Venni Kauppila

DESIGNING A BASIC ANIMATION MOVESET OF A PLAYABLE CHARACTER FOR A 2D SIDE-SCROLLER GAME

Bachelor's thesis

Degree programme in Game Design

2023

South-Eastern Finland
University of Applied Sciences

South-Eastern Finland University of Applied Sciences

Degree title	Bachelor of Culture and Arts
Author(s)	Venni Kauppila
Thesis title	Designing a basic animation moveset of a playable character for a 2D side scroller game
Commissioned by	NA
Year	2023
Pages	53 pages, 1 appendix page
Supervisor(s)	Marko Siitonen

ABSTRACT

The objective of this thesis was to study and examine the basic principles of animation and fundamentals of video game animation. These studies were then used as the basis for an animation project.

While 3D animation is the standard for modern video games, 2D animation is still relevant in the game industry, and it has been especially popular among indie video games recently. An in-depth look was taken at the positive and negative aspects of creating 2D video games, the animation process for 2D video games, and different approaches to 2D animation in video games.

The production part of this thesis included the creation of basic move-set animations, including idle, running, and attacking animations for a playable character in a potential 2D side-scroller type game. The author has provided an in-depth description of the process of creating animations for a potential video game, including implementing the animation into a game engine.

Keywords: character animation, 2D animation, video game animation, side-scroller, 2D video game

GLOSSARY

1 INTRODUCTION 6

2 BASICS OF ANIMATION 6

2.1 Timing and spacing 7

2.2 Pose-to-pose technique 8

2.3 Other approaches to animation 9

2.4 Squash and stretch 10

2.5 Anticipation 11

2.6 Overlap and follow-through 12

2.7 Weight 12

2.8 Staging and silhouette 13

2.9 Arc of motion 13

2.10 Secondary action and twinning 14

3 ANIMATION IN VIDEO GAMES 15

3.1 Video game animation fundamentals 15

3.1.1 Feel 15

3.1.2 Fluidity 17

3.1.3 Readability 17

3.1.4 Context 19

3.1.4 Elegance 19

3.2 2D animation in video games 20

3.2.1 2D video game animation workflow 22

3.3 Animation in 2D side-scroller games 25

3.3.1 Hollow Knight 25

3.3.2 Cuphead 27

3.3.3 Skullgirls 29

4 ANIMATION PROJECT 31
4.1 Character design 31
4.2 Animation process 33
4.2.1 Animating in Clip Studio Paint 34
4.2.2 Setting up the animation file 34
4.2.3 Animation phases 37
4.3 Designing basic moveset animations 37
4.3.1 Animating idle animation 38
4.3.2 Animating running animation 39
4.3.3 Animating attack animation 40
4.4 In-engine testing 41
5 CONCLUSIONS 44
REFERENCES 46
LIST OF FIGURES
APPENDICES
Appendix 1. Animation project frame counts

GLOSSARY

Moveset	The moves (such as jumping or attacking) a character has in a video game.
Side Scroller	A video game viewed in a 2D side-view camera perspective.
2D	Two-dimensional graphics.
3D	Three-dimensional graphics.
Frame-by-frame	2D animation style where each frame is drawn individually.
FPS	frames per second.
Sprite	A 2D asset in video game development.
Sprite sheet	Compilation of 2D assets.

1 INTRODUCTION

Traditional 2D animation is a classic technique that has been used for decades in film and television, but it has additionally played a significant role in video games, particularly in the early days of gaming when pixel animations were found in nearly every video game. However, over time, the use of non-pixel 2D animation in video games has become more prevalent, leading to new possibilities for game developers. Therefore, this thesis aims to explore and research traditional 2D animation and its potential applications in the medium of video games, especially side-scroller video games. The objective then is to create video game animations based on this research.

The primary research question of this thesis is how to design the basic animation moveset of a playable character in a 2D side scroller video game. To design and create optimal game animations, it is important to research the basic animation principles, as well as their application and utilization in video games. The secondary questions that these studies aim to answer are how to create game-ready animations, and how to implement them into a game engine for playability. To provide answers, an in-depth examination of video game animations and a small video game prototype will be conducted.

The research material includes written material such as books on animation principles and video game animation, video material such as video essays and Game Developers Conference lectures, and other material such as interviews and web pages. Additionally, the research includes case studies of video games that utilize traditional 2D animation. The production phase of this thesis will include creating 2D animations with digital art software Clip Studio Paint EX and testing the animations in the game engine Unity.

2 BASICS OF ANIMATION

Traditional 2D animation process involves creating multiple frames, with each frame differing slightly from the previous one. When the frames are played back in a chronological order, the drawings give an illusion of movement. (Moreno 2014, 4.)

Animation, especially traditional 2D frame-by-frame animation, requires an extensive amount of planning. For example, the number of frames is an important aspect which should be known before the beginning of the animation process. The industry standard for film is typically 24 frames per second, while television often uses 25 FPS (Roberts 2011, 2). However, video games generally have at least 30 FPS, with the modern standard being 60 FPS or even 120 FPS. Therefore, 30 FPS is considered the standard for video games. (Cooper 2021, 290; Klappenbach 2022.)

However, animating 30 or even 24 frames for each second of animation can be a nearly impossible task due to being very time-consuming and expensive. To reduce the workload of the animator, holds can be used. Hold means a frame is shown (held) for a longer time, which reduces the number of unique frames needed for animation. For instance, if an animation has six unique frames, each of them can be held for five frames in a 30FPS animation (5 multiplied by 6 equals 30). (Cooper 2021, 290; Halas & Whitaker 2009, 1.)

2.1 Timing and spacing

Timing and spacing are essential aspects of animation. Without proper attention to timing and spacing, the quality of animation will suffer greatly, and the illusion of movement may not be convincing.

Timing is a crucial element in determining when certain parts of the animation will happen. For example, a bouncing ball hitting the ground can be timed in a way that each hit is a part of a rhythm. Spacing refers to the placement of the animated object(s) in different frames. For instance, fast movement can cause

the spacing to be greater (more movement between frames) while slow movement can cause the spacing to be smaller (less movement between the frames). As shown in Figure 1, when the ball is at its slowest stage (up in the air), the frames are spaced closer to each other. (Williams 2001, 36-37.)

