LAHDEN AMMATTIKORKEAKOULU
Lahti University of Applied Sciences

ANIMATED LOW POLY CHARACTERS

LAHTI UNIVERSITY OF APPLIED SCIENCES
Faculty of Technology
Media Technology
Thesis
31 March 2014
Valtteri Iisakki Jolma

Lahti University of Applied Sciences
Degree Programme in Media Technology

JOLMA, VALTTERI: Animated low poly characters

Bachelor's Thesis in visualization engineering, 34 pages

Spring 2014

ABSTRACT

This thesis deals with the creation of animated low poly game characters. The models in question are very low resolution and so are mostly used in mobile games. The goal of the paper is to specify information that is most essential to each process.

The stages under closer scrutiny are modeling, rigging and animation. Each section focuses on providing information needed to create functioning and believeable characters instead of actual mechanics of the process. Studying these was done in Blender, a free 3D modeling package. These studies are based mostly on the author's own experience in the field, and online wiki-sites.

The case in this paper describes the work done on a personal PC game project. The aim is to create a prototype to evaluate future development possibilities. Many of the methods described in the paper are used in this project.

In conclusion one can state, that efficient optimization and a good visual style are critical in creating enjoyable low poly characters for modern mobile hardware. Even relatively small changes can affect overall performance or image quality.

Key words: video game, animation, low poly, character, graphics, rigging, modeling

Lahden ammattikorkeakoulu
Mediatekniikan koulutusohjelma

JOLMA, VALTTERI: Animoidut low poly-hahmot

Mediatekniikan opinnäytetyö, 34 sivua

Kevät 2014

TIIVISTELMÄ

Tämä opinnäytetyö käsittelee peleissä käytettyjen animoitujen low poly-hahmojen luomisprosessia. Kyseessä ovat hyvin pieniresoluutioiset mallit, joita käytetään pääsääntöisesti mobiilipeleissä. Työn tarkoituksena on eritellä tärkeimmät tiedot, joita tarvitaan eri työvaiheissa.

Tarkemmassa tarkastelussa ovat hahmojen mallinnus, riggaus ja animointi. Varsinaisten tekniikoiden sijaan jokaisessa osa-alueessa perehdytään tietoihin, jotka ovat toimivien ja uskottavien hahmojen kannalta tärkeimpiä. Näiden aiheiden tutkiminen suoritettiin Blender mallinnusohjelmassa. Nämä tutkimukset perustuvat lähinnä kirjoittajan omaan kokemukseen alalta, sekä internetin wiki-tietokantoihin.

Opinnäytetyön projektina esitellään henkilökohtaista peliprojektia PC-alustalle. Tarkoituksena on luoda prototyyppi, jonka perusteella voidaan tutkia mahdollisuuksia jatkokehittelyyn. Projektissa käytetään useita työssä tutkittuja metodeja.

Lopputuloksena voidaan todeta, että low poly-hahmon tehokas optimointi ja visuaalinen tyyli ovat tärkeitä nykypäivän mobiilipeleissä niin laitteiston, kuin myös lopputuloksen kannalta. Pienetkin muutokset saattavat vaikuttaa yleiseen suorituskykyyn tai kuvan laatuun.

Asiasanat: tietokonepelit, animaatio, low poly, hahmo, grafiikka, riggaus, mallinnus

CONTENTS

1 INTRODUCTION

Creation of low poly characters began in the mid 1990s. 3d graphics and computer power were both underdeveloped at the time, so this was the only possible course of action. As technology gradually improved, the amount of polygons possible for game characters rose. Towards the end of 2010 characters were able to contain thousands of polygons, and the term ”low poly” only applied to their relation to the high poly models from which normal maps are baked. Even though the term is largely contextual, it is often used when referring to somewhat blocky models with very low polycounts. This is also the definition that is used in this paper.

Along with these technological advances, computer animation also grew more sophisticated. Early workflows based on vertex animation were replaced with skeletal animating systems, and supplemented with the ever growing motion capture technology. While some games would often feature realistic graphics and animations, there were times when this was not feasible. Having this level of visual fidelity throughout many different sets of environments and characters is very expensive to produce. Also, some genres of games display such large numbers of objects and effects on screen that the model and texture resolutions might have to be toned down to keep the framerate at reasonable levels. Lastly, it could be a conscious choice to go for a low poly look in order to achieve a certain artistic style.

This paper aims to present elements important to achieving well functioning and visually pleasing low poly characters. Working with these models requires fundamental knowledge about how 3D modeling works in general, as one will be dealing with relationships between individual polygons. Another goal is to study how common practices in animation transfer into game animation.

2 LOW POLY

With the recent rise of independent developers and mobile gaming, low poly as an art style has witnessed a boost to its popularity. It is a great compromise of performance, production speed and artistic visuals.

2.1 Technical details

Low poly has some distinct technical differences when compared to the mainstream realistic style. The first noticeable attribute is the polycount and textures. The ideal resolution for a character in a modern game ranges from 1500 to 4000 polygons (Unity Documentation 2011). In the case of a hero character in a large and expensive high end game, the polycount can go as high as 7000 or beyond, depending on available processing power. For texture maps, the size usually varies between 1024x1024 and 2048x2048 pixels. Some large or important characters might require even a 4096x4096 map, but these are rarely used due to the considerable memory requirement.

Texture maps function in power of twos (Strong 2007). Because of this, one can extend these maps in the X or Y direction while retaining functional values. The UV maps of a model can be easily translated to for example 1024x2048 maps, by stretching them in the appropriate direction. If more detail is needed, one can use two maps instead of one.

Image 1. Diffuse texture example

The polycount for a low poly character tends to stay at around 1000 triangles, going as low as a few hundred, or even as high as 2000 triangles. This depends on how intricate animations, and/or how many accessories the character will have. Textures are generally as small as possible, ranging from 128x128 to 512x512 pixels. Due to this small size, cramming as much information as possible in them is crucial. This requires skillfully optimized UV unwrapping, an example of which is seen in Image 1. Depending on the artistic direction and target platform, a number of specialized maps often used in games, such as normal, specular and alpha maps, might be left out.

In the example game shown in Image 2, a choice is given if these normal and specular maps should be used or not, while the game in Image 3 does not give such freedom to the player. Seen on the left side in Image 2, these maps bring out

a more polished look to the units, striking a balance between stylized and more realistic graphics.

Image 2. Far away and close up shots of zerg unit, with different settings on both sides.

Image 3. Far away and close up shot of a hero character.

2.2 Visual Style

> *Though commentary, management and markets may have an effect on the aesthetics and culture of games just as they do any other form of media, discovery will remain squarely in the hands of artists and programmers. (Hayward 2005)*

The most prominent visual styles for games tend to be more or less realistic, and the quality of the result depends on the ambition and resources of the studio. With the biggest developers pushing for graphics that are almost indistinguishable from reality, the public has gotten used to high quality graphics. With these trending realistic graphics being financially the safest bet, publishers are less likely to invest in risky artistic approaches. Asian game developers and publishers are the biggest exception to this with Nintendo producing nothing but stylized games aimed at younger audiences. In the west though, experimentation and discovery is prominently in the hands of the independent developers. Still, there has been some development towards more stylized approaches from larger companies, as more experimental games become financial hits. The rise of mobile gaming is an even bigger influence, where low poly and otherwise stylized approaches are both required and an asset. A good example of this is the popular mobile game Clash of Clans, seen in Image 4.

Image 4. Screenshot from the game Clash of Clans (Supercell 2014)

Even though many technical developments are originally made to further the photorealistic aspect of video game graphics, more artistic styles also benefit from this. The increasing variety of tools can be used to create interesting results, or just simply make the game look better. A sophisticated lighting engine can be used to bring life and mood to an otherwise simple, low poly scene.

