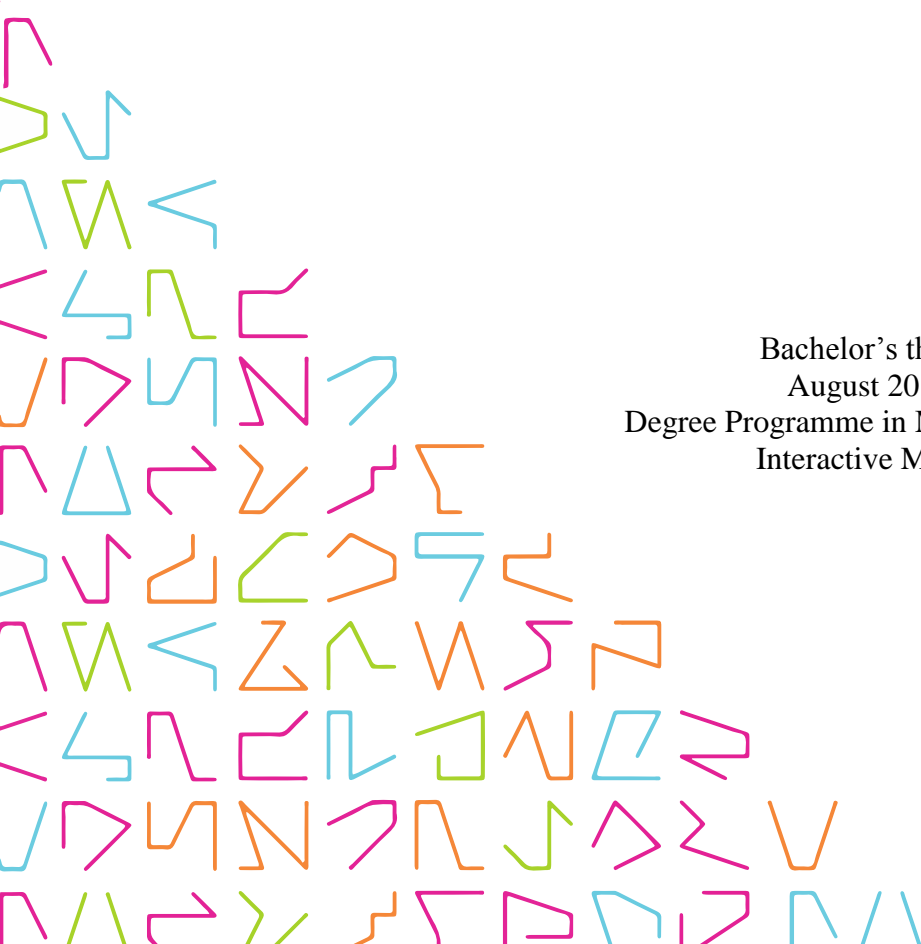


The Production Pipeline of Video Game Animations

Andreas Heinrich

Bachelor's thesis
August 2018
Degree Programme in Media and Arts
Interactive Media



ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Media

Heinrich, Andreas
Production Pipelines of Videogame Animations

Bachelor's thesis, 51 pages
August 2018

The purpose of this thesis is to give a brief summary of animation production pipelines in video game production workflows. It progresses by briefly going through all the steps required in producing an animation and clarifying their value. At the beginning, there is a detailed explanation about the functions of the team members who have influenced the creation of animations. It is followed by an explanation of the individual steps which have to be made in order to complete a functioning animation.

The included information is partly based on personal experience from animation projects completed in professional environments. This experience has highlighted how standardized the practices of the industry are across multiple professional animation houses. The experience of working with several different developers has made it apparent that despite aesthetic and internal company-specific differences there are similarities in team dynamics and approaches. The approach used in this thesis was to find the similarities and describe their connection to one another.

It is common especially amongst beginners in the industry to be misinformed or under experienced in the ways how company dynamics and communication work. The intent of this thesis is to provide an easy to follow walkthrough for an aspiring animator.

Keywords: pipeline, 3d, animation, workflow, guidelines, hierarchy.

CONTENTS

1	INTRODUCTION	6
2	The Hierarchy within a Company	8
2.1	Creative Director	9
2.2	Creative Lead	10
2.3	Lead Animator	10
2.4	Animators	12
2.5	Coders	12
2.6	The Creative Part	12
2.7	Game Design Document	13
3	Communication	15
3.1	Time Management	15
4	Feedback Session	17
4.1	Feedback on the Tasks	17
5	Software Used	18
5.1	Various Interfaces	18
5.1.1	Maya	19
5.1.2	Blender	20
5.1.3	Unity	21
6	The Pipeline	23
6.1	Research and Development	26
6.2	Modelling	27
6.3	Texturing	30
6.4	Armature	31
6.5	Weight Painting	32
6.6	Rigging	34
6.6.1	FK – IK	35
6.7	Animating	36
6.7.1	Graph Editor	37
6.7.2	Dope Sheet	38
6.8	Principles of Animation	38
6.9	Difference between In-game and Cinematic Animations	41

7	Implementation.....	43
7.1	Baking.....	43
7.2	Implementing.....	44
7.3	Naming Conventions.....	44
7.4	Testing.....	46
7.4.1	Does it Work?.....	46
7.4.2	Does it Look Good?.....	46
8	Conclusions.....	48
9	References.....	50

ABBREVIATIONS AND TERMS

3D Model – 3D object with representative features of a real life object or person.

Cut Scene – Narrative driven animated scene during a game play session

Engine Rendered – Process of the real-time visualization of three-dimensional space

Lead Artist – Leading person of an artistic production group.

Mudbox – A program offering 3D sculpting

Nurb – Non-uniform rational B-Spline

R&D – Abbreviation of the term “Research and Development”.

Sculpt – Shaping a 3D object by directly altering its surface properties.

Viewport – Screen or window dedicated for the preview of a three-dimensional space with its contents.

Walk Cycle – Seamlessly looping walking animation.

X,Y,Z axis – Three-dimensional coordination system to pinpoint a position

Z Brush – A program offering 3D sculpting

1 INTRODUCTION

The scope of this thesis is to create a functional synopsis of the processes involved when creating animations for a video game. The emphasis will focus exclusively on working animation pipelines that exist within the industry. There are two reasons for this decision. The initial reasoning is founded upon personal experience and understanding of the industry and its processes; specifically experience within rigger, animator and concept artist roles. This understanding has been gained through professional experience within several Tampere-based game production companies. These companies have international presence within the industry space, and their achievements include working closely with industry leaders as partners, which has internationalized their reputation and standing.

The second reason is an awareness of shared and personal misconceptions of aspiring creatives who wish to enter the game industry. This misconceptions has been highlighted through interviews and mentoring sessions with company interns, academic students and new recruits to the industry. The aforementioned misconceptions relates to the assumption of having larger artistic liberties than the reality provides. The reality, is often, that those liberties are frequently overshadowed by the hierarchical structures, the technical limitations and the personal intentions of artistic leads that exist within the corporate structure of game production companies. Therefore I deemed it to be necessary to summarize a list of ideals and guidelines I have experienced myself. To reinforce this summarization, relevant experiences have been detailed, along with supporting references, in order to display the seminaries of individuals across the industry.

The following chapters intend to clarify aspects of the personally experienced pipelines, the impact that they had, and the effectiveness of their usage within these specific environments. Right after the introductory chapter, the progression starts with the clarification of hierarchy within a company. Within this, the skills and experience that is required within production roles will be defined and explored, including how these roles contribute to the final product.

To provide an overview of this thesis, I will first briefly explain a company's internal communication system. This will be followed by guidelines and the basics of a feedback session - an often overlooked but vital part in the creative process. The next step will be

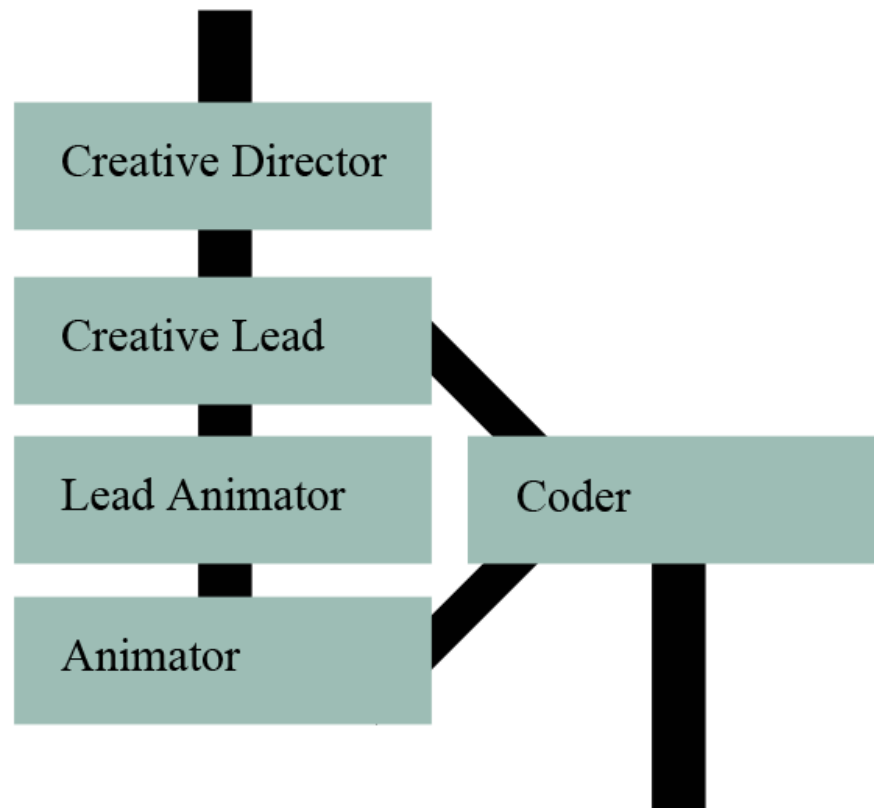
to outline the programs predominantly used in the aforementioned companies and give a brief idea of the tasks that can be achieved with them.

The next chapter is about the pipeline itself. During this chapter I will speak in detail about all the steps needed to be taken in order to bring the basic concept into a final animation. The chapter “Implementation”, will discuss some of the technical hindrances experienced by animators during the final production implementation and implementation phases. And the last chapter will clarify the conclusion.

2 The Hierarchy within a Company

The hierarchy within a company is a system in which certain co-workers have direct authority over others. It has shown itself to be fruitful in groups of 4 – 6, as described by Sutherland. “The hyper productive teams would always split into subgroups of 7 or less, while the poorer performing teams insisted on working as a group of 15.”(Sutherland, 2003). The Supervision and communication between smaller groups tends to be more efficient. Inexperienced groups of developers tend to insist that a hierarchy is absent in their groups. However, that usually results in an incoherent aesthetic in the final product.