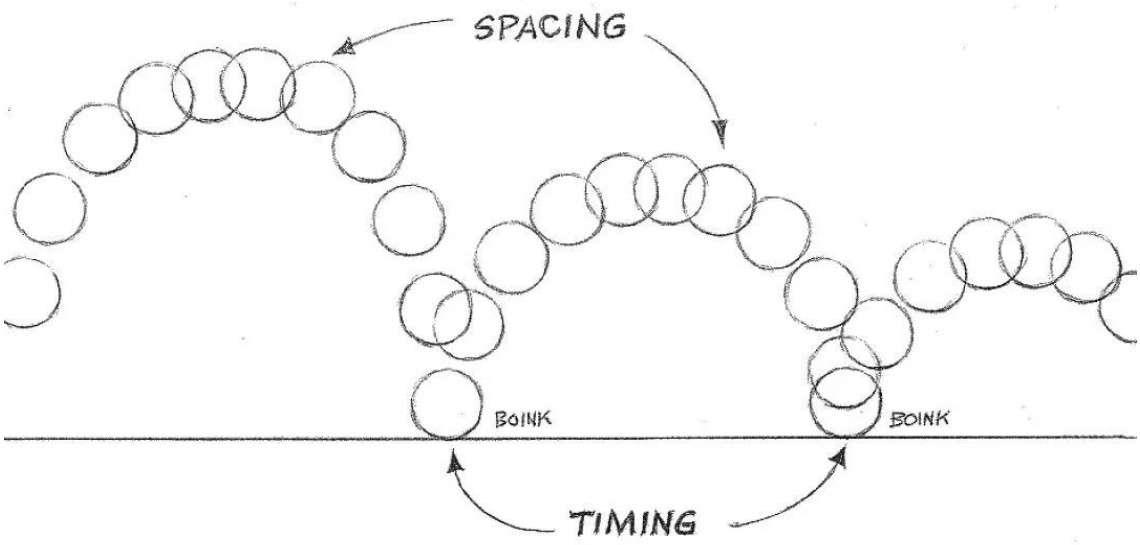

Figure 1. Timing and spacing of an animation of a bouncing ball (Williams 2001)

2.2 Pose-to-pose technique

One of the most common animation techniques is called pose-to-pose animation. This involves drawing the main positions (extremes) first, and then blocking out the rest of the action based on these main positions (inbetweens/breakdowns). (Besen & Hallett 2008, 124.)

The main positions are known as extremes, which tend to indicate a beginning, an end, or a change in action. After establishing the extremes, an even amount of inbetweens can be added. In five frames of animation, frames one and five would be extremes, and two and four are inbetweens. The third frame would be a position often called breakdown, or passing position, as shown in Figure 2. (Williams 2001, 48-49.)

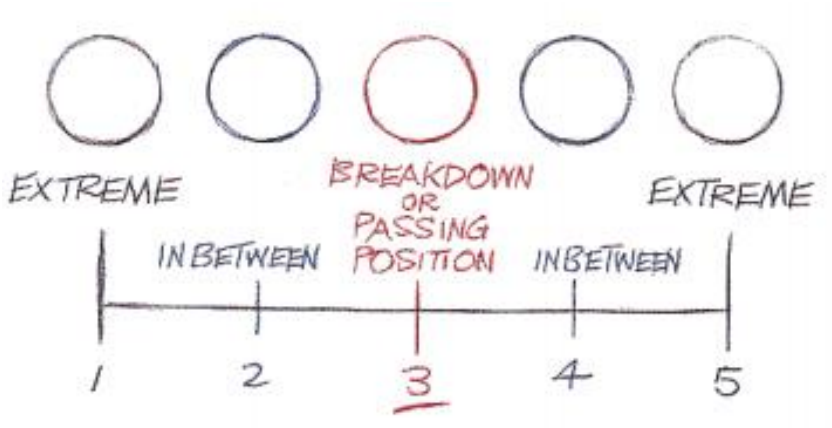

Figure 2. Extremes and inbetweens in an animation of five frames (Williams 2001)

If an animation needs to have eased transitions into and out of extreme positions, more inbetween frames can be added. This technique is often referred to as easing in and easing out or slowing in and slowing out (Williams 2001, 50-51). Halas and Whitaker (2009, 51) elaborated that “the time it takes to reach a hold depends on the momentum of the object or character". In other words, slower movements require more frames to ease in and out than faster ones.

Keys are few frames that can be used to describe the story. While Keys and extremes are often thought of as the same element, separating them can be useful for animation. By separating keys and extremes, a cleaner result can be achieved. This is because having a few storytelling keys keeps the animation focused on the purpose of the shot. (Williams 2001, 57-60.)

2.3 Other approaches to animation

In addition to pose-to-pose, there is a technique known as straight-ahead. When animating straight ahead, the animator creates all the frames in chronological order, based on the previous frame(s) (Williams 2001, 61). Therefore, it can be described as a more spontaneous technique, as it does not require as much planning as pose-to-pose animation (Besen & Hallett 2008, 124). While straight ahead can be good for achieving spontaneity and flow, it can be difficult to maintain consistency in the animation when using this technique (Williams 2001, 61).

Additionally, it is possible to combine the structured pose-to-pose technique with the more spontaneous straight-ahead technique. First, the animation is roughly planned with a few storyboard keys. Next, some other important frames, such as extremes, are added. The beginning composition is the same as with the pose-to-pose technique. However, instead of viewing these positions as strict and permanent points of animation, they can now be viewed as guides. The rest of the animation is done by going straight ahead while using the starting positions as guides but understanding that they can be changed if it benefits the animation overall. Improvisation and revising are important aspects of this technique. This technique is described as a good balance of spontaneous and structured, therefore, it can be considered one of the best to use for animation. (Williams 2001, 63.)