An important part of low poly as a visual style is the apparent painterly quality of the textures used. This is the main contributor in achieving an artistic feel for the game. Using different colors to bring contrast helps when the player camera is placed at a long distance from the action. This also brings more variety and makes the result more visually interesting. A small detail that comes with photo texturing is not as effective when viewed from afar, and only serves to clutter the texture with unneeded visual noise. In Image 3 the assets are mostly flat colors with painted highlights and shadows to fake form and give the models some detail.

A third popular characteristic of low poly styles is the exaggeration of body parts (Truth Labs 2013). Pushing these to cartoony levels emphasizes the stylization. As a plus, the character's silhouette is vastly affected, which can be used to its advantage. Enlarging a body part significant to the skills of the character, like head for intelligence or big shoulders for carrying massive guns, brings out the role of the character in a visual way. In Image 3, the character's arms and hands are considerably larger than normal. The character's main objective is to use an assortment of hand held weapons to defeat monsters. Bigger hands can be better seen from afar, and thus allow the artists to make large and detailed weapons.

3 CREATION

When constructing a low poly model and its rig for animation, one has to be aware of certain important stages of the process. Depending on the platform, different restrictions are imposed. PCs and consoles are powerful enough to allow more detailed models and complex rigs. Yet they too benefit from using as few resources as possible. With mobile devices being considerably less capable, there is not as much leeway.

3.1 Polycount

Polycount is the defining feature of any model, thus early on it is advantageous to think about how to divide the available polygons throughout it. Cylindrical parts of the mesh, such as arms and legs, often tend to regulate the overall resolution. These parts will be given as few edges along the circumference as possible, while still achieving the desired silhouette. The amount of vertices at a joint where an appendage connects to the main body influences how many edges the latter will contain.

It is possible to alter the edge count between parts of the body by combining two or more edgeloops into one, or changing the direction of one. The general rule is to keep the mesh in quads in order to avoid unwanted surface artifacts and maintain predictable subdivision edgeflow (Ward 2008, 5). However, the restrictions imposed by the low polycount force the modeler to use triangles at several locations. A sufficiently knowledgeable artist can utilize these triangles at numerous locations in greater numbers, while achieving acceptable results. Subdividing the model is the exact opposite to low poly style and can be disregarded.

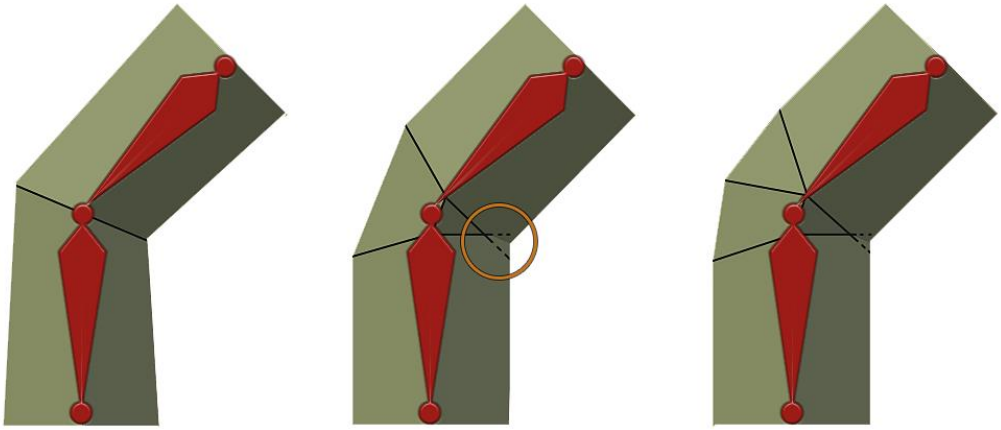

Image 5. Joint deformation in three different six-sided cylinders. Red shapes representing bones.

Image 5 displays the situation at joints. The left side of the image presents the option with the least number of polygons. This is also the weakest option, since both bones share the middle edge loop. In this way the mesh is forced to deform unrealistically, as the upper and lower quads lose their structure. The rightmost vertex is forced in towards the bone joint.

The mesh in the middle is the most viable with the least number of polygons. By adding another edge loop, one can now fully weight a single edge loop to its corresponding bone. The upper part now clips into the lower part, and so both parts retain their shape. This maintaining of form is of highest importance when considering joint topology (Ward 2008).

Finally, on the right side is shown how by adding an additional edge to the backside of the mesh, with equal distribution of weight for both bones, a smoother transition has been achieved. With the increase of only a few triangles, this is a very good tradeoff between quality and performance.

There is a level of experimentation possible with the placement of the bones. The movement of the vertices bound to the rotating bone move in relation to the base of the bone. Thus, moving the bone around within the mesh can create a noticeable difference in the way the mesh is deformed. In Image 6, the joint of the

bones has been moved to the left edge of the mesh joint. The base of the bone is now almost in the same location on the X axis as the leftmost vertices, so they barely move at all. Having chamfered the edge on the right side of the mesh, a similar clipping situation is present as in middle and right examples in Image 5. If the joint is supposed to rotate in only one direction and along only one axis, this solution becomes completely viable. In other cases problems can appear.

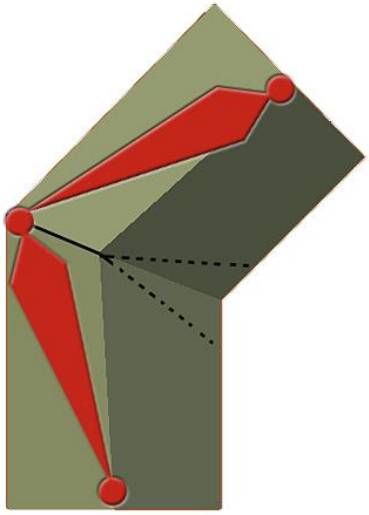

Image 6. Joint deformation with a different bone location

3.2 Topology

Topology is a critical factor when it comes to mesh deformation. Having a clean mesh that deforms convincingly, while still being well optimized, is the ideal target to strive towards (Ward 2008, 5). Edge loops should always follow some form of logic in their placement and termination. It is a good idea to use helper bones to determine possible problem areas. Customarily edges follow the contours of the body, trying to capture the subject matter as accurately as possible.

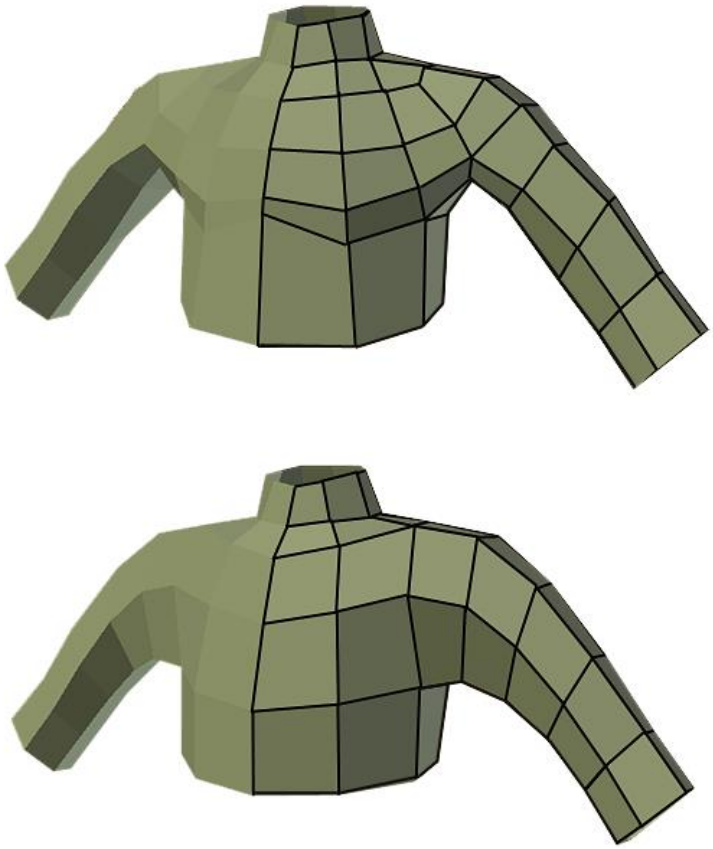

Image 7. Two different approaches to shoulder topology.