“If the middle management team does not have effective channels of communication with the executive team, important company information could take days before it reaches the entire staff.” (Root, 1985). While Root appears to be out of context and outdated, the point still applies to developing teams and is still relevant. The hierarchy opens up a chain of command with specifically creative projects, in particular projects that take place in a strictly-followed social structure in a company. “Creatives are typically more neurotic, more antisocial, and less conscientious than others.” (Chamorro-Premuzic, 2017). Large, fragmented groups never seem to get a proper project ready on time and on budget, which could be due to the reasons suggested by Chamorro-Premuzic. Having hierarchy is having supervision, from a few people, over the output of the entire team. A smaller group of leads often have a better overview of the visual and technical cohesiveness of the entire product. “A community with several arts groups can achieve a synergistic boost from the combination of programs and activities.” (Byrnes, 2003, 131). Byrnes suggests it to be beneficial to keep individuals who are having similar tasks to remain in smaller specialized groups.



PICTURE 1. The hierarchy of an animation pipeline for a game

The graphic represents the hierarchy of a pipeline for a video game animation. It is important to note that the coder (the person in charge of implementation) is situated as a successor of the animator. However the coder also receives input from the creative lead. This input is normally a clarification of the technical necessities. In this case the coder features a similar and potentially higher rank than the lead animator. This is because the lead animator is in charge of the animation department's workflow and may not be fully aware of the technical requirements.

2.1 Creative Director

“Deciding exactly what it is we want to do, setting realistic goals (what the organization intends to accomplish), and then determining the objectives. (the specific steps to take and the timetable for completing the tasks) to be used in meeting the goals is hard work.” (Byrnes, 2003, 24). Byrnes defines a fitting summary of the problems faced within production organisations. This could be interpreted as an argument for the role of “Creative Director” within hierarchical structures. Creative Directors are the professionals in charge

of the artistic or technical consistency of a project. It is the creative director's influence that decides which finished assets end up in the final product.

Set and expand the overall Design Vision for BioWare Live Service games, providing the blueprint for the overall game experience and directing our functional teams (writing, ux, systems, gameplay) to bring the game vision to life.” (BioWare Creative Director Application, 2018.)

As the career requirements of Bioware suggest, the role of Creative Director is more about managing the different teams with the company. The Creative Director needs to have an intricate understanding of the products intensions, marketing strategies, design requirements and technical aspects of the project. Based on their assessment of a projects status, they should delegate a set of tasks onwards to the respective leads, such as the lead animator(s).

2.2 Creative Lead

The Creative Lead is a relatively new job title and refers to a person who is situated between the Creative Director and a lead animator. The primary function of the Creative Lead is to work with the “more creative groups” of a company. Specifically, they should focus on technical solutions for artistic purposes. An example would be working closely with the coding teams in order to understand the technical limitations and find solutions for achieving the intended visual aesthetic. This job is still relatively rare and more commonly found in bigger companies with a heavy reliance on technology companies. In contrast, smaller production teams often assign this range of tasks to co-workers who have a skillset focused on the technical aspects.

2.3 Lead Animator

The Lead Animator is in charge of the animation team. Once the visual and technical requirements reach the lead animator, then it's within the lead's tasks distribute the tasks amongst the animators within their team.

Infinity Ward is seeking a Lead Gameplay Animator who will bring innovation and creativity to our character animation. We welcome a motivated

and inspirational leader that is technically sound, who will work with team leaders to create unrivaled character animation for our upcoming titles.

Responsibilities:

- Manage and inspire a team of talented and experienced Animators.
- Work closely with the Animation Director to establish character motion and personality.
- Direct motion capture talent to achieve a sense of realism and authenticity signature to Infinity Ward games.
- Work closely with the AI Programmers and Designers to create animation state machines and blend trees to drive smooth and fluid animation flow.
- Create high quality realistic keyframed and motion capture animations, demonstrating a strong sense of posing, timing, and weight. (Activision Lead game play animator Application, 2017.)

As Activision suggests, a lead animator is required to have an understanding of the multiple steps in creating a character which can be animated, an understanding of animation practices, and a capability for communicating with people. The lead animator must, at the very least, comprehend the workflows of modelling, texturing, setting up armature, rigging and animating. That gives them the opportunity to explain to their workers how a certain process should work in order to make the workflows interconnect. In addition to having technical knowledge, the lead must be able to properly communicate, in order to distribute the proper information either downwards to team members, sideways to other leads or upwards to higher management hierarchy. Even within a group of two or three animators, the creative visions of each individual can go in vastly different directions. These different directions can lead to the development of components which are often not required and do not make it into the final product. These discarded ideas and unnecessary tangents can lead to increased time expenditure and budget costs. Creative leads should have the ability to prevent this from frequently happening, and also should have a wide range of experiences related to their tasks.

2.4 Animators

Animators are the force behind any production which feature animated content and they are the ones who are tasked by the leads. The term animators, or ‘workers’, in this context, refers to different groups of people who belong to their respective subsection. The Lead Animator commands a group of animators / workers. The lead sound designer has authority over a sound designer and the composer; the lead artists over the artistic workers, and so on. The workers main task is to drive the ideas and concepts forward. They have to retain the technical and artistic requirements of the project. Creativity is usually restricted to the expectation of the leads.

Larger groups tend to limit the workers of their respective branches to their branch specific set tasks (animators only animate, concept artists do concept art etc.). Smaller groups like “Indies” (independent game developers) can shift between the branches, for example, if the animator happens to know aspects of sound design. In smaller / independent production companies, this is frequently due to the convenience that having team members with multiple skillsets has upon reducing costs and time in production, in the absence of specialists.

2.5 Coders

Coders are essential members of any technical department. They are the ones who make the “technical part” work. They do have their own hierarchy, and for the most part, work outside of the animation department’s reach. Nonetheless, animators and coders do need to communicate with each other – particularly the coder, who is tasked with implementing and organising the created assets in the engine. The coder, who implements an animation, normally “outranks” the Lead Animator. Animators usually work in a slightly more creative part of the team, and on occasion, unintentionally create content which is visually appealing, but does not meet the technical requirements. It is within the tasks of “the coder” to convey to the Lead Animator the technical specifications needed.

2.6 The Creative Part

By having many different roles with authority over creative assets, it is sometimes hard to estimate where the creativity in the process resides. The holders of the original vision

(They can be either creative directors or writers with the authority over the game design document) of the final product require a certain quality in the end results, however they are not always capable of comprehending the inner workings of the animations. This is normally where Creative Directors and different leads come into play. They establish the guidelines about how the animations are supposed to look.

Creativity in the sense of artistic creativity does not directly apply here. Where the creativity lies is in problem solving. For a “common” animator, there are two major problems to be solved: The first is to gain understanding on the artistic direction of the final product (is it realistic or cartoonish?). Additionally, animations are highly technical assets, and in order to make them work in a game engine, technical discrepancies need to be addressed. The Animator is required to solve all of the hindrances, gaining a certain understanding of the intended quality of the final product. The hierarchy of directors, creative leads and team leaders are required to reach and communicate an understating of the implemented framework. Once the framework is established, the artistically creative work of creating animations can be started.

2.7 Game Design Document

The Game Design Document (sometimes called a ‘Production Bible’) is a document featuring a summary of information about the details of the entire production. It is essentially the whole game with all its needs, designs, and concepts written down. It must feature an explanation of the lore, clarifying influences, pinpointing the visual style, control schemes and much more.

A game design document can explain important details for an animator. For example, if a character is limping or has a habit of scratching its neck. The entire team has to have access to this document to solidify the intended look of the final product.

A game design document acts as a nexus and hub to connect and list every aspect of a game. It consists of written descriptions, images, graphs, charts and lists of information pertinent to each segment of development, and is often organized by what features will be in the game, and clearly lays out how they will all fit together. (Sayenko, 2015.)

As Sayenko suggests, a game design document is a written record with added images featuring all aesthetic and technical standards of a project. For an animator a game design document can explain important details about the animations. For example, if a character has a limp in its left leg, or if it has nasty habits such as scratching its neck. It is very important for the entire team to have access to this document. It helps to solidify the intended look of the final product. The Game Design Document creates the base on which directors and leads delegate the tasks, as it should clarify all aesthetical and technical requirements.

3 Communication

“The whole idea of work being a social activity is common today. People working in arts organisations, like those in other professions, spend significant portions of their waking hours working with others” (Byrnes, 2003, 85). As Byrnes suggests, a significant amount of time is spent, within any profession, in working alongside a group of individuals of various professions. It is beneficial to the entire team that the communication between each member is well worked out. There is a tendency for people to only communicate with others in their particular fields of knowledge. For any group, it is advisable to occasionally interact with members of the different branches. That not only helps to raise the general enthusiasm of the team, but also provides a possibility for team members to develop an understanding of each other’s workflows. The desired goal is to develop a state of seemingly “silent agreement” between team members whose tasks are interconnected. The main feature of this state is to be able to deeply understand the workflow of every other team member and expect results of a certain quality within a specific timeframe. This state is commonly reached not only by having worked together for a long period of time, but also sharing extensive extracurricular activities such as coffee breaks and work/social events. It proves itself beneficial on multiple occasions, especially when the company rewards its workers for reaching goals in important tasks. Personal experience has reinforced this.

3.1 Time Management

Time management is one of the hardest tasks to calculate, especially in the early phases of a production. Many obstacles are just impossible to foresee. It all starts with setting up “milestones”. Milestones are goals expected to be reached within schedule in a production. The importance of the tasks influence how strict the milestones are: some may last weeks; others months. Milestones should be realistically set goals which ensure the attainment of the subsequent milestones up to the final deadline of the production. The planning of a milestone relies heavily upon the expected speed of each team member’s productivity. To give an example: the time investment of creating an animatable character. The time of completion of a single animation can vary from hours up to weeks for it to be finished. This is determined by the complexity and length of the animation. Naturally, the more experienced an animator is the quicker quality outcomes can be achieved.

These are the approximate distinctions for small to midscale productions. Larger productions might have different resources.

- Animations taking hours to complete

Animating individual body parts, simple characters or simple motions

- Animations taking one or several days to complete

These are the animations with intermediate complexity, or common animations like walks; jumps; simple hits; basically actions which are predominantly seen. Walk cycles in this case normally need to be reiterated on multiple occasions.

- Animations which may take weeks to complete

This tends happen to trainees or inexperienced animators. However, normally these happen to be animations with a cinematic narrative. (Companies with larger budgets tend to outsource these types of animations.)

This is merely a brief overview of the timescale of animations to be taken into consideration when setting up a milestone. Budgets, number of animators, technical requirements and the very concepts themselves influence the planning too. An approximate production time must be calculated for every single member of the team.