2.4 Squash and stretch

Squash and stretch is a technique which makes animation more exaggerated and engaging (Cooper 2021, 30; Besen & Hallett 2008, 128). In Figure 3, there are two animations of a ball: Example A does not implement squash and stretch while Example B does. In Example A the ball retains the same shape throughout the entire animation, resulting in a stiff and unnatural animation. In Example B the ball stretches more depending on its movement speed and gets squashed when it hits the ground. The outcome of example B would be more convincing and engaging.

Animations that do not implement squash and stretch tend to look inflexible and stiff. Therefore, every animation style benefits from some degree of distortion to look natural, with the amount depending on the style. Realistic animation styles might only require a small amount of squash and stretch, while more cartoony styles often rely on this technique heavily. (Besen & Hallett 2008, 128.)

While squash and stretch is an acceptable form of exaggeration, changes in volume are generally not. Unless the change in volume is related to change in perspective, increasing or decreasing the size of the animated object tends to look unrealistic and lessens the quality of the animation. (Culhane 1988, 162.)

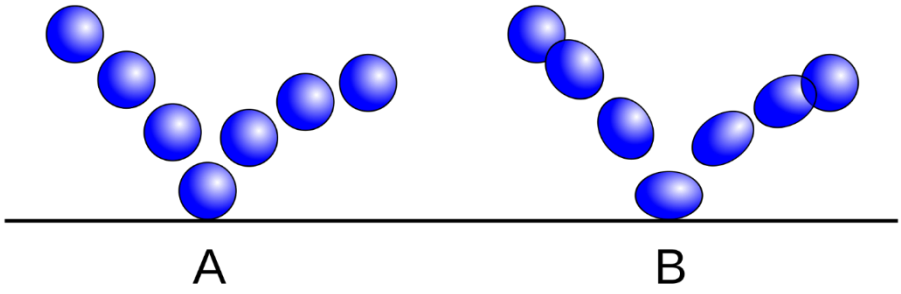

Figure 3. Animation of a ball. A does not implement squash and stretch-technique, while B does (Lampman 2008)

2.5 Anticipation

Besen and Hallett (2008, 130) explain that before the main action, there should be a preliminary movement in the opposite direction, known as anticipation. The purpose of anticipation is to strengthen the upcoming action, as if gathering energy to release during the main action (Besen & Hallett 2008, 130). Additionally, anticipation lets the audience know what is going to happen, so they can anticipate the coming actions as well (Williams 2001, 274).

In Figure 4, the viewer's attention would be focused when the anticipation starts to happen. Therefore, without a proper anticipation, it is possible the viewer would at least partially miss the main action (Halas & Whitaker 2009, 56). Additionally, Figure 4 displays squash and stretch as the character is compressed before the jump, fully stretched during the jump, and then compressed again in mid-air.

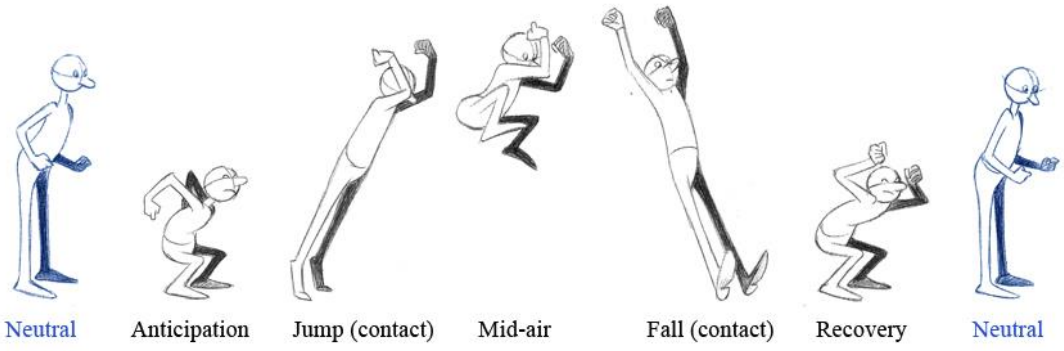

Figure 4. Anticipation is a preparatory movement before the main action, in this case jump (Williams 2001)

2.6 Overlap and follow-through

In general, not all parts of the subject move at the same time during movement. The main body tends to move first, while other attached elements, such as limbs or clothes, follow. This is known as overlapping action, where one part moves and others follow. (Williams 2001, 226.)

When an additional element is following the main body, its movement will most likely be different from the main body. Therefore, predicting the movement of the additional element can be rather difficult. Some factors that can influence extremity movement include the weight and flexibility of the extremity, the action being performed and any potential air resistance. (Halas & Whitaker 2009, 59.)

After the main body stops moving, there may be a delay of a frame or two before certain elements catch up, such as a piece of clothing or animal's tail and ears. When the main body is already in motion, some extremities may follow behind. Eventually, the extremities catch up, but at that point, the main body may have already stopped moving. The extremities might move forward slightly, and then swing back. This type of overlapping action is more commonly referred to as follow-through. (Besen & Hallett 2008, 132-133; Halas & Whitaker 2009, 59.)

2.7 Weight

Weight is a significant factor in animation, affecting everything from each body part and extremity of a character to any object the character may interact with. Accurately representing weight in animation is crucial to creating believable interactions between characters, objects, and their environments.

For instance, when animating a character picking up a heavy object, it is important to consider how the heavy weight affects the character. At first, they might struggle to pick up the object and then adjust their position. They may be gradually able to lift the object, only for it to fall with a great force. (Halas & Whitaker 2009, 66-67; Williams 2001, 256-257.)

2.8 Staging and silhouette

The purpose of staging is to ensure that the animation is both readable and engaging to the audience (Moreno 2014, 15). Proper staging creates a strong silhouette, and a successful animation is readable even in silhouette. If the animation lacks a strong silhouette, the audience may not understand the action. Readability of an animated action can often be increased by showing an action from profile instead of front view (Williams 2001, 251). (Besen & Hallett 2008, 144.)