Presented in Image 7 are rough takes on two baseline methods to approaching the shoulder area with low polycounts. In the upper sample, edges have been made to follow the musculature of the body. This results in a more realistic portrayal of anatomy, and thus a well accentuated model and deformation. There is an increase in the polygons needed to achieve this, however.

The second approach is a cylinder directly connecting to the body. Electing performance over detail, this is the best alternative when dealing with extremely low polycounts. Also due to the simple nature of the mesh, the process of rigging is made considerably easier.

There is generally a lot of freedom on how to approach topology construction, as long as the basic guidelines are kept in mind. There are a variety of different topology styles that artists use. In addition to this, many artists use their knowledge to change these methods to work better with the subject matter they are dealing with. Working with a human model, no matter what polygon count, is rather straightforward with the tried and proven practices. However, when creating more fantastical creatures, there is more room for experimentation.

With sufficient knowledge of how models are rigged and how topology affects deformation, the artist can utilize triangle deformation to his advantage. Polygons are essentially sets of two triangles that, when uneven, can look very different depending on which vertices are connected. Especially in very low poly situations, a simple change of direction can make a big difference. Image 8 presents a situation where a quad polygon is deformed when the arm is rotated towards the body. On the right side of image the yellow colored vertices are connected, which results in a noticeable gap. This is an inaccurate representation of shoulder anatomy. Instead, on the left side the cyan colored vertices of the quad are connected, resulting in a more real looking form.

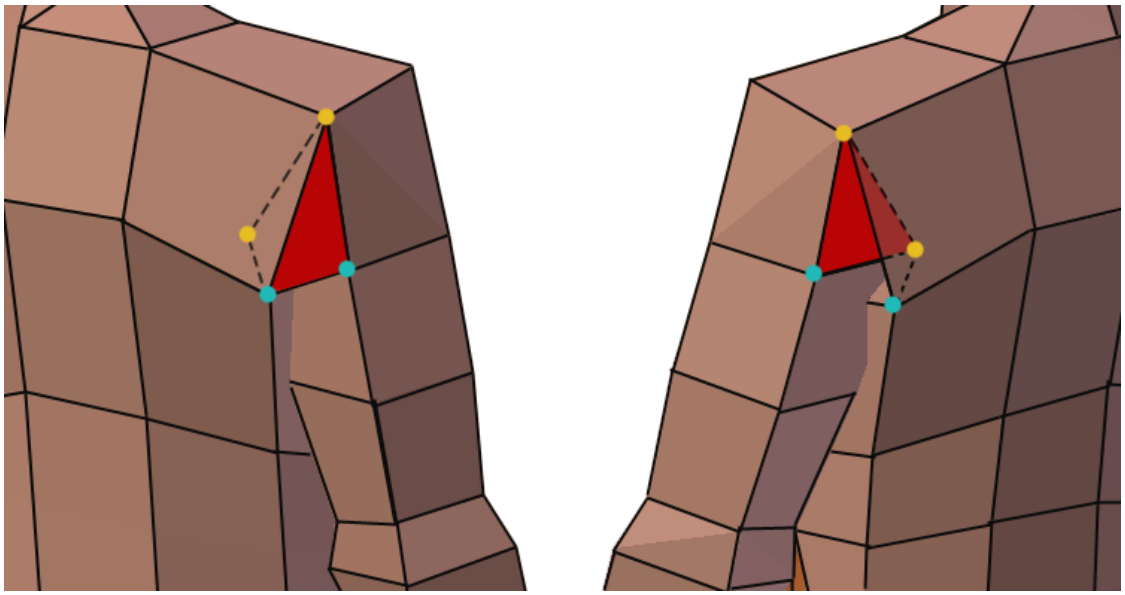

Image 8. Importance of quad triangulation at key locations

3.3 Rig

A typical game character has a rig, which consists of around fifteen to sixty bones. Approximately thirty bones is a good average number, which gives very good quality on desktop platforms and fairly good quality on mobile platforms. (Unity Documentation 2011.) This amount depends on the level of detail that the model has. The main difference between these two extremes is the bones in fingers and face. A low poly model in an environment limited by hardware is not likely to have individual fingers or complex facial expressions, so these bones are rarely needed. Bones for eye movement are a slight exception to this. If the camera can be focused on the character's face at any point, assigning bones to the eyes can be a good idea. Moving the eyes is a good and simple way to direct attention and bring life to the character.

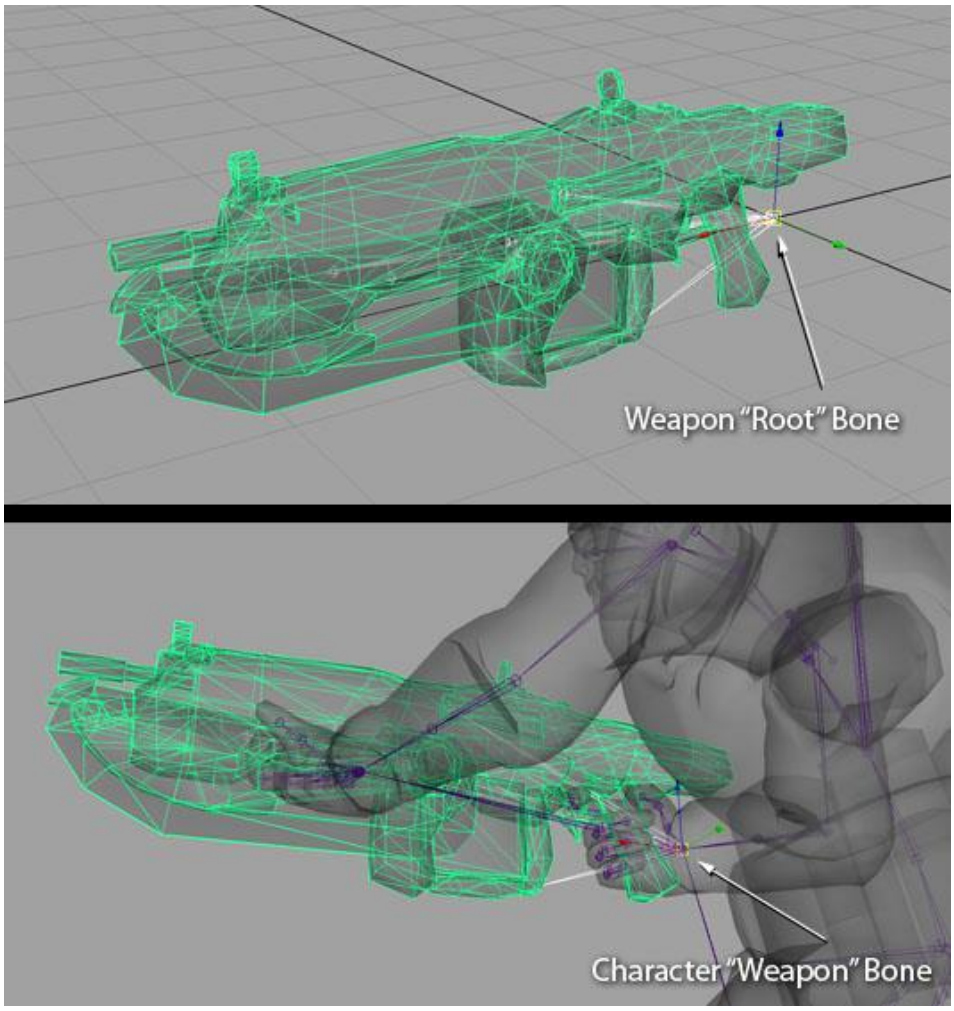

Image 9. Attaching a weapon to a character (Epic Games 2012)

In addition to the body-influencing bones, there might be a need for special-purpose bones. These can act as attachment points to which items can be inserted, or simulate the movement of clothing and other accessories. (Epic Games 2012.) Usually weapons and other hand held items come with their own bone hierachies. The character and the weapon are animated at the same time in the chosen 3D software. The weapon's root bone is constrained to the attachment bone on the character, as seen in Image 9. The character and the weapon are then exported out in two separate but synchronized animations (Epic Games 2012).