4 Feedback Session

Regular feedback sessions are a must for the successful functioning of a company in general, not just for the animator. Depending on the company's structure, feedback on tasks can be provided by either the creative leads or by the rest of the team. They serve the purpose of either enhancement proposals, or visual and technical standardization. In most cases it's not feasible to go through every task individually. Regular feedback sessions also provide a valuable insight into the status of the entire production. Any task-related criticisms should not be given with malicious intent, but enhance the quality in order to meet the standards the final project is expected to have.

4.1 Feedback on the Tasks

Feedback on animations is normally provided by co-workers but the decisions should be made by those who are higher in rank. It is the Lead's role to manage their workers. The "common" animator (or designer), when focusing on their work, may sometimes drift away from the artistic vision of the project. Aside from enhancing and updating the visual feedback, it's also necessary to work on technical shortcomings. As mentioned before, not all errors are immediately evident.

For artistic personalities it is occasionally a "harsh dose of reality" when some creative work needs to be drastically changed in order to accommodate either the vision or the technology. The capability of accommodating potentially harsh criticism in this instance is mandatory – in some workplaces, animators and designers need to develop a "thick skin". This can cause a problem, especially in the context of Chamorro-Premuzic (2017) and his description of "Creatives" and their "sensitivities".

5 Software Used

From the inception to the final animation there are many steps to be passed. It happens frequently that different steps and tasks require different programs to work with. There is also a variety of programs with overlapping capabilities. It is normally up for the user and the budget of the production to decide what is used to create the final product. Here is a list of programs used frequently in the Tampere game industry.

-Modelling or Rigging

Blender, Autodesk Maya, 3DS Max and ZBrush

-Texturing

Photoshop, Gimp and Paint tools Sai

-Animation

Blender and Autodesk Maya

-Game Engine

Unity

Indie game developers normally tend to go for Free and open source software's. So Blender, Gimp and a free version of Unity are the programs which are commonly used. Productions with bigger budgets and publishers of course use Photoshop, Maya 3DS Max numerous other programs with for development useful features. Big productions like triple A developers also have access to game engines and tools for their specific purposes and are commonly hidden under strict non-disclosure agreements.

5.1 Various Interfaces

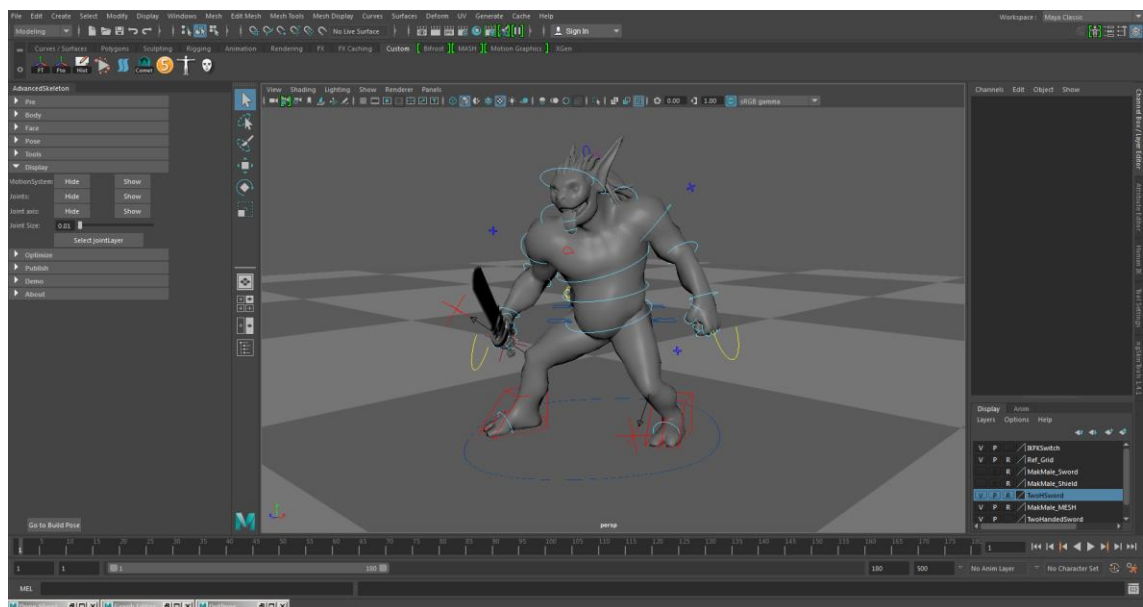
All of the 3D capable programs offer different sets tools to achieve different goals. ZBrush offers tools to sculpt objects similarly to working in the physical world with clay. Maya offers various tool specifically designed for animation and 3ds Max for modelling etc. The interfaces of each program makes the needed tools easily accessible, for a specific purpose. The user must know what he or she needs and choose the program. All that is left is the program-specific learning curve.

Most of these programs have different ways of being navigated. Nevertheless, the logic of using them is very similar to each other. Once an animator masters one program, understanding the others will be much easier. But it is important to stay with one 3D program during the learning curve.

Most companies use multiple different 3D capable programs but it is advised to stick with as few as possible. The reason for that is using multiple software packages with similar functions at the same time may lead to time consuming confusion. Issues with importing and exporting are very common, despite the usage of standardized file formats. Animations and models need to be re-exported on numerous occasion and hindrances like that can cost days of working time.

5.1.1 Maya

Autodesk Maya is one of the more common programs with which 3D animations are made. It is not cheap to obtain for Indies, but it features the most developed tools of the 3D industry in general.



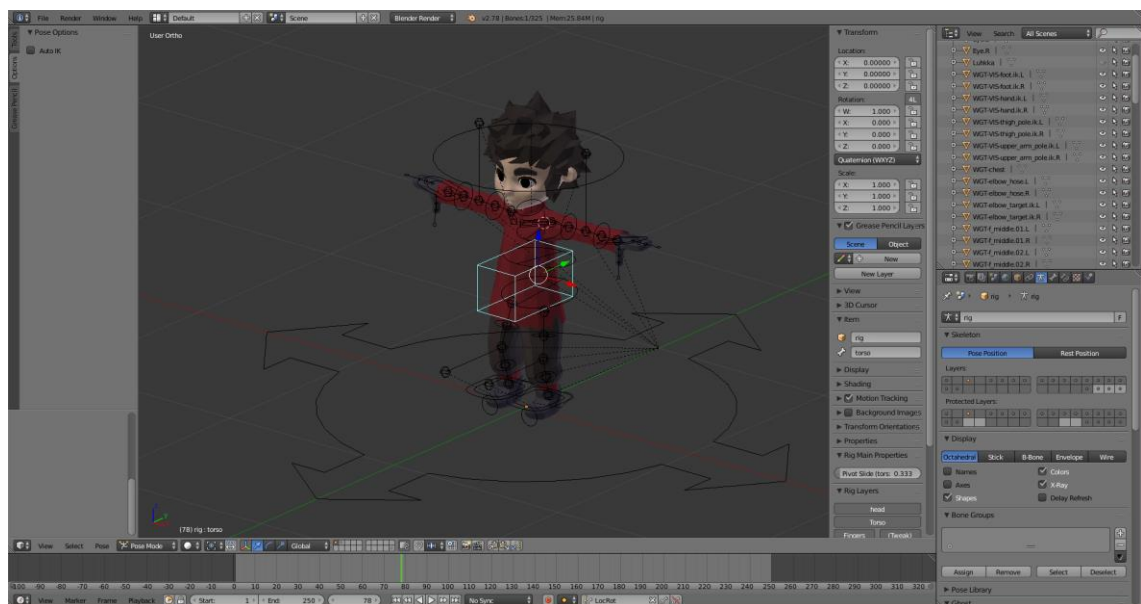
PICTURE 2, Autodesk Maya 2017 Interface (Character: Dreamloop Games Oy 2016)

Picture 2 shows the viewport, the interface and a rigged and animated character. One of Maya's greatest advantages is its easily comprehended interface, paired with the intuitive Viewport controls. Its title of being the industry standard has the side effect that every

other 3D Program follows its methods of working in one form or another. Maya is extremely beginner friendly and finding tutorials is easy since its core features are very streamlined and do not stray away from its purposes. It is probably among the best programs to animate with. But Maya is not a perfect program: it is developed to be a tool for animation first and is designed to accommodate the animator's needs generally very well. However, modelling, rigging, weight painting etc. bare numerous hindrances. Most of these tools tend to be updated frequently, resulting in repeated relearning of their use.

5.1.2 Blender

Blender is free software. Therefore it's widely used for a multitude of different disciplines. It is particularly common among indie game develops due to its accessibility and potential to create quality results. The amount of good quality tools partially outreaches the possibilities of Maya for instance.



PICTURE 3, Blender interface (Character: Red Stage Entertainment 2018)

Picture 3 shows the basic blender interface and view port. Its vast potential comes with numerous disadvantages. Blender has the tendency to be rather unstable. A frequent saving of progress with security copies is mandatory because it tends to corrupt the files that are being worked with. While the vast array of tools blender provides makes it a generally good tool to use, problems arise with its cluttered screen and great reliance on shortcuts. This makes it not very beginners friendly. Blender might offer several different ways to

building up a level useful for games, the basic 3d navigation skills and use of 3d tools are enough, but to create something worthwhile, a coder must be involved. Unity is a great engine to work with but it's not a tool which can be used by one person only. The reason is that unity is a composition software which offers several different tools not just for animations. Additional tools are level designing, sound implementation, adding functions particle effects and many more. Many of those tools Unity offers require understandings specific of the other branches.

6 The Pipeline

“You can picture the 3D animation production pipeline as a car assembly line. Each one does the job in a sequential order to create the entire car in an efficient, affordable, and timely way.” (Beane, 2012, 21). This quote appears to fit incredibly well, since every worker within the pipeline has one or several specific roles which are supposed to interconnect with the workflow of the subsequent member of the team. For example, when an animator receives a 3D model, said model should meet a set of certain technical requirements. It should be textured, weight painted and rigged. Insufficiencies in the former steps can bring the creation of an animation to a stall or a halt depending on the intensity of the error.

However, it’s always better to have every step finalized in the right order. When every step is successfully completed and the animation finalized, it is in the animators’ hands to export the animation with the specific technical additions (if there are any specific additions needed), so that the animation can be implemented into the product.

No one person ever specializes in a singular task of course. A worker may specialize on certain tasks, but it is still within the workers abilities to branch out into different tasks. When a worker is hired, it is up to the hiring party to determine what tasks can be completed by said worker. An animator who specializes in 3D animation is by nature not exclusively experienced in animation only. Everybody who has created a certain amount of animations tends to gain minor experiences in the various other steps and creating an animatable object. One effective way to make the most out of a worker in terms of productivity is to determine primary and secondary capabilities.