Furthermore, including a clear line of action can aid with staging. A line of action typically refers to the line that runs through a character's pose, as illustrated in Figure 5. It is a useful tool, as a strong line of action can make poses more dynamic and exaggerated. (Cani et al. 2013, 2.)

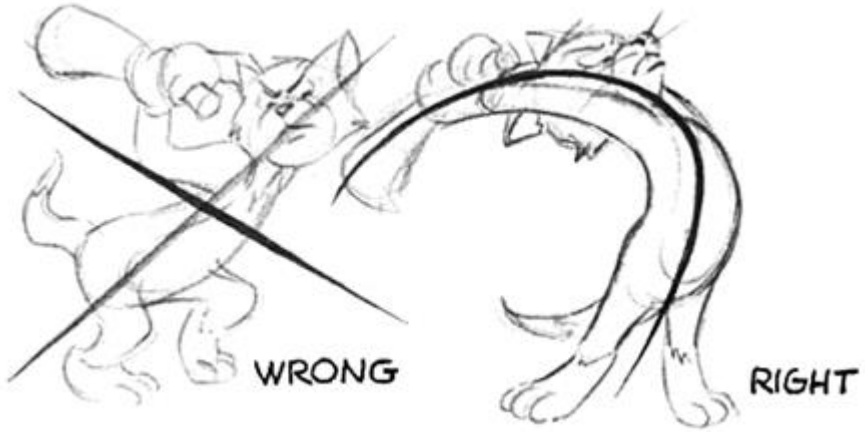

Figure 5. Left pose displays a weak line of action, while the right one displays a strong one (Blair 1994)

2.9 Arc of motion

Most motion follows an arc. For example, in Figure 6, a realistic result can only be achieved with an arc of motion. If the ball moves straight, the arm will appear to shrink. However, there are some exceptions. For example, some quick or powerful actions might have a straight line of motion. A bullet shot out of a gun would have a straight line of motion due to high momentum. However, if it does

not hit anything on its way, it will eventually start moving in an arc. (Williams 2001, 90.)

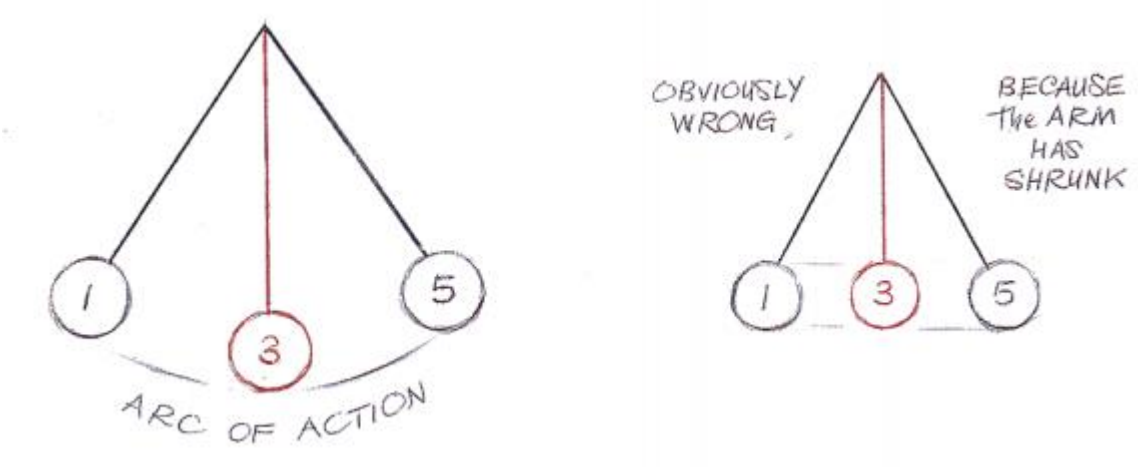

Figure 6. Arc of action compared with lack of arc (Williams 2001)

According to Blair (1994, 167) subjects in nature, including humans, animals, and other living creatures, do not move in straight lines. Therefore, it is important to keep in mind that while mechanical machines or robots may move along a straight or circular path of motion, trying to animate an animal, human or any other living creature in this way will result in unnatural, mechanical-looking movement (Roberts 2011, 60).

2.10 Secondary action and twinning

A secondary action can enhance the main action. For example, if the primary action is walking, the secondary action could be swinging arms. Or if an animal is walking on four legs, its secondary action can be flailing ears or tail. The secondary action is often delayed compared to the main action. When the action is similar or has the same rhythm as the other action, it is called twinning. While twinning can result in a dull, robotic animation, it can be utilized in some cases. For example, twinning can be used to emphasize the feeling of balance and harmony in animation. (Williams 2001, 324-325; Blair 1994, 144; Roberts 2011, 60.)

3 ANIMATION IN VIDEO GAMES

While animation is used in various mediums like film and television, it is essential to understand that creating animations for a video game has its own challenges. Animations designed for video games must work seamlessly with the gameplay, making video game animation a more technical profession compared to animation for film or television. However, artistic ability is still crucial when it comes to video game animation. And while having both artistic and technical skills is considered a great strength, it is not required if the game animation team has animators with complementary skills. (Cooper 2021, 1-2.)

3.1 Video game animation fundamentals

According to Cooper (2021, 41-42), understanding the five principles of feel, fluidity, readability, context, and elegance is crucial for those aiming to become video game animators. These five fundamentals are important to consider in addition to the principles of animation, but not as a replacement for them. Therefore, anyone aiming to become a video game animator should understand both the traditional animation principles and the video game animation principles.

3.1.1 Feel

What sets video game animation apart from animation for traditional mediums is the interactivity. Therefore, video game animators must consider how the player will feel about the animation. For example, players often feel frustrated if their input is not answered with an animation in a reasonable amount of time. Therefore, responsiveness must always be considered when creating game animations. However, this often means that animations should be shorter to increase the responsiveness. Nonetheless, anticipation, weight or force should not be sacrificed to improve responsiveness. Essentially, animators must balance creating animations that are both fluid and responsive. (Cooper 2021, 42.)