3.3.1 Inverse kinematics

“Inverse kinematics, or IK, is a way to animate a hierarchy, or chain of objects, such that each link in the chain is automatically rotated so that a lower segment is at a specific position or orientation” (Thomson Course Technology PTR Development, Jones & Oliff, 2006, 309). Having the angles and locations of joints calculated automatically, a great deal of time is saved. As seen in Image 10, one can use IK to plant the character’s feet to the ground. The character's left leg stays still while the pelvis is moved downwards, thus forming an illusion of a ground plane formed by the X and Y axes. The right leg is moved with the pelvis, so it penetrates this plane.

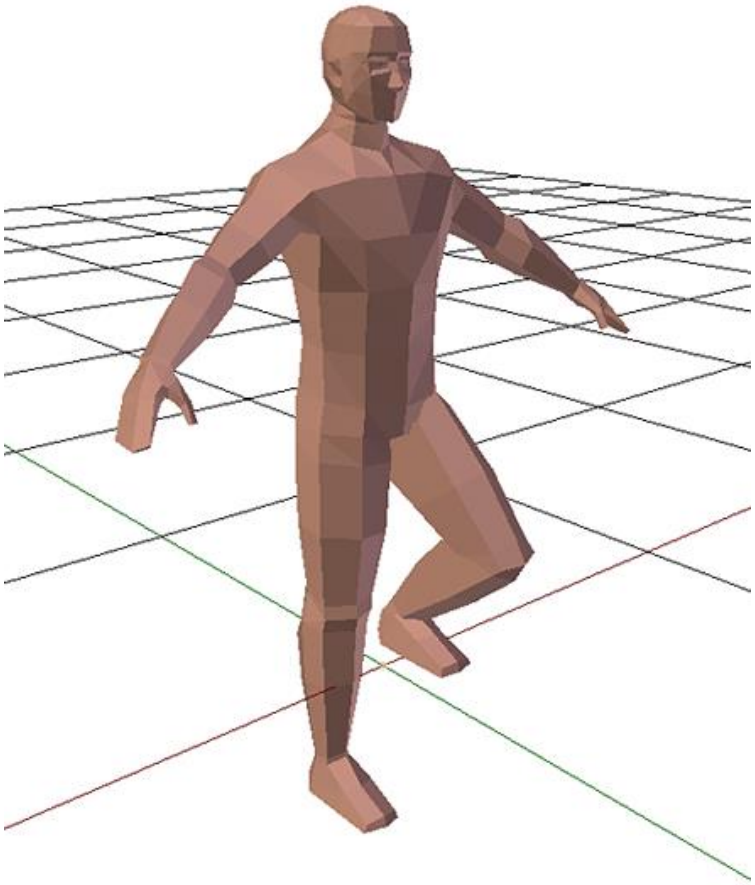

Image 10. IK functionality

Except for some advanced procedural animation cases, IK bones are not needed in a game engine and thus should be removed to avoid unneeded calculations (Unity

Documentation 2011). However, because the IK system drives the movement of the bones between the root of the system and its target, they do not have any actual motion added to them. Before removing the IK bones, the motion should be transferred over to those bones using forward kinematics (FK). Over a process called baking, these FK bones can be assigned with the transforms the IK bones go through. Here the location, rotation and scale data of the IK bone's current position are added to each individual frame in the form of a keyframe, as seen in the lower part of Image 11. This way the IK bones and any supporting bones are no longer required, and can be deleted.

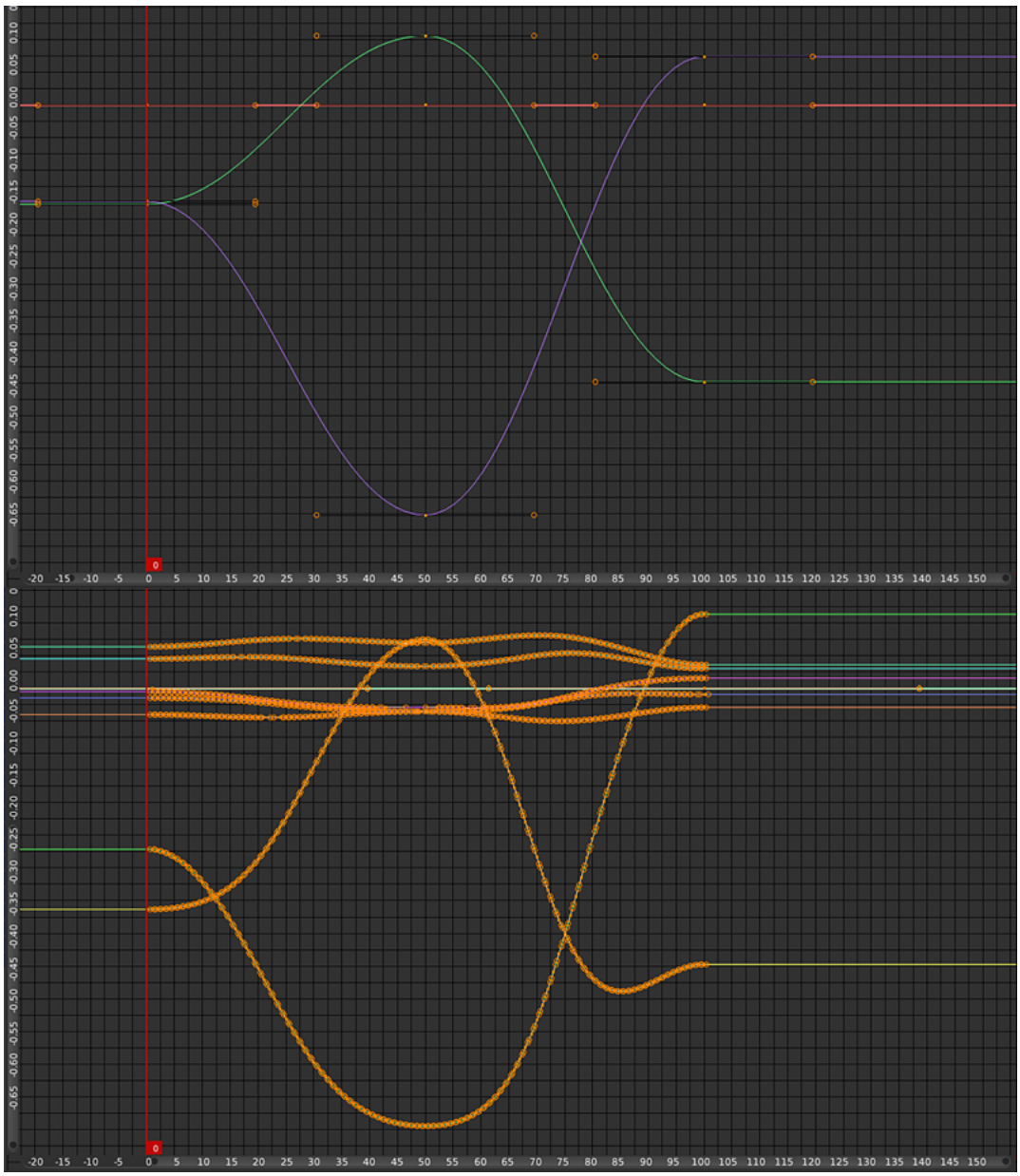

Image 11. Pre-bake bezier curves → Post-bake keyframes

3.3.2 Bone hierarchy

The image above presents the key parts of a functioning biped armature. With a total of exactly thirty bones in the exported rig, having removed unnecessary components, this would work fairly well in any situation. The model shown in image 12 would be best used in a game, where the camera is at mid-level, with some levels of zoom allowed. In the case of a real time strategy game, the bone count can be brought down by deleting some of the more detailed areas. Combining the chest bones and removing the eye, toe and hand bones would bring the count down to a well optimized sixteen. Going for less than this would noticeably impact the moving silhouette.