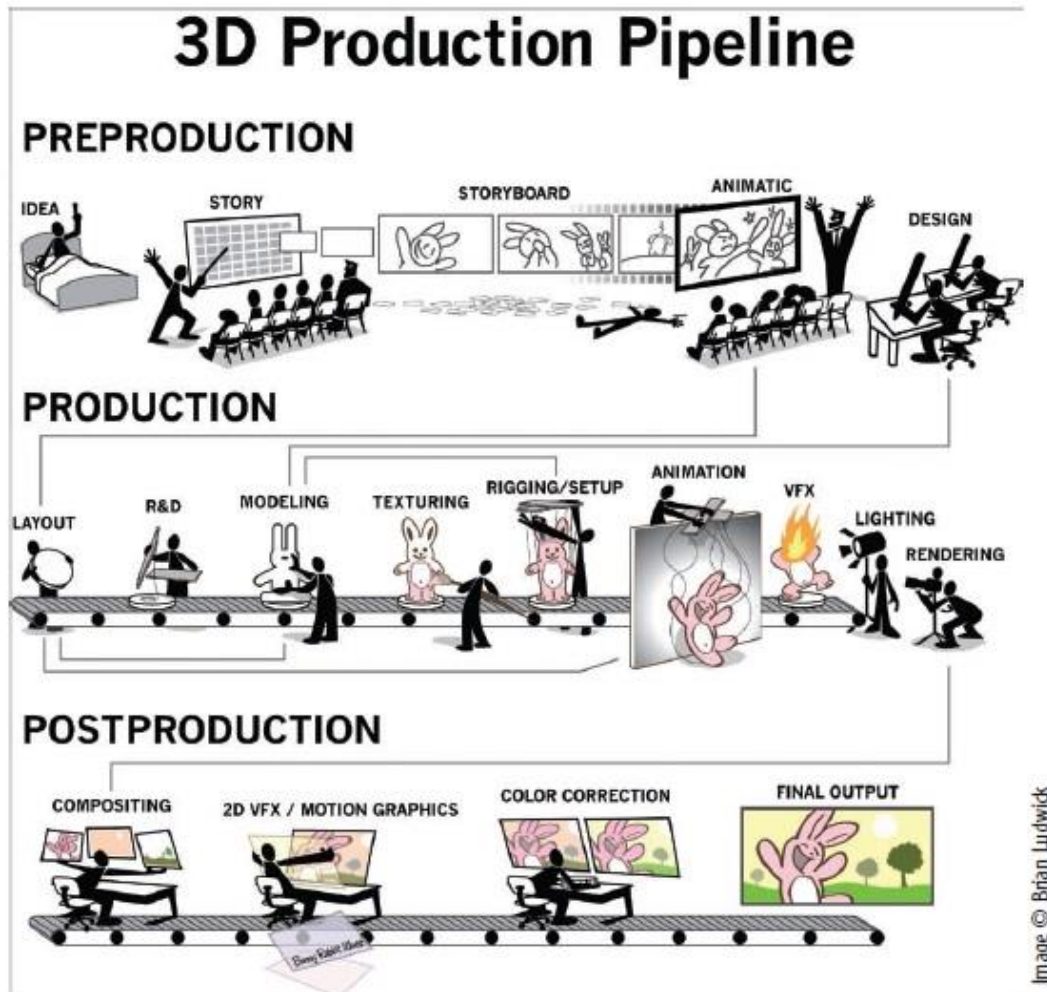
Primary Capabilities

This is the set of tasks a worker is primarily hired to work on. For example an animator will invest the biggest time in a production in animating the assigned characters. Anybody who aspires animation as the main goal will invest their full potential into this goal.

Secondary Capabilities

These include skillsets/knowledge additional to the primary capabilities. The idea is that everybody is capable of doing other tasks. Any seasoned animator is familiar with the

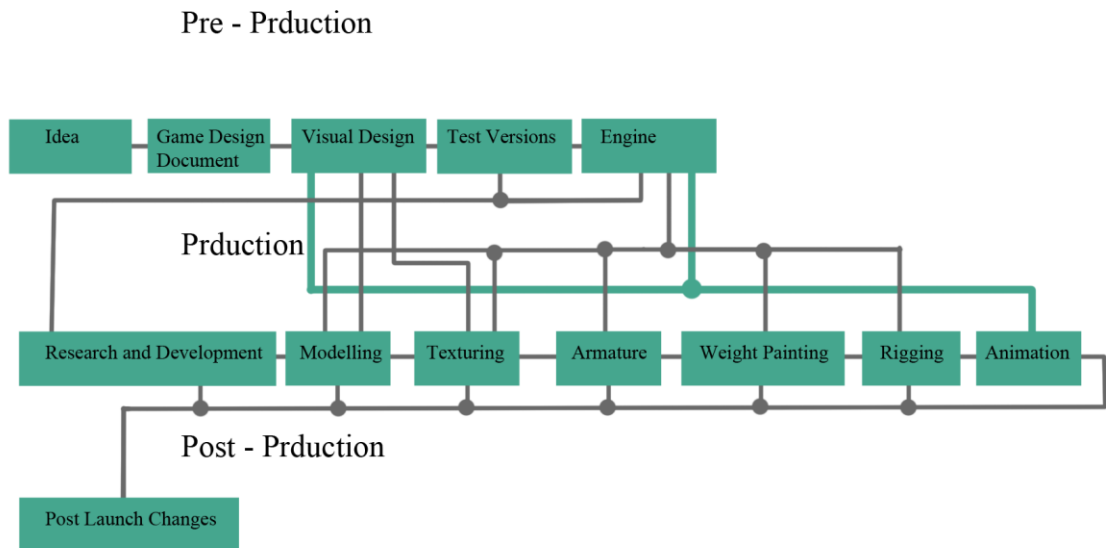
wide set of disciplines required to create a 3D character. These capabilities are rarely completely refined. However when time and circumstance requires the animator to act upon different tasks then it's highly recommended. Versatility is always a positively perceived trait. Once the place of the animator is established, an overview of the workflow needs to be established. Andy Beane wrote in his book "3D Animation Essentials" (Andy Beane, 2012, 23) a very valuable summary. The workflow can be summarized into the three components as it is established.



Picture 5, 3D Production Pipeline (Beane, 2012, 23)

Picture 5 shows the three components with the respective stages. The case depicted here is the production of a full motion video. Game Development or specifically creating animations for video games is applicable into those three components too, but the stages differ to an extent.

Pipeline For Videogame animations



PICTURE 6 show the working pipeline for video game animations.

Picture 6 shows an altered variation of Beanes production pipeline, the alterations show steps more typical for video game animations. The graphic shows what steps are placed in which part of the production and what steps are building on top of one another. It is important to note that throughout the entire production the technical and visual aspects are the most important steps for the animator.

- Pre Production

In its essence the pre-production stage should establish the basic concept of the project and its required tasks. It is in this phase where the artistic properties and technical specifications need to be worked out.

Idea, Story, Visual Design (creation of the Game Design Document), creating early test levels and tech demos these things are the parts that make up pre-production.

- Production

During this phase the amount of assets is to be produced.

The components that make up the production phase are Research and Development, Modelling, Rigging Animation and Implementation. Designing the Levels which all the assets and content (The major part of the production).

- Post Production

The final stages very often feature changes and last minute additions to the final product. Such as play testing, doing last-minute changes and post launch game development (a different topic all together). As is made apparent here, the biggest share for the animation relies within the Production stage. Generally speaking this is also the stage where the first tier of tight schedules should be in, because it is of great importance to get the basic required set of animations ready. The basic set refers in this case to the common in-game animations such as walks, punches, and jumps of the main character, also including those of side characters and enemies. Lengthy, narrative-oriented animations are also created in this phase.

The second tier of tight deadlines happens in post-production usually after receiving play testers' feedback. Play testers are groups of people not affiliated with the content creation. Their main goal is to play test versions of the games, to evaluate the visual cohesiveness and technical finesse. Often they give feedback on errors not obviously visible during the production. Occasionally it happens that a "post-launch phase" needs to be initiated. It is a phase when content needs to be added or changed based on feedback received from the test players. Those changes are very necessary changes on issues which weren't apparent during the course of development.

6.1 Research and Development

"In R&D a team of artists from many different components work with technical directors on upcoming technical challenges." (Beane, 2012, 17). That what Beane is referring to movie production but it shows similarities to videogame development. Usually this is among the first stages where it is determined what approaches and methods will be used in order to reach certain goals. Goals are set determining the level of quality of textures,

the complexity of the 3D models and the assigned rigs gaining an approximate idea about the number of animations that will be needed.

A simple rule of thumb is if the final product has multiple characters displayed on screen - as in strategy games - then it is advised to keep the complexity of required models rather low. On the other hand, a story driven game should usually have models with more complex structure to accommodate the requirements of the narrative. Making more expressive characters always requires a high level of complexity. This is also the stage where technical limitations are discussed. Not every type of animation can be technically achieved. The technical team should be frequently consulted about the possibilities of realising the final product. However some issues can and will occur in a later stages. Pre-planning a potential research and development round in a later stage of the development is recommended.