Floyd (2020) explains why Mario's triple jump animation from *Super Mario 64* (1996) feels satisfying to perform. He describes the jump as exaggerated, almost like jumping on a trampoline. While it may not be particularly realistic, it is extremely fun for the player. Additionally, Animation principles can be used to enhance the feel of gameplay animations. For example, in *Celeste* (2018), squash and stretch is used in the jump animations (Brown 2019). Another way to make sure that the movement feels satisfying is to ensure that the player feels in control. In *Hollow Knight* (2017), the developers ensured that the jump had a lot of initial lift, but no acceleration or deceleration. The focus was to make sure that the player feels in control of the movement, even if it means breaking the rules of gravity and acceleration. (Gibson & Pellen 2018.)

Furthermore, visual feedback can enhance the feel of video game animations. Video games are often fast paced, with many on-screen elements. Appropriate visual feedback does not only improve the feel and impact of actions but increase their readability as well. For example, in Figure 7, when the player attacks in Hollow Knight, the swing of a sword generates a large effect on the screen, making the action feel more powerful and noticeable. (Cooper 2021, 44; Floyd 2019b.)

Figure 7. Hollow Knight's attack animation effect (Team Cherry 2017)

3.1.2 Fluidity

It is important to consider that playable characters, as well as other characters such as enemies, have multiple moves that need to work seamlessly with each other. For instance, a playable character may have moves such as walking, running, jumping, and attacking, and the animator must ensure that the moves are fluid and maintain a flow regardless of the order in which the player uses them. (Cooper 2021, 45.)

In other words, the animator must ensure that the animations blend consistently together. The most effective way to achieve this varies depending on the type of video game and what kind of animations it uses. For example, in *The Legend of Zelda: Breath of the Wild* (2017) nearly every vertical surface can be climbed on, making climbing animation one of the most common animations in the game alongside walking and running. This means that there are multiple entry and exit points for the climbing state, and the animations need to flow smoothly between them. The game addresses this issue by well-thought-out transition animations, such as those that depend on whether the player enters climbing by running or jumping. This does not mean that every possible way to enter or exit climbing has its own transition animation, but it is worth considering whether a certain state should have a transition animation or not. (Floyd 2019a.)

3.1.3 Readability

Video games are often fast paced, requiring players to quickly read the movements of characters who may move swiftly and change positions constantly (Cartwright 2014). To ensure that a character is easy to read, it needs to have a strong silhouette (Cooper 2021, 49-50). Figure 8 demonstrates how a silhouette can be used to determine whether a game animation is readable or not.

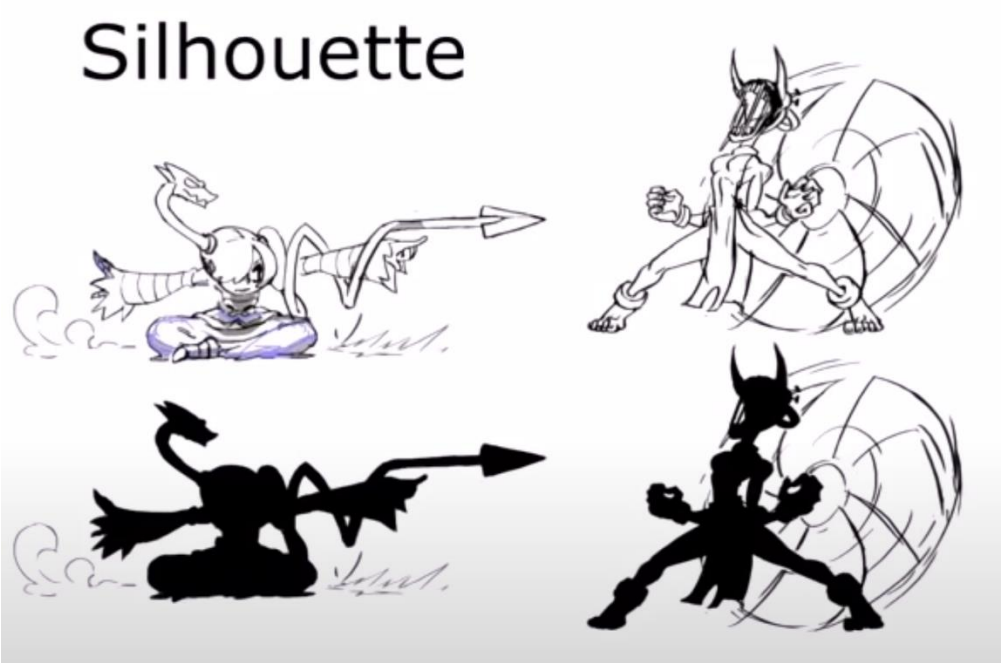

Figure 8. game animation sketches that display a strong silhouette (Cartwright 2014)

When developing a 2D game where the camera is fixed to the side, the actions only need to be readable from one angle. However, for 3D games, the action needs to be readable from all angles. This means that 3D animators must ensure that the action is readable from any angle, while 2D animators can focus on animating an action which is readable from only one angle. (Cooper 2021, 48-49.)

The correct use of anticipation can increase readability by letting the player know what kind of movement or attack they can expect from an opponent or enemy character. As previously explained, too much anticipation in the animations of the player character can feel frustrating for the player who expects responsiveness. However, not enough anticipation in the animations of the enemy can lead to frustration as well, as the player might not be able to read incoming actions properly. Therefore, enemy characters can generally have more anticipation as it increases the readability of the gameplay. (Cartwright 2014.)

3.1.4 Context

In linear mediums such as film and television, the context of an animated action depends on the scene in which it is used. However, in video games, the player has what are essentially free actions to use any moves they have at any time. Other characters' actions tend to be somewhat affected by the player's actions as well, such as when the player is fighting an enemy. Therefore, video game animations are commonly used repeatedly in various settings. (Cooper 2021, 50-51.)

To enhance video game animations, animators should consider what kind of unique perks a character could have when animating them. Giving characters animations which reflect their personality, design or other unique traits makes them more distinct from other characters. (Cooper 2021, 51.)

