each of these channels represents a very important part when trying to create believable materials for a model.

This model's diffuse channel contains the skin's base color and variation in the skin tone. The ambient occlusion map has been rendered as a separate map, as the new Unity shader accepts an occlusion map. The diffuse channel also contains hair color and color variation. The height channel will slightly increase the bumps achieved with the normal map. The original normal map was baked from Mudbox and imported into Substance Painter to make use of the final surface detail using the physically based rendering. The normal map will be used as it came out from Mudbox.

The most significant part of this texturing process was to paint the roughness map. This map will control how the model is going to react when light hits it. Most of the physically based shaders are based on the ray-tracing algorithm (Pharr 2010.). This is based on following the path of a ray of light through the scene. The Computer then calculates how it will interact and bounce off objects in the environment. In Substance Painter this will be controlled with the roughness- and metallic channels. Unity 5 will use this map in the specular map slot, to interact with the scene lighting.

6.3 Bone structure

During this project, many tutorials and different models with facial bones were studied for advice. A simple bone structure was used to stay in schedule. The facial bone system was taken from the tutorial made by Digitaltutors, as is seemed to be a good fit for a game character.

The basic structure is that there is a root bone that controls the entire head. This bone is found in the neck area and it is connected to the jaw bone and the head bone. These three bones control the major movements of the head. The bones are manipulated by rectangles around the character's head to simplify the controls of the animation. Each eye area consists of four bones, one for the brow movement, two for eyelids and one for the eyeball. The bone for the eyeball has a look-at constraint to keep it targeted at the controller for the eyeballs. This helps when animating the movement for the eyes. The bone structure for this model can be seen in Image 24.

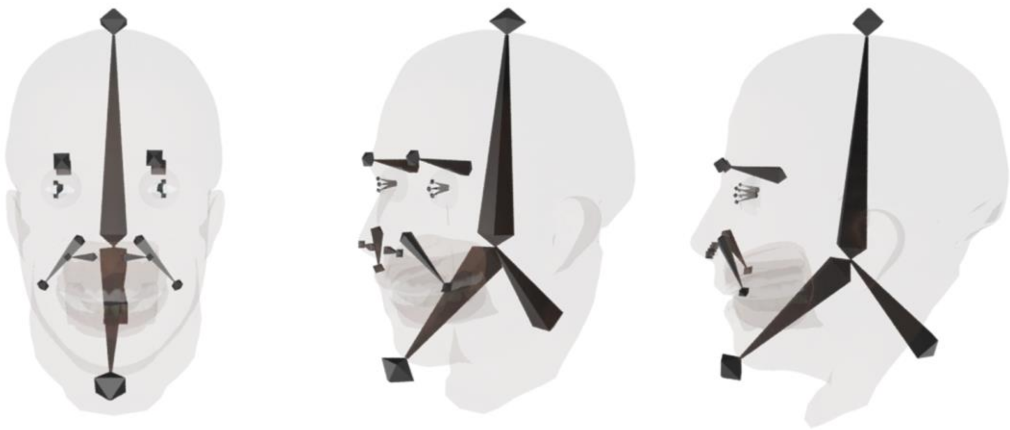

IMAGE 24. The rig.

The mouth area is controlled with twelve points that cause deformation in the mouth area and the major jaw bone. The twelve points are used to deform the mouth into the correct shape. Skinning defines how the bones are weighted onto the vertices. In 3Ds Max there is a modifier called skin, which enables the user to select bones and weight them onto the selected vertices. Typically, several bones affect the same area with different weighting. In this project the eyes are fully weighted to the eye bones, upper teeth to the head bone and the lower teeth to the jaw bone. For the rest of the face, the paint weights tool was used. This tool allows the user to paint the amount of weight for the skin envelopes. Envelopes are acting as containers for the weighting information for the vertices. An example of a weighted jaw bone and its envelope can be seen in Image 25.