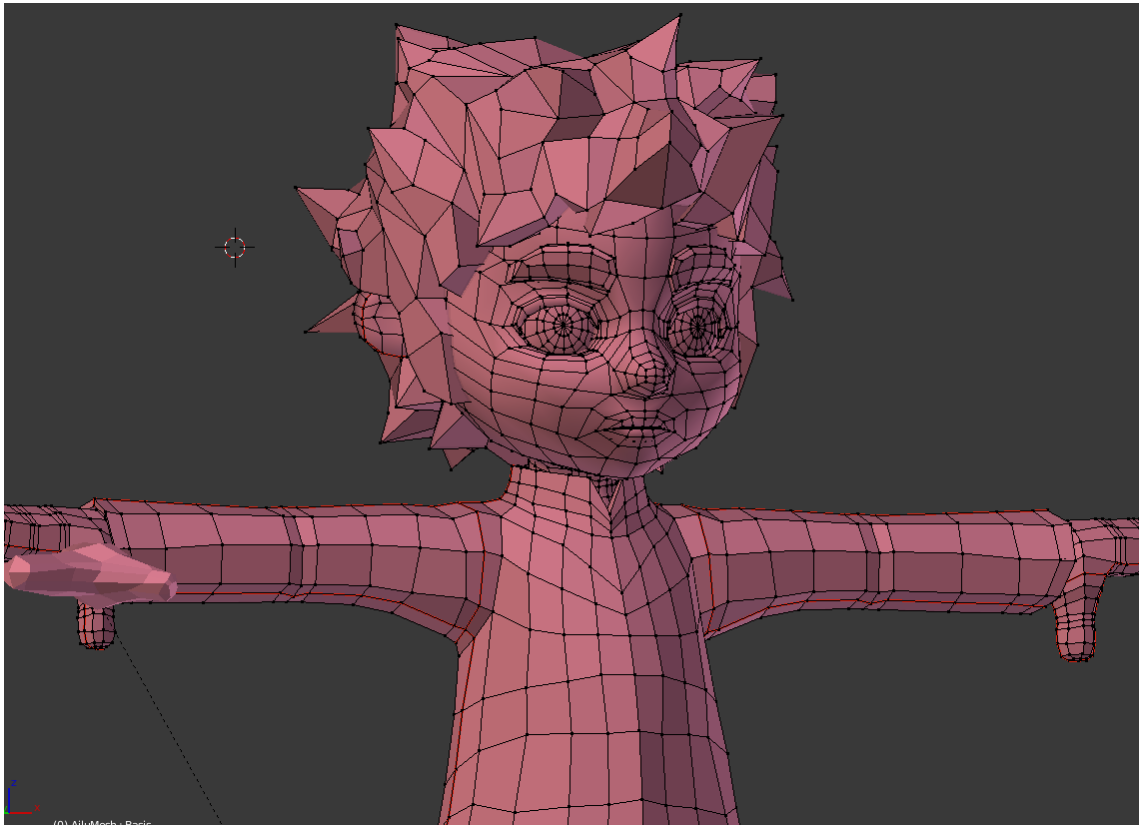
6.2 Modelling

“A model is a geometric surface representation of an object that can be rotated and viewed in a 3D animation software package.” (Beane, 2012, 37). As a follow-up to Beanes quote, modelling is the creation of such geometric surface representation. There are several programs used to create models and each one offers different tools to reach this desired goal. Some of the more commonly used tools are Mudbox and ZBrush which offer tools for sculpting models, however those tools require an extra step afterwards to decrease the amount of details to a more engine-ready level. (The extra step is referred as re-topology, essentially modelling based on a high poly model.) Blender, Maya and 3Ds Max are the tools where modelling is made via directly altering basic shapes by adding more detail on them.

“Although there is a fixed set of basic tools used in polygon modelling there are several possible approaches.” (Mullen 2007, 32). As Mullen rightfully suggests there is a variety of different approaches, however it is crucial within a team to stick with one approach. The reason is different approaches often require altered adjustments of the tools functionalities. User-defined functionalities often don't interconnect well with each other. So it's vital to agree upon the settings early in the production. The creation of a three-dimensional geometric structure which features an aesthetic that is unique to the game is not only meant to achieve the intended visual design, but also a few technical necessities.

Modelling is not exclusively referring to playable characters. It also includes the creation of any other three-dimensional shape needed for a game, like weapons, environments, objects etc., some of which can be also animated if needed. However, in order to create a model which is to be animated it must feature a set of properties.

Topology



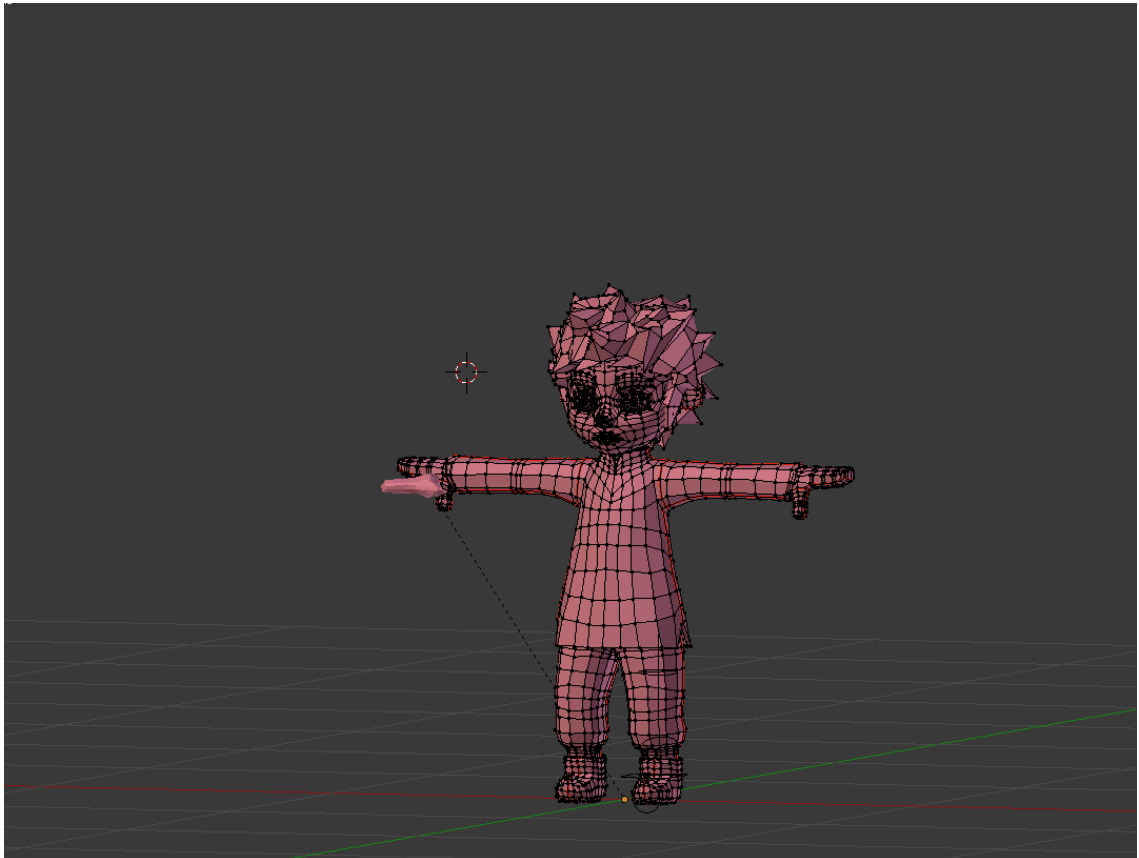
PICTURE 7, Topology of a video game character (Character: Red Stage Entertainment 2018)

The model must consist only of square shaped polygons. “A polygon is typically the surface between four edges.” (Gahan 2011, 27). Picture 5 collates with Gahan’s quotation, It is important that almost all polygons must be squares with four vertices and four edges. There are occasional situations when vertices require to be connected by more than four edges. Even in this situation a polygon must be square shaped that also may cause triangular polygons. Certain methods of modelling round objects can have triangular polygons. This very specific surface layout is important for the later stages of the production. All those polygons should follow a basic tube shape. Exceptions as shown in the picture

5: hair, facial structure, and shoulder hinges can have multiple vertex points but they still must retain the polygonal shape.

Position

An object's origin marks the location of Model in three-dimensional space. In the case of Blender, the object's origin is visualised by an orange dot.



PICTURE 8, the Position of the Model on the Y Z Axis. (Character: Red Stage Entertainment 2018)

Picture 6 shows the proper placement of a model in the three-dimensional space.

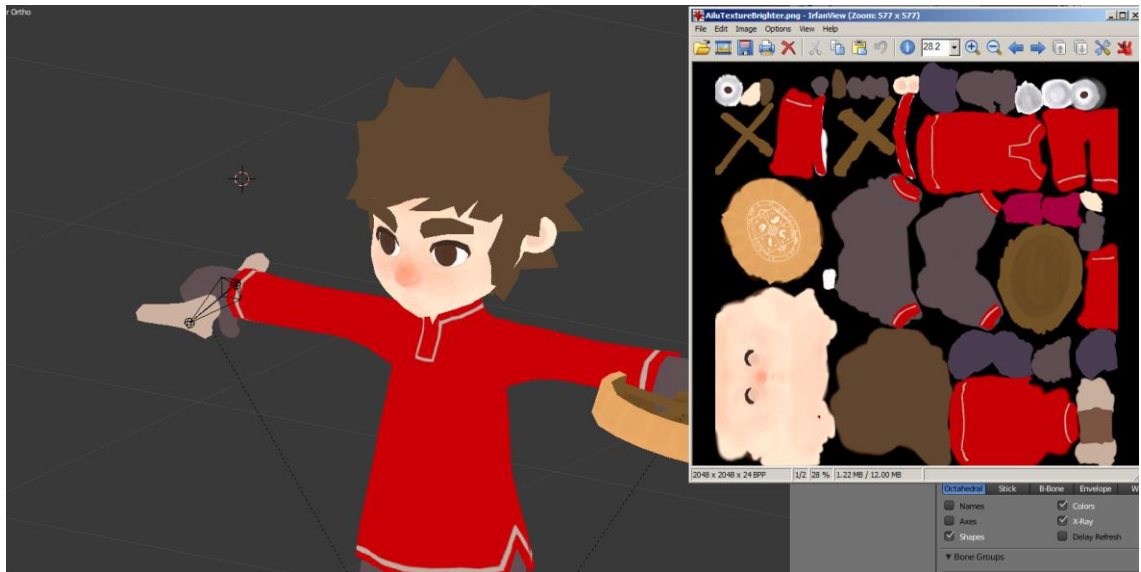
The placement of the point of organ must be at the 0 point of the x,y and z axis. The models lowest point in this case the bottom of the feet, should be grounded while the rest of the model protrudes upwards.

Objects

There are many situations in which the model features numerous separated objects. The most common segmentation of objects to be found are eyes, hair and weapons etc.: parts which are connected to the main character but not necessarily part of it. It makes it easier to select those parts during the weight painting phase. Segmented objects also tend to be devoid of any deformations aside of rotation movements.

6.3 Texturing

“The Terms Textures and Texture maps refer to 2D images that are Projected onto or wrapped around a 3D mesh.” (Gahan 2011, 26). Texturing is the phase of production when the Model is getting one or several two-dimensional planes applied to it. Set images have the intent to represent the skin colour, fabric, metal, etc.



PICTURE 9, The basic Texture on the Model and as the separated file. (Character: Red Stage Entertainment 2018)

Picture 9 shows the first function of the texture, which is to give the model a basic color tint, and skin properties like freckles and the reddish nose. The second function is to show material properties which don't need to be created on the model itself. These can be scratches, scars, rust, and ripples - basically anything which appears to be flat from several angles. This helps to further enhance illusion of realism without increasing the topology.

Besides the basic texture, there are numerous other types of 2D images such as alpha maps, diffuse maps, and bump maps. Those images files usually help to increase shadow effects and fake a roundness of otherwise flat surfaces. Those additional maps however are topic of less relevant nature for the animations: they tend to be more a part of the art and technical departments. In addition, viewports in animation software tend to lag when complicated texture systems are set up.

6.4 Armature

“An armature acts like a skeleton: you actually move bones of the armature and those bones drive the animation of the mesh,” (Hess 2007, 321). What Hess suggests is that setting up an armature requires basic knowledge of anatomy. Even though armature bones follow different logics than their anatomical counterparts as Hess suggests, the logic of the positioning is interconnectable. The complexity of the rig also varies a lot depending on the needs.