3.1.5 Elegance

The principle of elegance refers to the systems which animations use. These systems must ensure that the animations are consistent and functional in any situation or context. While creating complicated animation systems may be tempting, simplicity is often a better approach. For example, a game like *Animal Crossing: New Horizons* (2020) benefits from a simple effect animation that effectively does its job (Figure 9) rather than a complex alternative. Detailed effect animation or simulated physics not only require more work, but they could potentially worsen the performance of the video game as well. (Cooper 2021, 53-54; Root 2020.)

Figure 9. Simple water splash animation from Animal Crossing: New Horizons (Nintendo 2020)

When it comes to video games, the animator must understand which animations to prioritize. Not everything in a game requires animation, and not everything should be animated. For instance, in a game with a focus on environment and atmosphere, there might be more environmental animations or effects which enhance the atmosphere. In contrast, action games tend to prioritize fluid and eye-catching character animations. Therefore, it is critical to identify the type of animations to prioritize to avoid wasting resources on animations that may not be essential for the given project. (Cooper 2021, 54; Root 2020.)

3.2 2D animation in video games

2D animation remains relevant in the video game industry despite the advancement of technology. Many recent and popular indie games use 2D animation, and some are even fully 2D animated, such as Team Cherry's *Hollow Knight* (2017) or Studio MDHR's *Cuphead* (2017). The popularity of indie games utilizing 2D animation is likely the result of several aspects. For example, 2D is an eye-catching stylistic choice, as most video games are in 3D. Additionally, 2D animation can result in a tighter gameplay compared to 3D animation, as 3D animation has the potential to glitch and break. Lastly, the artistic control is greater in 2D animation. Since the animation is only done from one angle, a great amount of effort can be put in nailing the action just right. (Cooper 2021, 270.)

In addition, many mobile games are 2D rather than 3D, as it is generally easier to develop a 2D video game instead of a 3D video game. Creating a 2D game is usually cheaper as well, and according to Cooper (2021, 270) 2D games tend to be a better choice for those with a limited budget. Furthermore, mobile platforms generally have less powerful performance compared to other platforms, like PC and console. 2D games typically require less from a system than 3D games, which can be considered another benefit of 2D animation in mobile game production.

However, 2D animation for video games has its downsides. For instance, there is less versatility for animation reuse. In a 3D video game, a character can have multiple costumes and animations between characters can be shared. However, this is usually not possible for 2D video games. Another downside of 2D is that characters cannot be changed during production. For example, if an issue with a character design was noticed after the animations for the character were produced, all their animations would have to be remade. (Cooper 2021, 270.)

Frame-by-frame 2D animation especially can be expensive and time-consuming, and it may have technical limitations. However, there are other approaches to 2D animation in addition to frame-by-frame animation. For example, Klei Entertainment, a video game studio best-known for their 2D video games such as *Don't Starve* (2013), decided to mix traditional hand-drawn animation with Flash animation. Their animation process begins with creating the keys, as in most other 2D animations. However, instead of creating every inbetween from scratch, they use character builds to create animations efficiently. This means that already-made body parts can be re-used to create inbetweens. (Agala & Bouthillier 2014.)

While there are various ways to create 2D animations for games, such as Klei Entertainment's animation process, many developers still want to include traditional frame-by-frame 2D animation, even though it can be expensive and time-consuming. For example, according to Clark (2017), the main reason for the

delay of *Cuphead* (2017) was the time-consuming process of creating hand-drawn, 24 FPS animations. However, the game proved to be financially successful, selling one million copies in just the first two weeks of release (Moldenhauer & Moldenhauer 2017). This success enabled the development of additional downloadable content for the game, which was released in 2022.

Gibson and Pellen, the directors of *Hollow Knight* (2017) had multiple reasons why they chose 2D over 3D for Hollow Knight. Firstly, both directors already had backgrounds in 2D animation. Additionally, Gibson (2017) thought that developing a 2D game is more achievable with a small development team. Furthermore, there are more potential technical problems with 3D animation compared to 2D animation: Gibson (2017) explains that they had no technical problems with 2D animation during the game's development, which made the development a smooth process.

Another approach to 2D animation in video games is pixel animation. While pixel animation can be considered a type of traditional 2D animation, it can be viewed as a simpler and possibly cheaper alternative compared to high definition 2D animation. The approachable nature of pixel animation makes it a popular choice among solo and indie game developers, not to mention it is the original visual style of video games. (Cooper 2021, 271.)

3.2.1 2D video game animation workflow

The animation process usually begins with the reference. While image references can be useful, video references are often even more useful. In fact, many animators even film their own references (Cooper 2021, 87-88). Filming oneself or others can be particularly useful because most animated characters, even non-human ones, have similar features such as head, arms, and legs (Clark, 2017).

Often, the first sketches for an animation are rough storyboard sketches. For example, if the subject for the animation is a boss character, the storyboard sketches may depict their main attacks (as seen in Figure 10). Some animators

may prefer doing a few rough sketches prior to referencing to get their initial ideas out before they are influenced by something else. (Clark 2017.)

Figure 10. Rough sketches of a boss battle (Clark 2017)

When creating 2D character animations, it is necessary to stay on model. This means that every frame must be consistent with the character, its proportions and other aspects must remain the same despite the movement (except when it is done purposefully; squash and stretch for example). One way to stay on model is to have a model sheet, which is an image that displays the character from multiple angles. According to Clark (2017), an idle animation can be used as a reference point for the rest of the animations instead of a model sheet. Every animation starts from the idle animation and ends up returning to it. Another way to stay on model is to study the model by breaking it down into its basic components and shapes and memorizing them (Agala & Bouthillier 2014).