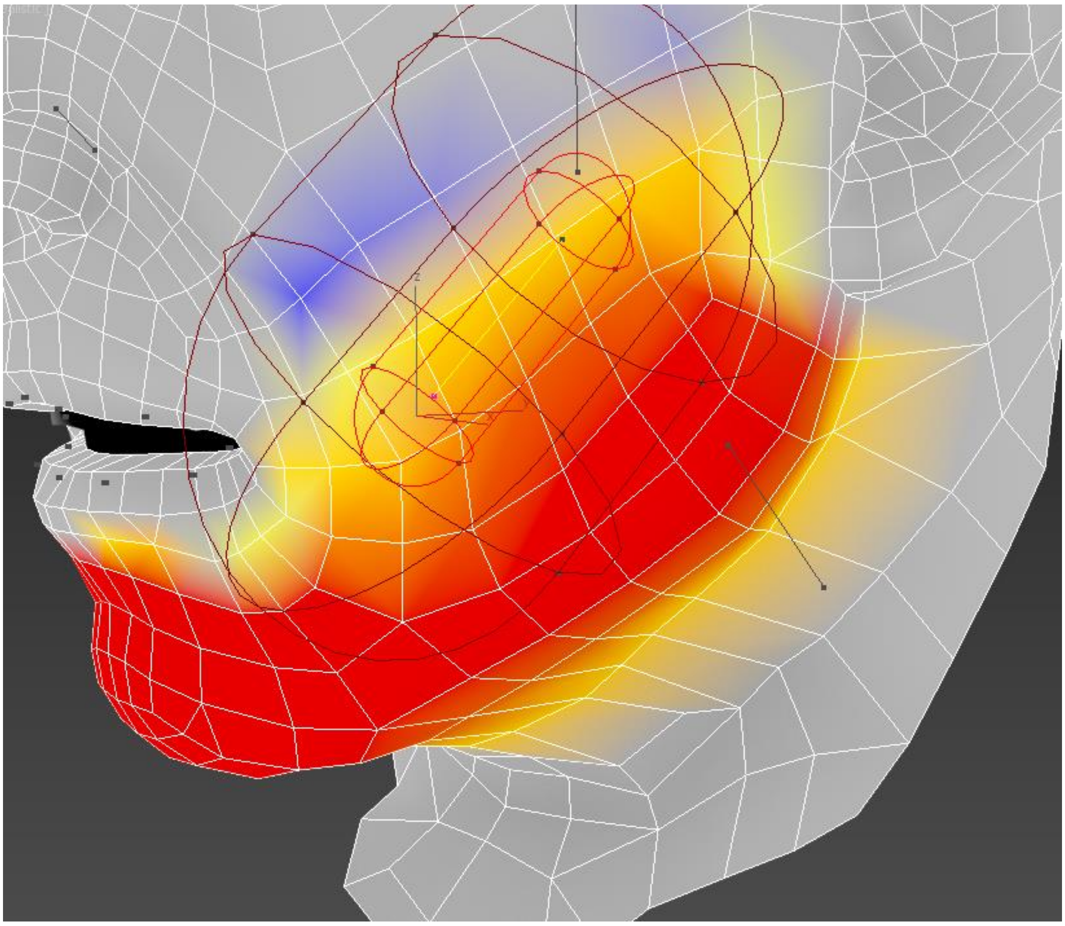

IMAGE 25. Weight area and envelope in 3Ds Max.

6.4 Animations

References for facial expressions can be found in *Artists Complete Guide to Facial Expressions* by Gary Faigin. This book gave a total understanding how and why facial muscles react to movement in each microexpression. The animation started from the pose that was made during the modeling and sculpting phases. Keyframe animation proved to be good for this project as the schedule was rather tight. The biggest problems were encountered with the anger expression. The cause of the problems was the lack of wrinkles when eyebrows start to move inwards.

This could have been avoided by using wrinkle maps or morph targets instead of basic keyframe animation.

Regarding the animations, it was found that more controls for the mouth area could have been used for smoother curves in the lip area. Small movements for the head, like nodding or shaking the head, work really well with this type of rig. Also targeting the eyes is really easy with the help of the look-at constraint. The most complex area is the mouth, which could have made use of additional controls and morph targets to achieve more accurate results. The expressions work well when observing the character from a medium distance, but close-up animations would need more detail. Textured 3D model can be seen in Image 26.