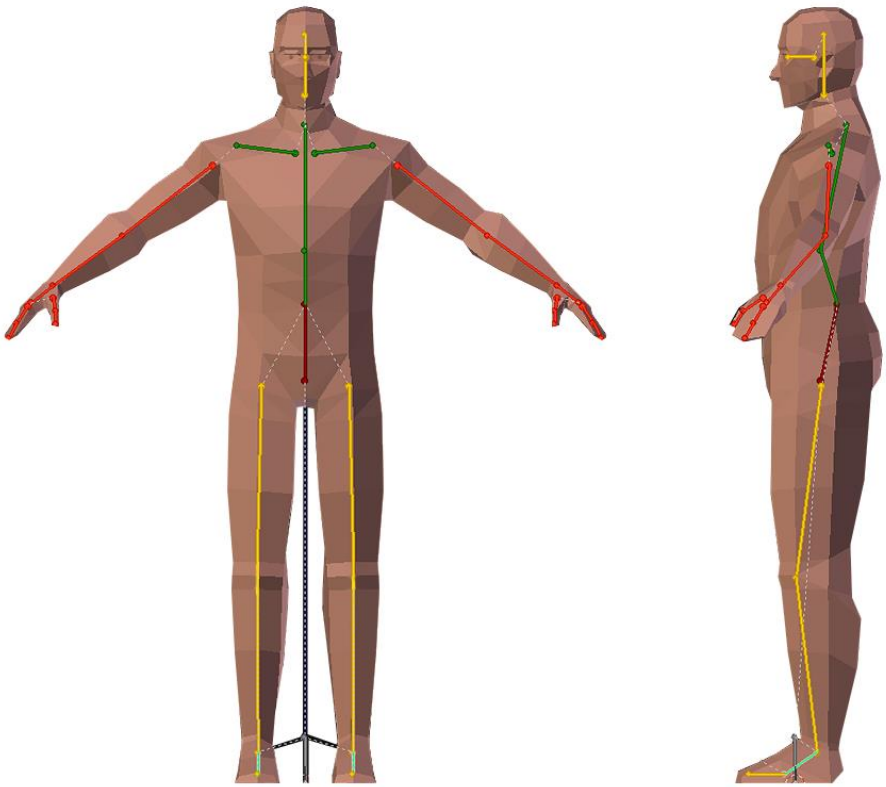

Image 12. An example bone hierarchy.

3.3.3 Root bone

Root bone is critical when making root motion animation. Here, the forward velocity of a character is calculated according to the movement of the root bone, which is represented as the red line in Image 13. This method also allows non-linear movement where the character might slow down during the animation cycle and then speed up back to normal speed. The movement could then be correctly portrayed in the game. Without using root motion, and instead letting a predetermined velocity drive the character, a visual artifact might appear, where the feet slide when they should stay in place. This happens when the speed assigned to the character by the engine is different compared to the forward distance the animation loop would travel. This can be mitigated with proper planning, but never really equals the accuracy of root bone movement.

> *Having a root bone separate of the pelvis is important. It doesn't matter that much where it is, although it may make things a bit more complicated for certain things. We find it easier to make sure animations will work together when the root is at the feet and origin. (Epic Games 2012)*

The pelvis and chest bones in most cases have some sort of internal movement or rotation to them, so using these as a root bone may have adverse effects when external transformation is applied to them. Also, their position is not always fixed to a certain height. Having the root bone at the base of the character makes it easier to visualize where the feet should be located.

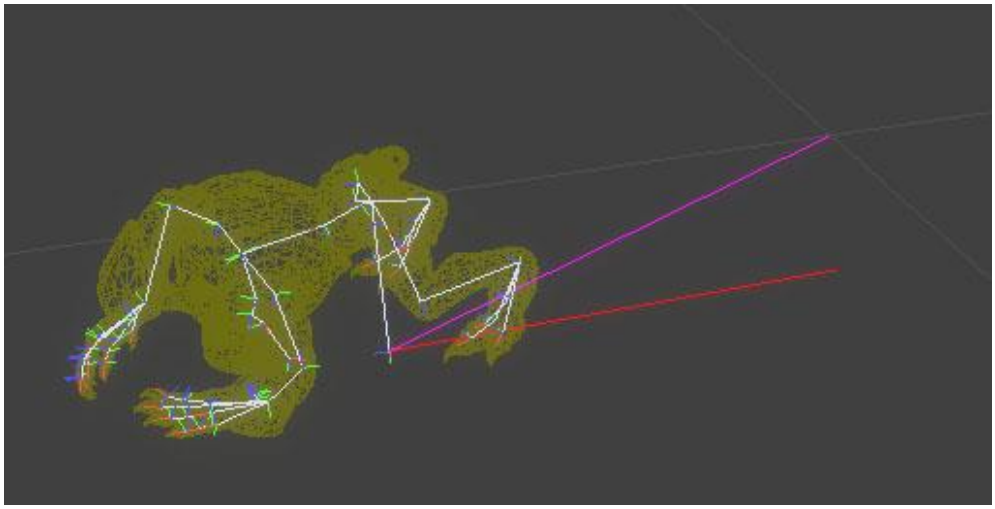

Image 13. Root motion translation (Epic Games 2012)

4 ANIMATION

Animating a character in games is a process not unlike those that are used in movies, or 2D animation. The goal is always to get as believable animations as possible. It is important to know how gravity and other forces during interactions, affects the movement of different body parts. Sometimes though, depending on the context, realism is not always needed. This is especially true in animated movies, where body proportions may be stretched to emphasize the action. Beyond the laws of physics, animation is also acting. Telling a story through action and giving the character a personality and emotions is an important part of this process.

4.1 Game animation

There are two ways characters are moved in games. An easy straightforward method is doing it directly through code. Assigning numerical values to determine speed is a simple task to program and manage. After establishing a consistent unit system between the game engine and the modeling program, a walk cycle could be animated to move forwards exactly one meter. How fast the character moves forward can then be calculated from the animation. The more accurate the calculations are, the less the character will display a sliding effect, which comes from the discrepancy between animation cycle speed and the value inserted into the code.

The second method is to use root motion. Here, the forward velocity is extracted straight from the movement of the armature's root bone, and forwarded to the game engine. Movement of biological entities is hardly ever mechanically precise: they sway from side to side or speed up and slow down in turn. Thus, with root motion this level of realism can be achieved without error while without, there would be a noticeable sliding effect.

Animating a character for a game has some unique challenges to the process. The movement can usually be seen from multiple angles so it must look correct from all points of view (Landgraf 2014). Most animations are also created in loops,

where the start and end frames are made to be in the same pose to ensure smooth looping transitions. The number of animated frames needed in between them depends on what kind of animation is in question. A walk or a run cycle could start with the left leg firmly planted to the ground, while having the right leg do the same in the middle of the loop. The animator would then proceed to add more motion information between the extremes and the middle point. In Image 14, the walk animation has been divided for a second time. These five states form a good base for the mechanics of walking. The second and fourth poses are present to fix some of the issues that come with automatic keyframing between the extremes and the middle. From here on out, the animator would start adding personality and overall polish to the character with more detailed movements.