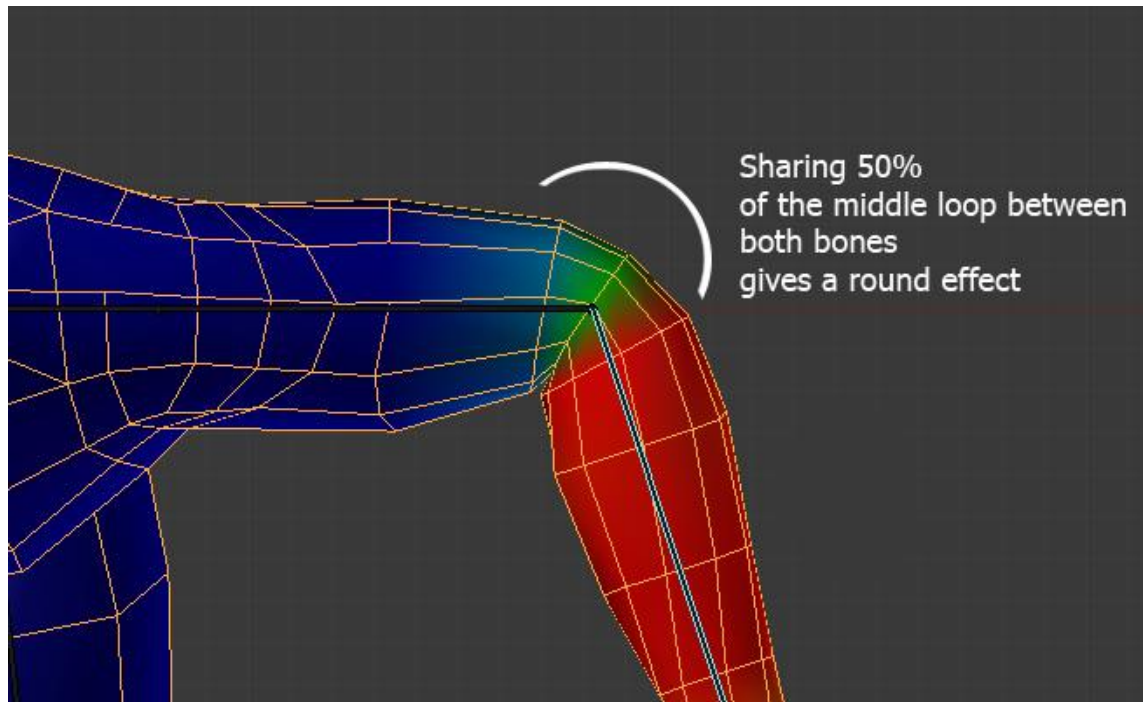
PICTURE 10, Placement of the armature in relation to the model. (Character: Red Stage Entertainment 2018)

Picture 10 shows a very simple Rig for a gaming character. Note that all of the bones in the arms and legs are mostly placed on those spots where anatomical bones would be, and also, that all bones feature a specific directional hierarchy shown by the tip parts of each bone. It means if the hip bone is rotated, the lower hip and foot follow accordingly (in the same vein as forward kinematics work). Particular attention needs to be given to the spine in this instance. Normally the needs of the upper body are discussed beforehand. Only 3 – 4 spine bones (neck and head excluded) are to be set up for decent and partially exaggerated movements. However if more naturalistic motions are required, then the number of bones can also be increased up to 6 bones. If the character has tools or weapons those require additional bones.

6.5 Weight Painting

Once the armature is successfully set up the following step is usually weight-painting.

This term describes the assigning of influences from the bones to the model's polygons. The idea is to have the model deforming according to the location and rotation of the bones in the most natural manner as possible. In the example of an arm bending at the elbow, the intention is to have the bone of the upper arm moving only the upper arm and the bone of the lower arm only the lower arm. The crucial place is the joint part where the elbow movement occurs.



PICTURE 11: The influences of the bones on the model. (Blender Foundation and cg.tutsplus.com 2012)

Picture 11 shows the red coloured fields of influence from the lower arm bone. Now the green part shows a diminished intensity of influence. This effect is desired and strived upon in the weight painting phase because it further enhances the chance of natural poses even further. Every single bone in the armature must be weight-painted individually in order to create the best looking outcomes. Segmented objects of the models must be weight-painted individually. Every program which features weight painting also offers numerous automatic systems to automatically assign weights to their respective bones. Those systems however are still not perfect and the outcomes they produce still require manual adjustments. On average, if the automatic weight painting goes smoothly, the time needed to make further adjustments is significantly shortened.

6.6 Rigging

Once the armature is set up properly, the rigging is the next step. An armature on its own has no rules nor physics and can be placed randomly in any position, in a way which is very reminiscent of shadow puppets. But natural movement is sometimes restricted by physical properties, for example the direction in which an elbow bends. To avoid this situation where the elbow bends the wrong way, a control rig must be set up. Normally this is represented by nurb shapes placed at any important joint requiring movement. “It’s important to create animation controls that are both logical and easy to use while providing sufficient detailed control as needed.” (Montgomery 2012, 1). Montgomery suggests that it is a traditional workflow of setting up controllers to place a nurb with its center point at the same spot as the bone it drives. Then the bone is connected via parenting where the control Nurb moves are followed by the bone.



PICTURE 12, Controlling Nurbs alongside the model. (Character: Red Stage Entertainment 2018)

As it is seen in picture 10 when control nurbs are set up the armature beneath is normally made invisible to decrease unnecessary shapes on screen. The end goal of the rigging is to create a simple-to-understand movement control system. One important aspect for the rigger and animator is to note that all animations should occur either via translation or rotation; scaling doesn't work for most gaming engines. To avoid accidental scaling, Maya and Blender normally offer a locking system.

6.6.1 FK – IK

When an armature is to be animated, it normally follows two distinct methods: Forwards Kinematics and Inverse Kinematics. These two movement methods are the only ones the animator should be concerned about since all the mathematical functions are hidden beneath a user-friendly interface. Of course a deeper understanding of the program specific functions is beneficial for proper use. But for the most part, it's of a lesser importance for the animation.

Forwards Kinematics

“Forward Kinematics always uses rotations for animations. Translations or scaling of the joints in the armature chain would scale and deform the arm unrealistically.”(Montgomery 2012, 51). Montgomery refers to complications which arise when the armature is scaled during an animation. It is suggested to avoid scaling all together for in-game animations unless it is specifically required. This process is sometimes referred to as “Puppet animation”, because it features a similar logic to that of moving a puppet: each bone can be rotated into the intended position and every subsequent bone in the hierarchy follows. Bones of an armature are organized as chains, ‘Forward’ in this instance refers to the direction into which the bones are pointing. Parent bones rotation affects all the bones that have child rotation to the bone.

Inverse Kinematics

“In all 3D animation an IK setup is created from a basic joint chain.” (Montgomery 2012). IKs, as Montgomery suggests, are chains of joints which calculate the rotations of every joint within a chain to accommodate the positions of the beginning and the end of the

joint. It creates the opportunity of taking control of several bones while changing the positional values only of the last bone in the IK chain. IK is most commonly used in arms and legs. The reason is it provides a quick and natural looking positioning of limbs. IK bone rotation may affect its parented bones rotation values. This takes place in the opposite direction in the chain, compared to FK. Thus the name inverse kinematics.

IK FK Switch

An IK – FK switch is a system which can be keyframed and enables a smooth transition between both forms of kinematics. The application for a switching phase is to create a smooth transition between an IK controlled joint chain into a FK controlled joint chain. If for instance a jumping character is to be animated, one fitting spot to switch the limbs from IK to FK is during the airborne phase, where the limbs can move more freely. Setting up IK and FK switch is not that complicated for an experienced rigger, however, creating a nurb controller can cause complications. In numerous occasions the switch controllers aren't made in the first place, instead the switch is placed a for the software specific tool bar or properties section.

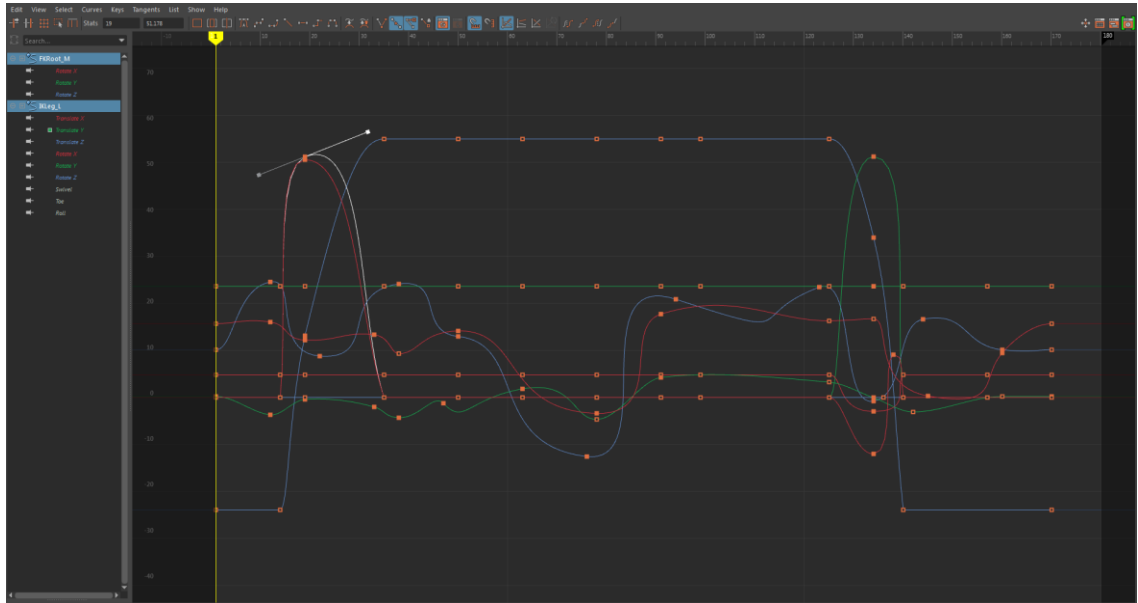
6.7 Animating

Once the character is rigged and weight-painted properly, animating it is the subsequent step. The animation process itself is theoretically rather simple. The idea is to move the character into a desired pose, then a keyframe has to be made, followed by a transition of the marker on the timeline to a later stage where the character is posed anew and keyframed again. By default, only control nurbs should be keyframed. Bones in the armature can be keyframed too, but that usually leads to issues in the subsequent animation process such as “ghost keyframes”. Ghost keyframes are keyframed motions which are not notified in the keyframe timeline, dope sheet nor graph editor. They can be caused by user errors, input malfunctions or glitches.