IMAGE 26. Textured model.

7 CONCLUSION

Faciall animation is a rapidly evolving feature in gaming. In every new game more and more animations are implemented to give players the tools to express themselves inside the game. Good animations get more and more important with online games, where people interact with each other.

There are numerous plugins and utility programs to use. Some of these programs are essential for game studios to achieve great results with tight schedules. Software providers are developing new tools for game studios to use, to achieve stunningly realistic animations.

Creating believable facial animations for a character is an interesting and challenging part of the game creation process. The subtle changes in the human face can make it very difficult to express emotions through animation. Creating a good animation is not quite enough, as there are many fields such as texturing that influence in the final outcome greatly. Game studios have introduced many tools from the film industry to achieve more realistic results in their own games.

During the case section it became clear how many different skills are needed to produce a complete animation. To optimize the polycount and achieve good animation readiness is a new artform and requires many specific skills. The animations for the case model proved to be more time consuming than originally planned for the thesis. For this reason the animations were carried out as keyframe animations only. The microexpressions were quite enjoyable to animate.

PRINTED REFERENCES

Bousquet, M. & McCarthy, M. 2006. 3Ds Max Animation with Biped, USA: New Riders.

Chang, C. 2006 Modeling, UV Mapping, and Texturing 3D Game Weapons, USA: Wordware Publishing, Inc.

Faigin, G. 1990. Facial Expressions, USA: Watson-Guptill Publications Inc.

Maraffi, C. 2003. Maya Character Creation, USA: New Riders.

Murdock, K. 2015 Autodesks 3Ds max 2015 complete reference guide, USA: SDC Publications.

Pharr, M. 2010. Physically Based Rendering: From Theory to Implementation, USA: Elsevier, Morgan Kaufmann, 2nd edition.

ONLINE REFERENCES

Autodesk softImage. 2011. Animating with forward kinematics [referenced 12th February 2015]. Available at: http://softimage.wiki.softimage.com/xsidocs/ik_AnimatingwithForwardKinematics.htm

CreativeBloq. 2013. Character design tips [referenced 5th February 2015] Availabe at: http://www.creativebloq.com/character-design/tips-5132643 (20

CreativeBloq. 2015. Why motion capture is set to radically change in 2015 [referenced 5th March 2015]. Available at: http://www.creativebloq.com/3d/why-motion-capture-radically-change-2015-121413749

Ehrenhaus, S. 2014. Rigging Guideline for the Artist [referenced 1st March 2015]. Available at: http://blog.digitaltutors.com/rigging-guideline-artist-whats-important-good-rig

Gamasutra. 2000. [referenced 5th March 2015]. Available at: http://www.gamasutra.com/view/feature/131827/planning_and_directing_motion_.php?

Landgraf, H. 2012. The increasing role of character animation in video games [referenced 22nd January 2015]. Available at: http://www.animationarena.com/character-animation.html

Luc- Sanders, A. What are Morph Targets? [referenced 18th February 2015]. Available at: http://animation.about.com/od/glossaryofterms/g/What-Are-Morph-Targets.htm

Masters, M. 2014a. From the 80s to Now: The Evolution of Animation in Video Games [referenced 10th January 2015]. Available at: http://blog.digitaltutors.com/80s-now-evolution-animation-video-games/

Masters, M. 2014b. How Animation for Games Is Different from Animation for Movies [referenced 13th January 2015]. Available at: http://blog.digitaltutors.com/how-animation-for-games-is-different-from-animation-for-movies/

Masters, M. 2014c. Texturing for Games – Maintain High Level of Detail without Extra Geometry [referenced 2nd February 2015]. Available at: http://blog.digitaltutors.com/texturing-games-maintain-high-level-detail-without-extra-geometry/

Masters, M. 2014d. The Rise of Motion Capture and What it Means for You [referenced 22nd February 2015]. Available at: http://blog.digitaltutors.com/rise-motion-capture-means/

Masters, M. 2015. The Ultimate Guide to Preparin Your Students for the Game Dev Industry [referenced 15th January 2015]. Available at: http://blog.digitaltutors.com/preparing-your-students-for-the-game-dev-industry/