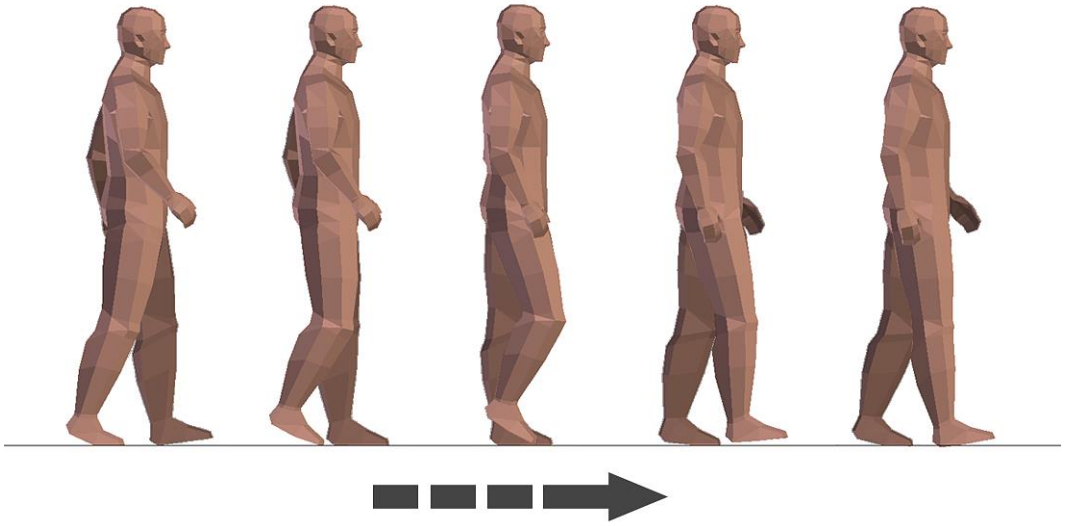

Image 14. Walk cycle broken down into five poses

For the majority of the game, the person playing it is in control of the character. With the unpredictability of a player dictating what actions are made and when, all animations should allow smooth transitions from one to another. The most basic computer assisted mechanic to this, is linear interpolation of positional and rotational values. Instead of instantly switching poses from the current animation to the beginning frame of the following animation, a computer generated transition animation is calculated between these two animations. Making the start

frame of an action resemble the end frame of a common loop animation which it is related to, such as idle, walk or run, makes the transition barely noticeable.

It is plausible that a game will be featuring actions that link different animations together, such as shooting while walking, or running in a straight or crouched pose. In the past, animators were forced to account to these by making dozens of custom animations for each possible variant. The introduction of parametric animation in the mid-2000's reduced this workload drastically. Here, the animator would only have to construct the core animations, which could then be blended together in the game engine.

4.2 Principles of animation

The principles of animation deal with creating an illusion of real, thinking and feeling characters, which adhere to the laws of physics. From the 1930s and onwards the animators of Disney studied how to express the personalities of their characters through movement, by searching for ways to better relate drawings to each other (Thomas & Johnston 1995, 47). All this knowledge was then gathered into a book called The Illusion of Life: Disney Animation. This book spawned Disney's twelve basic principles of animation, which is an assortment of conventions that are important in achieving realistic animation. Some even go as far as labeling this book the “Bible of animation”.

It is important to note, however, that one can create brilliant moments of animation without studying, or even having knowledge of these principles. A lot can be achieved with common sense and logic. Talent in understanding and performing social interaction is also useful. Animators are special in that they need to be competent actors in addition to being proficient on the technical side of the craft. They need to convey their message in a credible fashion to sell the scene to the audience, just like actors do. Through breaking down a situation into its fundamentals, all its emotional states and how to present them, the animator already has a lot to work with in building an effective scene.

While these principles were originally made for drawn animation, they have great relevance in 3D computer animation as well. There are a few differences that

come with the change of medium, mostly having to do with the fact that animating the so called “in-between” frames has been automated with mathematical calculation instead of doing it manually. While animated movies can utilize these principles to their full effect, animating for games poses some difficulties due to their action-based nature and limitations brought upon by game designers. Also, animators in video game studios are rarely allowed time to learn and improve their skills so they mostly get better at being fast and efficient (Rose 2013). However, some of these methods have good effort to payoff ratio and should always be paid attention to.

4.2.1 Anticipation

People watching an animated scene will have difficulty understanding the events unless there is a sequence of actions that lead from one activity to another (Thomas & Johnston 1995, 51). Adding anticipation to an action that a character is performing makes it feel clearer and more real. Whether a physical motion or narrative progression, they need to be preceded somehow. This can be overlooked if the goal is to surprise the viewer.

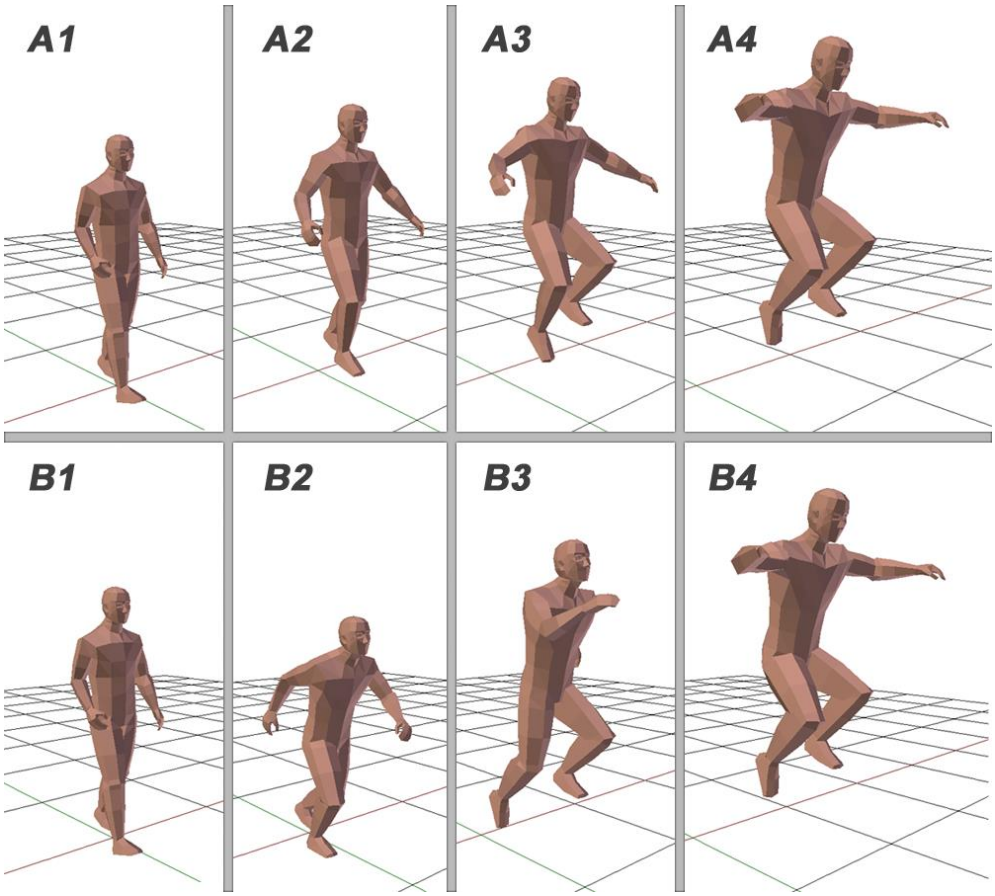

Image 15. Walkcycle to jump action with and without anticipation

Even though the transitions from one action to another can be near instantaneous, adding a few frames of anticipation provides a noticeable difference. In the upper row of Image 15, the character moves straight from a walking animation to a jump animation in a mathematically direct fashion. Below there are two frames of anticipation added, which makes the jump feel more organic and also to make it seem like there is an intention to jump.

4.2.2 Staging

When animating a character for a situation where a special mood is set up, his pose should clearly communicate it to the viewer. It has been found to be the best practice to show this in the character's silhouette. (Thomas & Johnston 1995, 56.) In Image 16 there are screengrabs from a variety of walkcycles, each showing a different mood. Exaggerating the poses makes the character’s disposition clear. It

is especially important in games where the player camera is placed far away from the character. It is quite hard to read small and subtle movements from afar.