Pose-to-pose is considered a better technique for video game animation compared to straight-ahead animation (Cooper 2021, 32-33). Therefore, animating usually begins with rough key sketches that prioritize clear communication of the action rather than detail or frames. Cartwright (2014)

explained that when working on *Skullgirls* (2012), the animation of a character would be implemented in the game at this stage to test it out (Figure 11). After a designer has tested the animation in the game, the animation would be passed back to the animator with feedback of what should be changed or fixed. Cooper (2021, 90), elaborates on the same topic, stating that pose-to-pose technique in game animation is extremely useful because the animation can be tested in-game even when it is at a very early stage. Ultimately, it is unlikely that the animation is usable without testing and changing it multiple times (Cooper 2021, 92).

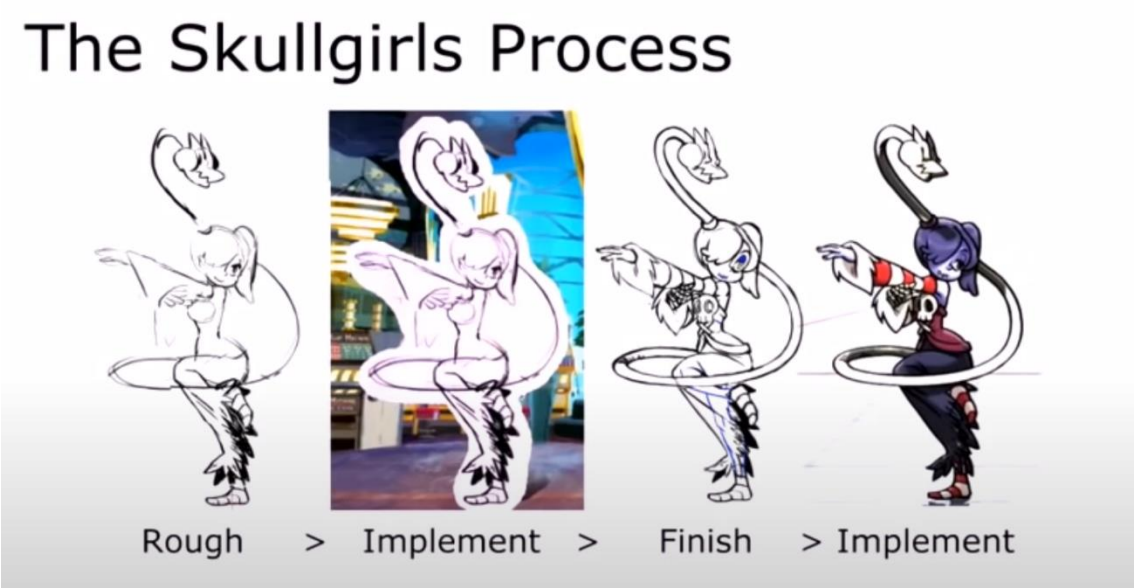

Figure 11. The animation process in Skullgirls (Cartwright 2014)

After the keys are created, inbetweens (or breakdowns) are added to the animation. Additional inbetweens are added until the animation has a smooth flow and is consistent with the other game animations. To ensure consistency across all game animations, it is helpful to establish some kind of frame count guidelines (Cartwright 2014).

After sketching the animation, it can be lined and colored. Most digital art and animation software includes a paint bucket tool, which is a tool that can color any closed area with relative ease. However, this method of coloring may leave transparent gaps, which can be difficult to notice in the art software but become more apparent when the frames are imported to a game engine, for example. To avoid this issue, the magic wand tool can be used. The magic wand tool selects

an area, and only the selected area can be colored. To ensure that the entire desired area is colored without transparent gaps, everything except the desired area should be selected with the magic wand. Then, the selected area can be inverted, and the desired area can be colored. (Pantoja 2021.)

3.3 Animation in 2D side-scroller games

This section examines the utilization of 2D graphics and animation by three popular indie video games from the 2010s. The objective is to study the animation techniques and stylistic choices used by each video game's animation. The analysis includes an overview of the animation, possible utilization of effects animation, and an in-depth examination of the idle, walking, and attack animations. Additionally, it should be studied whether the animations feel responsive and whether the gameplay animations retain a good flow or not. Furthermore, any unique gimmicks or solutions should be noted to find out how these successful video games enhance their animations and appeal.

3.3.1 Hollow Knight

Hollow Knight (2017) is a 2D action-adventure video game developed by Team Cherry. Hollow Knight was developed using Unity, and the art is hand-drawn and animated in Adobe Photoshop (Unity n.d.b).

The animation style of Hollow Knight could be described as rather simple, as the animation does not use any more frames than is necessary. This makes the game feel very quick and responsive. The game relies heavily on effects animations, some which are hand-drawn and others which are simulated in the game engine, Unity. The hand-drawn effects are usually part of the character animation, such as slashes or dashes following the player's movement or attacks. The in-engine animations include smoke and particle effects, which can often be seen when the player interacts with the environment or when characters hit each other in combat. (Unity n.d.b.)

When it comes to idle animations, Hollow Knight has a very subtle one. In fact, it may look like the character barely moves at all. The only noticeable parts of the animation are a small head bop and subtle wiggle of the character's cape. Despite being a very subtle animation, it fits the character.

When it comes to walking in Hollow Knight, the movement speed feels appropriate. It is not too fast to be considered running, nor is it too slow to feel sluggish. When inspecting the walking animation of Hollow Knight, one may notice that it consists of three different animations. The first animation is a transition animation from idle which plays when the player starts walking. During this animation, the character's head goes down and its cape does a great swing, as if the character is pulling itself with a force (this can be seen in Figure 12). The second animation plays as long as the player is moving straight ahead, the character is walking with its cape swinging subtly (Figure 13). The third animation is another transition animation which plays when the movement stops, and the character reverts to idle animation. The walking stops immediately to keep the controls responsive, but the cape does a final swing (Figure 14).