A keyframe in this context refers to the saving of positional and rotational data in a certain position on certain point in the timeline. Scaling should not be keyframed unless it's explicitly discussed with the person or people responsible for the implementation into the game's engine. Placing the keyframes on their own leads to crude-looking results which is good for creating the basic idea of animations. There is a way to make the animations

look more refined by placing more key frames in-between major movements. There are two tools suitable for the process of refining which are provided by blender and maya: the graph editor and dope sheet.

6.7.1 Graph Editor

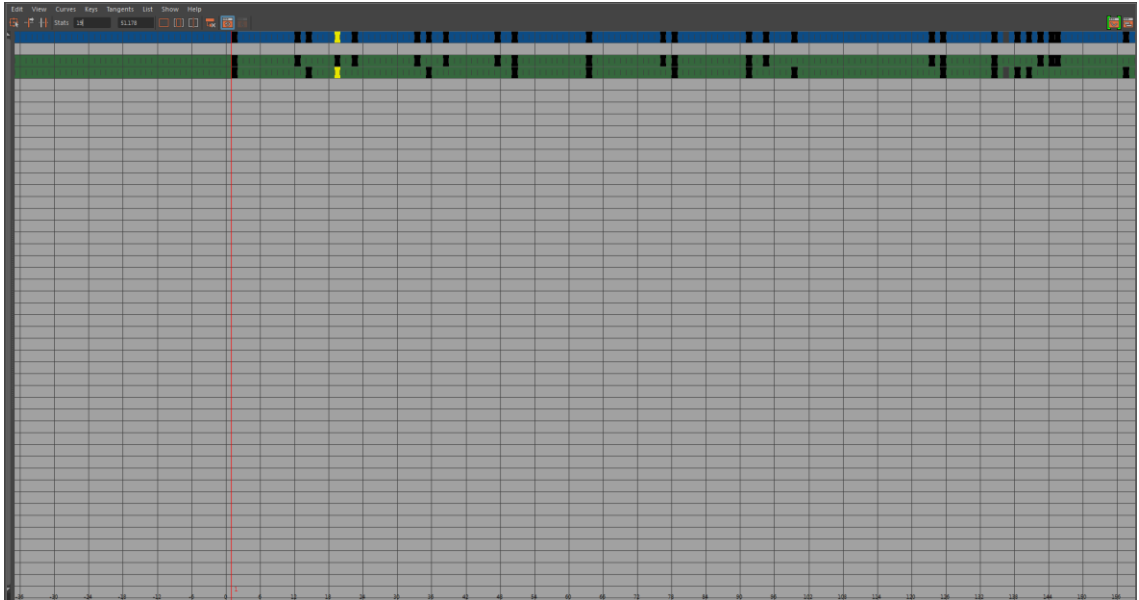


PICTURE 13: The Line Graph of Autodesk Maya 2017

Picture 11 shows the line graph of Maya. It is duly recommended to have it open at all times as a second screen during the animation, because it gives a vital secondary representation of the trajectory of the animation. “In the Maya Graph Editor, we can see a visual graph representation of motion for scene objects.” (Montgomery 2012, 15). To clarify Montgomery’s quote, when the control nurb features several animations already, in normal circumstances there will be nine line graphs shown: three for translation, three for rotation and three for scaling. All of them are colour-coded in either red, green or blue. The color coding refers in this case to the X, Y and Z direction. Each of these lines gets a path point when keyframes are made. These points mark the position and intensity in of the motion imposed on the selected nurb controller. In addition these nine main line graphs, there could be more line graphs shown in grey: those refer to other keyframable functions like IK-FK switching phases. The visualized graphs are infect vector shapes and thus can be used like vector lines similar to adobe illustrator. The access to and rela-

tively easy-to-follow interface enables a fairly sophisticated array of adjustment possibilities i.e. if there is the need to intensify or weaken a swinging arc by manipulating the directions.

6.7.2 Dope Sheet



PICTURE 14, The dope sheet of Autodesk Maya 2017

Picture 14 shows a simple representation of Maya's dope sheet. Similar to the graph editor, it enables the possibility to adjust the keyframes. However, it does not provide the adjustments of the intensity of motions, only its position in the time-line. It is a visual representation of all the keyframes associated with a control nurb. But what it provides is an easy method of adjusting the timing of various actions of the selected item. It is a perfect tool for blocking out animations and should be used in the beginning stages of animating a character.

6.8 Principles of Animation

The Basic Principles of Animation are designed to create the illusion of characters moving in physically plausible-looking ways. In order to achieve this illusion a set of rules were set up by nine animators of Disney in the 1930s (Johnston and Thomas 2008) in the book *The Illusion of Life: The Principles of Animation*. The Principles of animations are explained in detail and used as bases for numerous animators. Despite being created for hand drawn and 2D Animations, all the twelve principles of animation are applicable to 3D animation. Most (if not all)

feedback is related to any of the 12 principles in one way or another. Technical difficulties and aesthetic requirements may influence the animations however those discussion must have been made during the pre-production phase. It is therefore important for the animator to comprehend them all. The reason for that is that humans follow certain movement laws in order to distinguish a “lifelike” looking motion from a “mechanical looking motion”. To put it simply, animation production technology has changed but the human mind has not. Video game animation relies heavily on all twelve principles. These principles are:

-Arcs

Every realistic motion happens on a curved trajectory called arc. It is one way to create the illusion of gravity on the moving object.

-Anticipation

Anticipation is the principle of creating an expectation for a certain motion, usually by starting off with a slight retrograde motion which transitions into the actual movement.

-Timing and Spacing

Timing is a method of determination when something imitates a move. For instance, if the body begins to move before the limbs or other way around.

-Ease-in and Ease-out

In nature no movement is constant, it always varies in speed. Normally it works that the movement accelerates to the maximum speed (which is the intended motion) and then decelerates at its end.

-Staging

This principal is less related to 3d animation itself but more of a priority system for the individual pieces which are to be animated. Depending on the type of the game the animated characters are normally seen from a specific angle. Staging in this context is from

which angle is the animated character seen, this makes it apparent which parts of the character need to be animated.

-Appeal

Like staging appeal is less of a principal but more of a guideline. It is the ideal situation for the character to look appealing to the viewer. Essentially it requires that the animated character moves in an interesting way.

-Secondary Action

Secondary actions are usually additional movements to the main movement, e.g. when a character throws something with the left arm. The animation on the right arm would be secondary.

-Straight ahead action and Pose-to-Pose

Straight ahead is a method to animate in which a movement is created from beginning to end - a quite straightforward approach.

The pose-to-pose method is more commonly used in 3D animation, because the blocking out can much be achieved much more quickly by placing and switching the poses within the required times. After the poses are complete, all in-between animations can be worked out.

-Drawing Skills

“Drawing should become second nature so that the animator can concentrate on actual actions and the timing of them and give the performance life.” (Williams 2001, 23). Williams spoke from the point of view of a traditional animator. However his suggestion is very fitting for modern processes. Even though everything these days is done digitally, some of the traditional workflows have their use too. Sketching out thumbnails of motions the character is supposed to do has proven itself useful during the planning stages.

-Squash and Stretch

This principle is designed to show flexibility of character or object. Usually it is shown in a deformation of usually “solid” parts on a character. Extreme utilisation of stretching and squashing can be used to increase comedic value.

-Follow through and Overlap

Very often called “offset” in traditional animation, it is the rule which states that movement doesn’t happen simultaneously. The swing of an arm while walking never happens at the same time as a swing of a leg.

-Exaggeration

Exaggeration is necessary to show a certain movement more clearly than nature would do. a good example are strategy games where the characters actions are shown exaggerated mostly so that they view can see and register them.

6.9 Difference between In-game and Cinematic Animations

Cinematic animations are in their very nature designed to follow a strict narrative of interconnected and potentially long lasting motions. Normally it requires a complicated rig, with controllers allowing subtle and precise adjustments on the parts which are intended to drive the narrative forward. Facial animations need intricately designed controllers. Even though animations need to be seen only from one point of view, the typical process of creating those animations lasts much longer than the game animations by comparison.

In-game animations which are made specifically for gameplay related purposes tend to be short and segmented animations. The reason for these short segments lies in the interactive nature of videogames, where the player can more or less randomly decide when an action is triggered. For example, letting the player character run for a while, then slash some plants or enemies, and run some more to press a button. Depending on the style of

the game it is also necessary for the animations to look good from several different angles. The production of in-game animations is much quicker than their cinematic equivalents, but they also have to be revisited more frequently. Doing minor changes and reiterating is a common occurrence. There are cases when video games trigger engine-rendered cut scenes which tend to follow a narrative-driven set of animations in a cinematic fashion. They are used with models and rigs reminiscent of the simplistic rigs of the playable characters. A decision to do a cut scene with the in-game models must pass the following three criteria.

1. Does it work within the game engine?
2. Does it drive the narrative better?
3. Is there time to make something more complicated?

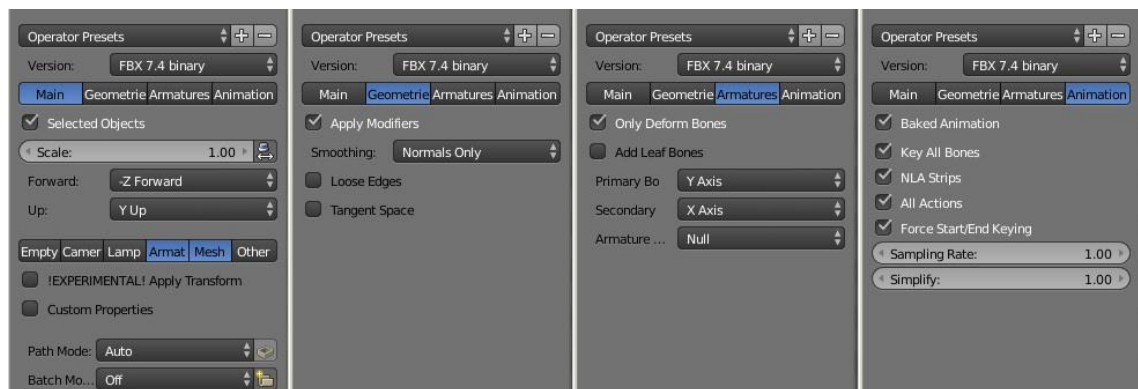
The first criterion is to make sure that the intended animation is compatible with the engine. The second criterion is make sure that the narrative content of an animation is enhanced with its usage. And the third criterion if creation of a cutscene can be created in the first place. Hindrances may be a lack of time, lack of experience or incompatibilities with the engine.

7 Implementation

When the character is fully rigged, modelled and animated the next step is to bring the animated character into the chosen game engine to work with. In this case there are basic steps which the animator needs to follow.

7.1 Baking

Regardless of the software being used, when progress is saved, Each software uses its own native file format (*.mb for Maya and *.blend for blender etc.). All of those formats feature storage methods and functions compatible exclusively with their respective software. The method of transforming the native file formats into more commonly used file formats is often referred to as ‘baking’ or ‘exporting’. Proper adjustments of the baking options are mandatory. Among indie game developers the most commonly exported file format featuring rigged, weighted and animated models is *.fbx. The reasoning behind this is that all major modelling and animations software are capable of exporting in this format and most game engines are capable of importing it.