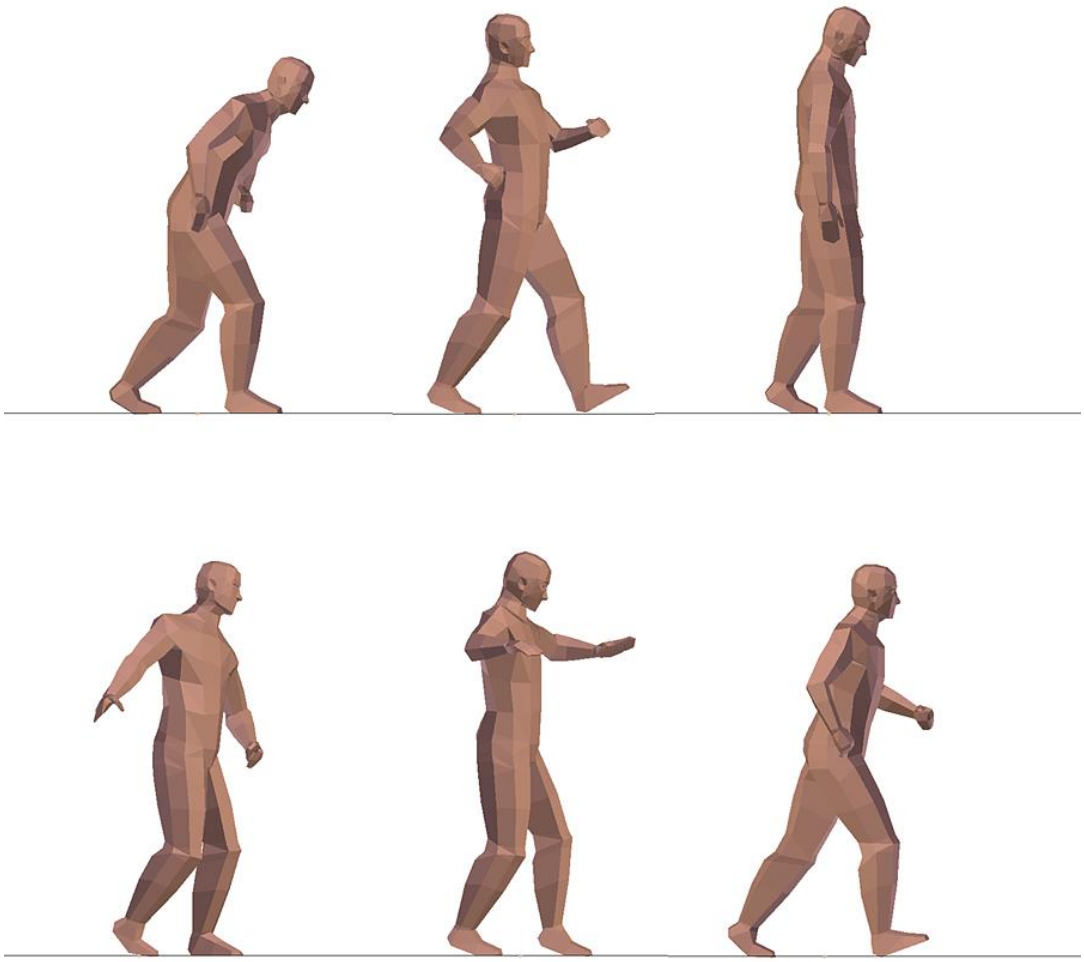

Image 16. Establishing mood with a pose

4.2.3 Follow through & Overlapping action

As a character finishes an action and his main body comes to a halt, the other parts connected to it maintain velocity towards the direction of movement. This happens due to momentum. Not only does this movement give the action more credibility by working according to the laws of physics, this also makes it more interesting. When working with looped animations one needs a unique finishing

action to accomplish this. Clothing and accessories can be simulated using a physics engine if the game engine supports one. It is common to use physics to simulate the character falling when dying or otherwise unable to function. This ragdoll effect is a good way to simulate follow through with little effort even though at times the uncontrolled flailing can start to look rubbery and fake.

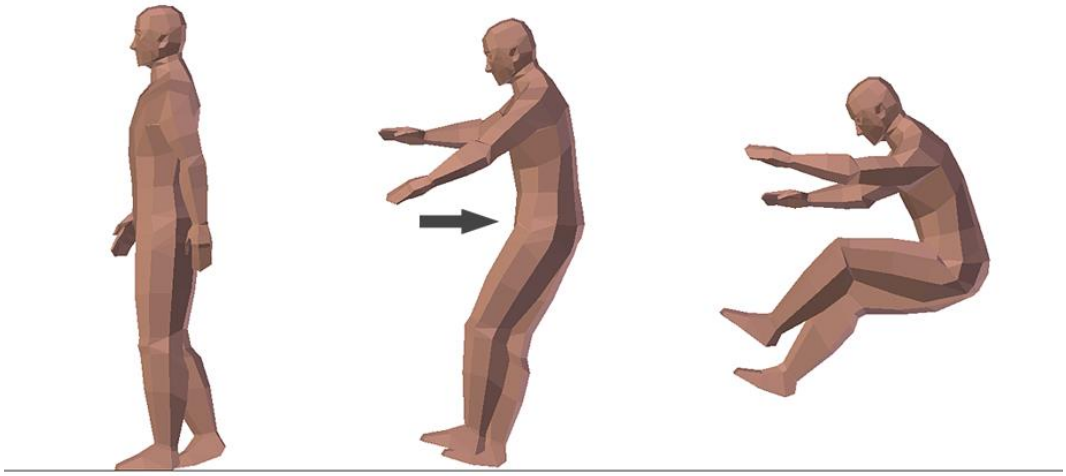

Image 17. Overlapping action in a sudden change of direction

Overlapping action works on the same principle. As the core body of the character changes direction, the other parts of him do not immediately follow. Instead, they drag behind for a few frames, as can be seen in Image 17. Applying a sudden force on the body of the character propels it backwards, while the arms stay still for a moment.

Associated with these principles is an idea called “moving hold”. A character who is not doing anything could stay absolutely still to direct attention to the important parts of the scene. This, however, looks boring and lifeless, so some minor movement should be added where the starting position is accompanied with a more extreme version of it, while still containing all the elements of the pose. (Thomas & Johnston 1995, 62.) Game characters come with looping idle animations just for this purpose.

5 CASE

The basis for this case was the interest in making a rather simple but robust arena combat game for the PC. The main source of inspiration came from a game called Bloodline Champions, a top down skill-based arena game. While these types of multiplayer online arena combat games are quite common, this variation to the genre is quite rare. The ultimate goal was to build a somewhat functional prototype. It could then be evaluated if the game has enough potential to put more effort into it.

5.1 Design

The basic gameplay design was the first important step of the process. The game that inspired this project had many elements that were done so well that it would be very unwise to try changing them in any significant way just for the sake of changing. At the forefront is the player camera. Having it look down on the character in a slight angle, approximately seven to eight meters away, provides a clear vision of the environment around the character. Also, making the camera react to mouse position gives the player more sight range for longer ranged fighting and spotting.

The combat is also heavily influenced by Bloodline Champions by focusing on having to aim all the skills and using a cooldown system to regulate how often the skills can be used. Implementing skill casting time with the possibility to interrupt is also very important to keep the game focused on trickery, mind games and dodging skills. The first major difference is reducing the number of useable skills from seven to five. With this and reducing cooldowns to only a few seconds maximum would make the game a little bit faster and slightly more forgiving.

The environment is where the biggest changes come to play. Instead of open, static maps with just walls and buildings blocking sight, the maps should have interactable objects. Some of these mechanics could for example be indestructable force fields that could be turned off by destroying a power generator or extendable bridges over wide chasms.

Image 18. Generator diffuse texture

The art style for this game is a simple low poly look with hand painted textures. This is mostly an aesthetic choice due to the personal preference of the author, and the fact that low poly models and small textures are simply fast to produce. Showcased in Image 18, the textures rely mostly on solid color usage and gradients. The texture is sized at 128x256 pixels, so smaller details are not needed. The need for even some of the present detail is debatable. In the end, the

idea is to maintain a clear distinction between characters and the environment, and to keep the detail versus flat surface ratios in a good balance.