Pitzel, S. 2011 Character animation: Skeletons and Ionverse Kinematics [referenced 1st March 2015] Available at: https://software.intel.com/en-us/articles/character-animation-skeletons-and-inverse-kinematics

Polycount Wiki [referenced 1st March 2015] Available at: http://wiki.polycount.com/

Slick, J. 2015. What is Rigging [referenced 23rd February 2015]. Available at: http://3d.about.com/od/Creating-3D-The-CG-Pipeline/a/What-Is-Rigging.htm

Unity Documentation. 2015. [referenced 28th January 2015]. Available at: http://docs.unity3d.com/Manual/ModelingOptimizedCharacters.html 2014

Ward, A. 2011. How to create character models for games [referenced 21st January 2015]. Available at: http://www.creativebloq.com/how-create-character-models-games-18-top-tips-9113050

WikiBooks. 2014. [referenced 17th March 2015]. Available at: http://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/Advanced_Tutorials/Advanced_Animation/Guided_tour/Mesh/Shape/Sync

Williamson J. 2012. Why Topology Matters in Modeling [referenced 28th January 2015]. Available at:http://cgcookie.com/blender/2012/11/28/why-topology-matters-modeling/

WiseGeek. 2014. What is 3d modeling [referenced 20th January 2015]. Available at: http://www.wisegeek.com/what-is-3d-modeling.htm

E-MAILS

Logue, M. 2015. Re: Bachelor’s degree [e-mail]. Recipient Wallin, K. Sent 3rd March 2015.

IMAGES

1. Usgamer. 2015. Lara Croft [referenced 20th January 2015]. Available at: http://www.usgamer.com
2. Kalle Wallin
3. Blenderartists. 2014. Head topology [referenced 23th January 2015]. Available at: http://blenderartists.org/forum/attachment.php?attachmentid=265136&d=1381997025
4. Turbosquid. 2013. 3D knight [referenced 23th January 2015]. Available at: http://www.turbosquid.com/3d-models/medieval-heraldic-knight-helmet-3d-model/755558
5. Kalle Wallin
6. Thecreativechris. 2014. Unity 5 standard shader [referenced 2nd February 2015]. available at: https://thecreativechris.wordpress.com/2014/10/01/rd-with-new-unity-5-graphics/
7. Kalle Wallin
8. Davidmlally. 2013. Specular map [referenced 8th February 2015]. Available at: Davidmlally.com/wip-wanderer.redux/
9. Kratos. 2014. Kratos from God of War series [referenced 10th February 2015]. Available at: http://static.comicvine.com/uploads/original/11/116475/4173477-kratos.png
10. Treddi. 2014. Example of a rig [referenced 18th February 2015]. Available at: http://www.treddi.com/upload/cecofuli/Upload/Animazione%20-%20Character%20Rigging/Rigging.jpg
11. Autodesk. 2011. FK and IK [referenced 10th February 2015]. Available at: http://www.softimage.wiki.softimage.com
12. Kalle Wallin
13. Kalle Wallin
14. Aoostergetel. 2010. Morph targets [referenced 28th February 2015]. Available at: http://aoostergetel.blogspot.fi/2010_06_01_archive.html

15. Ahlborg, J. 2010. Rig controls [referenced 19th February 2015]. Available at: http://gscept.com/index.php/sp2010/C29/
16. Hillwoltan. 2012. Microexpressions [referenced 1st March 2015]. Available at: https://hillwoltan.files.wordpress.com/2012/05/microexpressions.jpg
17. Mori, L. 2014. Line of action [referenced 25th March 2015] available at: http://www.3dartistonline.com/news/2014/02/how-do-i-animate-a-character-lifting-weights/
18. Bielefeld University. 2014. Motion capture [referenced 18th March 2015]. Available at: http://graphics.uni-bielefeld.de/research/faces/real-time.png
19. Parker, T. 2011. Reference topology [referenced 8th March 2015]. Available at: https://tomparkersartdump.files.wordpress.com/2011/07/topology_breakdown.jpg
20. – 26. Kalle Wallin

APPENDICES