PICTURE 15: the different settings that have to be made for a Unity-compatible file.

Picture 15 shows numerous adjustments that have to be made in order for the fbx to feature only the needed data. Which is:

- The armature without the control nurbs.
- The model with the weight paint data
- Animations whit the respective storing system.

If the exported fbx needs to have any other data baked too then it should be discussed with the person in charge of implementing the exported file. It's important to note that the axes' positions settings have to be treated with special attention. Every program which

features a 3D capable interface has their own strands for the x,y,z coordination systems. Therefore it's recommended to set it up as a company internal standard early on in development.

7.2 Implementing

After completing the baking the next step is to implement the exported fbx into the game engine. This step is completed by either automatic or manual implementation. The means the animations are brought into the system are of lesser importance for the animator. What is important however is a mutual understanding between the animator and presumably the coder. Even a simple character could retain from 4 – 9 separate animations which can be partially or entirely interconnected with each other. All of this information needs to be agreed upon in the beginning stages of the development. Whilst in smaller teams a more personal communication is possible, larger companies tend not to have a benefit like that. One of the most commonly used methods of communications in the implementation phase are naming conventions.

7.3 Naming Conventions

“File Naming Conventions help you keep track of your work.” (Lumsden 2010, 287). Lumsden referred that a naming convention is a tool to help the asset creator to keep track of the progress. But a companywide naming convention helps to keep track of the entire state of progress. Everything which is created in the company - not just one subsection - needs to be properly named. Names should already clarify the precise plethora of information on the properties a file contains.

Anim_Mak_M_Lt_ALK_Atk_N.fbx	3.11.2016 14:01	FBX File
Anim_Mak_M_Lt_Bow_Atk_N.fbx	2.11.2016 11:57	FBX File
Anim_Mak_M_Lt_Run_N.fbx	8.12.2016 11:37	FBX File
Anim_Mak_M_Lt_Run_N_01.fbx	7.12.2016 16:02	FBX File
Anim_Mak_M_Lt_SngSwoSh_Atk_N.fbx	25.8.2016 11:58	FBX File
Anim_Mak_M_Lt_SngSwoSh_Atk_N02.fbx	2.11.2016 11:46	FBX File
Anim_Mak_M_Lt_SngSwoSh_Idl_N.fbx	11.8.2016 13:32	FBX File
Anim_Mak_M_Lt_SngSwoSh_Mov_N.fbx	2.11.2016 11:32	FBX File
Anim_Mak_M_Lt_SngSwoSh_Trn_L90_N.fbx	2.11.2016 11:51	FBX File
Anim_Mak_M_Lt_SngSwoSh_Trn_L180_N.fbx	2.11.2016 11:53	FBX File
Anim_Mak_M_Lt_SngSwoSh_Trn_R90_N.fbx	2.11.2016 11:54	FBX File
Anim_Mak_Run_End.fbx	2.11.2016 11:47	FBX File
Anim_Mak_Run_Start.fbx	2.11.2016 11:44	FBX File
Anim_Mak_Slash_2Handed_Sword.fbx	16.11.2016 18:06	FBX File

PICTURE 16: The naming convention animation files. (Files: Dreamloop Games Oy 2016)

Picture 16 shows a screenshot of one of the archives with properly named fbx files. To clarify these terms:

“Anim” Stands for Animation which clarifies the type of content the file features without indicating what type of file format it is.

“Mak” stands for “Makapoa” Name of a Species of Creatures for a game. Already indicating that there is more than one species.

“M” stands for Male. This indicates that there is sub category of Females too.

“Lt” stands for light. Indicating that aside of light there is mid and heavy weight class.

“Alk”, “Run”. “SngSwoSh” Indicating the singular properties of either Alchemy, Running or sub category of swift sword swinging.

Every space within a naming convention must be written with an underscore. It is because automatic systems don’t work properly with empty spaces in names.

This detailed information already mentions a large amount of singular animations in that project. This is merely the example of archiving animations. Different instances of the development process also have their own respective naming conventions.

7.4 Testing

When everything is in order, the properly named fbx file is implemented into the game engine. The next step is to test if the file really stands up to the expected standards. Every step up until now features individually a great amount of work from multiple people and occasional errors in any of the processes above may have happened. Testing is mandatory. The amount in need to be tested varies in different parts of the production. But in the case of Animation only two major parts need to be tested.

7.4.1 Does it Work?

First and foremost the naming convention. If the project features a huge influx of assets, implementation often happens with automatized systems. Errors in the naming convention can lead to an improper implementation. Another very often recurring issues are improperly baked fbx files. Even though all of the options and set ups and are all right, the baking itself may not work correctly. Alternatively, some user errors might have been occurring during one of the multiple aspects up until this point.

7.4.2 Does it Look Good?

When everything from the technical side works properly it is important to check if the instated animation meet the visual requirements. This is a crucial phase because all the prior mistakes can be seen. Modelling and texture mistakes can be easily spotted by just looking at the character, armature and rigging errors can be seen by misshapen and jugged motion. Especially for an animator, it is here when all of the principles of animations are reviewed. It is also here when is determined if errors of some of the former steps have to be revisited, or if those mistakes can be solved with some of the game engines properties.

The reason for the drastic change of the outlook is the way that the game is rendered. 3D modelling programs very often have a simplistic method of rendering the 3d model as it is seen on screen. A singular light source with no visually enhancing processes, flat colours and no colour correction. All of those processes are commonly very Very detrimental to computer performance and create hindrances in the rigging and animations process.

Game Engines, on the other hand, are designed to have the above mentioned visual enhancing systems.

8 Conclusions

When I started my career as an animator for video games, I came from a different level of understanding. Like so many others, I misunderstood the workflow drastically. This mistaken assessment was based on the expectation of the field being far more artistically independent with offers of creative expansion. In reality it became quickly apparent that a significant amount of time is devoted to fulfilling someone else's artistic vision (in the early stages at least). And whilst at the time it felt like great disingenuousness of circumstances, soon after, the realisation struck me that even in the fulfilling of someone's vision there is a need for creativity. The creativity was not solely related to the creation of the intended animations, but also added human communications and technical solutions into the mix.

Working in numerous projects has been particularly insightful. Even though a certain amount of experiences were made in smaller projects, the best experiences were made longer lasting projects. The first and most vital experiences I made were as rigger, weight painter and animator by using autodesk maya for Skydome Entertainment. I had the opportunity to take over several different jobs and could directly see how the aforementioned tasks have been influencing my efficiency and quality of the animations. It helped me to get a grasp on how tasks are delegated and how to retain the desired visual aspects. The second set of valuable experiences I have made at my work for Dreamloop Games. For the most part I was assigned as an animator. The project at the time required a huge amount of animations, rather quickly I had to develop skills for creating decent animations in a short amount of time. In addition, during the time I was employed at Dreamloop Games a merge with another group occurred. Interesting challenges arose when the decision was set up to switch from blender to Autodesk Maya as well as developing a common of level communication with the new members. The third, and for this thesis, last relevant set of experiences have been made at Red Stage Entertainment. For this company I was hired to create animations in blender. Taking into account the experiences I made with the aforementioned companies gave me the opportunities to develop communication skills needed to take on a more leading role.

The sum of the working experiences and my background in fine arts have been the inspiration to put this thesis together. It is very common that people with a fine arts background to start doing game development with the intent of creating "great art". It is so because

games tend to mislead the player to believe that the creative process is equally simple. The truth however is different. When an artist truly desires to be a game developer, then alongside with developing a certain level of communication skills a set of technical skills needs to be developed. It is not an easy task to partake in seemingly boring or infuriating learning sessions, however, diligence pays itself off.

Historically, animation was always about shaping technology in order to help to tell stories. It wasn't any different with the old masters who did animations with traditional media as it is with the utilization of modern technology. In many respects, creating animations for interactive media such as video games is a seemingly logical evolution of those early forms of animation. As a final thought, it is easy to say that appreciation of the provided technology should be a basic requirement; also, the capability to communicate and understand one's own position with a team in general. And finally the mental preparation for the fact that problems are always there and in need to be overcome in one way or another.

9 References

Activision Lead game play animator Application 2017. Read 28.3.2018.

<https://activision.referrals.selectminds.com/jobs/lead-gameplay-animator-2272>

Beane A. 2012 “3D Animation Essentials”, USA John Wiley & Sons Inc.

Bioware Creative Director Application 2018. Read 28.3.2018.

<https://career4.successfactors.com/sfcareer/jobreqcareer?jobId=118416&company=EA&username=>

Byrnes W. 2003 “Management and the Arts” USA Focal Press

Chamorro-Premuzic T. 2017 “Working With Creatives: A Short Guide For Everyone Else”. Read 27.3.2018.

<https://www.fastcompany.com/40484057/creative-people-are-hard-to-work-with-heres-what-to-do-about-it>

Gahan A. 2011 “3DS Max Modelling for Games second Edition”, USA Focal Press.

Hess R. 2007 “The Essential Blender”, Netherlands Ton Roosendal Blender Foundation.

Johnston O. and Thomas F. 2008. “The Illusion of Life: Disney Animation”, USA Abbeville Press.

Lumsden C. 2010 “Computer Animation Complete”, USA Morgan Kaufmann Publisher.

Montgomery L. 2012 “Tradigital Maya”, United Kingdome Focal Press is an imprint of Elsevier.

Montgomery L. 2012 “Tradigital Maya, ch_9.1_bonus_chapter””, United Kingdome Focal Press is an imprint of Elsevier. Read 26.03.2018

http://routledge/textbooks/_author/montgomery-9780123852229/re-sources.php

Mullen T. 2007 “Introducing Character Animation with Blender”, Canada Wiley Publishing Inc.

Root G. 1985 “The Impact of Organization Structure on Productivity” Read 27.3.2018.
<http://smallbusiness.chron.com/impact-organization-structure-productivity-21902.html>

Sayenko A. 2015 “How (and Why) to Write a Great Game Design Document” 29.3.2018.
<https://gamedevelopment.tutsplus.com/articles/how-and-why-to-write-a-great-game-design-document--cms-23545>

Shah K. “Building A Basic Low Poly Character Rig In Blender” Image 51b Downloaded 10.3.2018
<https://cgi.tutsplus.com/tutorials/building-a-basic-low-poly-character-rig-in-blender--cg-16955>

Sutherland J. 2003 “SCRUM: Keep Team Size Under 7!” Read 27.3.2018
<https://www.scruminc.com/scrum-keep-team-size-under-7/>

Williams R. 2001 “The Animators Survival Kit”, USA Faber and Faber Limited.