5.2 Asset creation

The program used to do all the modeling is Blender, open-source 3D modeling software. Its capabilities are at the least comparable to mainstream modeling programs such as 3ds Max. The UI and some of the design choices make it quite difficult for a beginner to start with, but with proper dedication to learning, it becomes very fast and easy to use. Being open-source means the program is also free to use, which is ideal for a project with little to no budget.

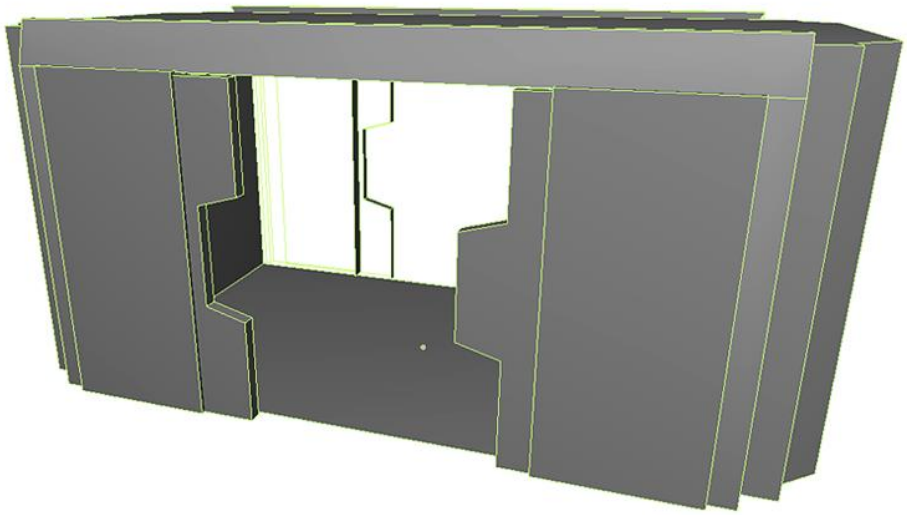

Image 19. Shipping crate model

When modeling the environment assets, some heavy optimization was implemented. The models themselves are low in polycounts to stay consistent with the art style and on top of that there is some strategic polygon removal. Since the camera will look down on the action from high above, there is no need to retain downfacing polygons in the models. Seen in Image 19, even some of the polygons facing sideways have been removed. Because of the roof of the model and rotationally constrained camera, these polygons are unnecessary.

The textures were done using GNU Image Manipulation Program, more commonly known as GIMP. Being also open-source software, the same free use policy applies, with its benefits. Even though this program is not quite up to par with Adobe Photoshop, it has enough functionality to allow drawing textures without any major problems. The process itself is quite straightforward with the majority of the work being mechanical by its nature. The biggest challenge is coming up with interesting patterns and decal designs

5.3 Characters

A driving decision in designing the art style for the project was the idea of trying to make all characters with exactly or under 300 triangles. The characters are all mechanical so the blocky shapes and hard angles work well in this context. There are some cylindrical shapes though that would require a rather large edge count to them in order not to look out of place. In Image 20 this was solved by painting in a perfect circle while leaving the edges very dark. Especially when seen from afar, the eye focuses on the bright yellow circle and creates an illusion of more polygons than actually exist. If seen up close and a little to the side, the eight-sided cylinder becomes apparent.

Image 20. Playable robot character

A major aspect of the game's visual design was character animation. While the textures attempt to make the characters sympathetic and easy to look at, the animations are there to give them personalities. Ideally, each character should have a distinct personality to them, which would correlate with their design. Best places to display these character traits are the idle, walk and long skill animations. When animating the robot in Image 20, emphasis was put into making the character feel playful. There was some contemplation if the animations should be more exaggereted, but this might have been counterproductive when the focus should be in the fast paced action.

5.4 Implementation

The engine that the game uses is Unity3D. The main reasons for this choice are the ease of use and general familiarity with the software. There is a free version of this game engine available, which allows commercial use up to a certain amount of revenue. The downsides of this version are some limited or removed features. Acquiring a pro license is not completely out of the question though, since its cost is within realistic limits.

The code for the game was written in C#. Some basic features were built for testing purposes. These were enough to try and find a good movement speed, camera height and general map design. The screenshot in Image 21 showcases the look the game is aiming for. The latest development build gave a positive feeling about the game's potential.

Image 21. Screenshot from the case game

6 CONCLUSIONS

Building animated low poly characters is an interesting field. Optimizing characters while retaining good animation capabilities is challenging and sometimes requires ingenious desings. Texture painting is also very important, testing the artist's skills in painting good looking details on flat surfaces.

Animating low poly characters appeared to be a more fun and relaxed experience than expected. Animating realistic looking characters demands realistic organic movements and that the body is constantly in motion in order to look real. Especially hands and fingers were noticed to be very important to make the arm animations seem fluid. In low poly there is the inherent stylization that is also present in animated cartoons and movies.

In conclusion, low poly characters are most likely to stay relevant no matter what sort of new technologies appear. The latest developments in mobile graphics already push the devices to their limits to achieve an unbelieveable level of visual quality. This however does not push video games forwards as a medium but instead allows for even more expensive games to be made for mobile platforms. It is the author's belief that the model of producing increasingly expensive games is eventually going to become unviable.

The development of the case in this project was slightly problematic. The time that was initially planned to be used for improving coding skills was unfortunately assigned elsewhere. For this reason most of the time devoted to the project was modeling, animation and game design work. The aspect of the game most relevant for the thesis was achieved though, which is movement of an animated character.

REFERENCES

Strong, B. 2007. Creating Game Art for 3D Engines. Boston: Course Technology / Cengage Learning

Thomas, F & Johnston, O. 1995. The Illusion of Life: Disney Animation. New York: Hyperion

Thomson Course Technology PTR Development, Jones, A. & Oliff, J. 2006. Thinking Animation : Bridging the Path Between 2D and CG. Boston: Course Technology / Cengage Learning

Ward, A. 2008. Game Character Development: Digital Sculpting for the Realtime Artist. Boston: Course Technology / Cengage Learning

INTERNET REFERENCES

Epic Games. 2012. Creating Animations for the Unreal Engine [referenced 25 Jan 2014]. Available: http://udn.epicgames.com/Three/CreatingAnimations.html

Gamasutra. 2005. Videogame Aesthetics: The Future! [referenced 19 Mar 2014]. Available: http://www.gamasutra.com/view/feature/130840/videogame_aesthetics_the_future.php

Landgraf, H. 2014. The Increasing Role of Character Animation in Video Games [referenced 30 Mar 2014]. Available: http://www.animationarena.com/character-animation.html

Truth Labs. 2013. Creating the Low-Poly Look in WebGL [referenced 3 Apr 2014]. Available: http://blogs.truthlabs.com/2013/10/02/creating-the-low-poly-look-in-webgl/

Unity3D. 2011. Modeling Characters for Optimal Performance [referenced 25 Jan 2014]. Available: http://docs.unity3d.com/Documentation/Manual/ModelingOptimizedCharacters.html

IMAGES

Image 1: Valtteri Jolma

Image 2: Screen captures from the game Starcraft 2: Heart of the Swarm.

Image 3: Screen captures from the game Torchlight 2.

Image 4: http://supercell-www-content.s3-website-us-east-1.amazonaws.com/prod/cache/images/made/aeae5845db88781b/clan_screenshot_1_1800_1350_60_s.jpg

Image 5 - 8: Valtteri Jolma

Image 9: http://udn.epicgames.com/Three/rsrc/Three/CreatingAnimations/3P_Weapon_Attach.JPG

Image 10 - 12: Valtteri Jolma

Image 13: http://udn.epicgames.com/Three/rsrc/Three/RootMotion/RootEnd.jpg

Image 14 - 21: Valtteri Jolma

APPENDICES