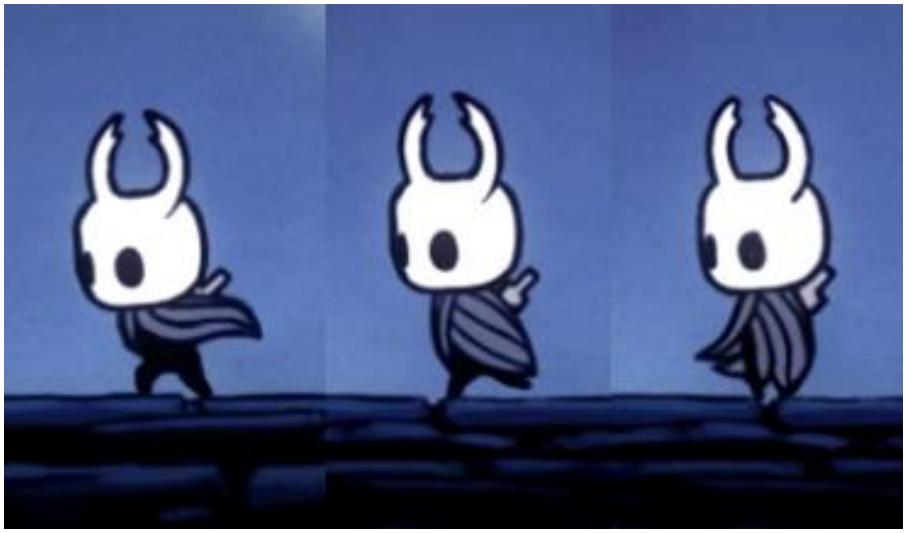

Figure 12., 13., and 14. Animations which play when the player starts walking, is walking and stops walking (Team Cherry 2017)

The basic attack animations of Hollow Knight are quick, consisting of only a few frames. There is no noticeable anticipation, though anticipation is less important for playable characters than for enemies. There are two variations of the attack: one where the sword swings from left to right, and another where it swings from right to left. Additionally, the character's head tilts slightly in the direction of the movement. This variation makes the attack more enjoyable when it is spammed, which is most likely why it was implemented in the first place.

The basic attack can be used while walking or jumping, and both horizontally and vertically. The player can attack upward at any time, but attacking downward is only possible while the player is in the air. Additionally, the effects animations are crucial for the attack animation. The slashing effect shows the attack's range and makes it much more readable.

3.3.2 Cuphead

Cuphead (2017) is a classic 2D run and gun action video game developed by Studio MDHR. Cuphead was developed using Unity, and the animations were sketched and inked on paper, and then colored in Adobe Photoshop (Unity n.d.a).

Compared to Hollow Knight, Cuphead's animation tends to use much more frames, resulting in smoother animation. In fact, everything in Cuphead is animated 24 frames a second which is the standard rate for film (Moldenhauer 2016). Additionally, a significant number of effects animations are used. Most effects in Cuphead are hand-drawn to maintain a traditional animated film aesthetic. There are many types of effects in Cuphead, such as sparkles and speed lines, though the most common effect is a white air cloud, which is generated by almost every type of movement. In Figure 15, the special attack move produces speed lines, air clouds, a light flash, and a flaming ball.

Figure 15. Cuphead's effects animation (Studio MDHR 2017)

Cuphead's idle animation is very bouncy with the character moving up and down in a noticeable manner. This animation can be described as twinning, as the character's movements are nearly perfectly symmetrical. While twinning can sometimes result in an unnatural animation, it is likely used here for exaggerated movement. The animation in Cuphead does not strive to be realistic but stylized and greatly exaggerated.

In Cuphead, walking feels somewhat slow even though the animation looks like the result of a faster movement speed. However, this is not an issue as the player does not need to move quickly in a game like Cuphead, unlike something like Hollow Knight, where the player needs to journey great distances in a vast game world.

The walking animation in Cuphead is highly exaggerated, like the rest of the animations. It does not aim for a realistic style of movement, and it does not need to. This unrealistic approach makes the animation interesting and unique. The walk does not have any noticeable follow-through, so all the elements move at the same pace. The character's head tilts slightly with the movement, and some milk spills from the mug on the character's head.

Cuphead is a shooter, or a “run and gun”-type game where all the attacks are ranged. The basic attack is simple, with no noticeable anticipation, and it is very responsive and quick. The animation is minimal, consisting only of a slight wiggle and some hand movement.

The special attack of Cuphead can be charged by hitting enemies or parrying enemy attacks. The animation of the special attack is significantly different from regular attacks, featuring a more complex and slower animation with drastic effects animations. The attack has some anticipation and a lot of force, which causes the character to get pushed back.

3.3.3 Skullgirls

Skullgirls (2012) is a 2D fighting video game developed by Reverge Labs and published by Autumn Games. It was developed with Z-Engine, which is the studio’s own engine (Mobygames n.d.). The animation is hand-drawn, without any assistance by tools such as Flash or 3D animation (Cartwright 2011).

While Skullgirls can be considered a 2D side-scroller game just like Hollow Knight and Cuphead, it has some differences due to being a fighting game. For example, it features a large number of playable characters, each with their own unique animations. Studying the different characters in Skullgirls can be a good way to learn how to animate characters with different personalities and physical traits. However, for now, only one character’s animations are examined so they can be studied more in-depth. The character whose animations are studied is Filia, who is considered the official main protagonist of the game.

Skullgirls features smooth animation that utilizes a large number of frames. The animations are significantly fast paced, and the player can repeat moves quickly. The game’s animation includes some effects animations, typically used to indicate fast movement or hitting a target.

Compared to video games of other genres, fighting games tend to have more expressive idle animations. This is particularly true for Skullgirls, where most of the idle animations are greatly animated and expressive. However, some characters have more subtle idle animations, possibly to reflect their personalities.

Filia’s idle animation is very bouncy and expressive. There is some follow-through in the animation of the clothes, as they are moving behind the main body. Additionally, her hair has its own animation, separate from the body. In fact, according to the game’s lore, what appears to be Filia's hair is alive and its own character.

Skullgirls is a fighting game, so the walking animations are different compared to most games. The characters must maintain a stance, as if they are always ready to attack. Therefore, this stance is retained even while walking. To ensure the character is always facing the opponent, there are two unique walking animations for each character, one forward and one backward. The movement speed is rather slow, as the size of the level, or arena is very small.

Filia’s walk has follow-through on both her clothes and hair, as their movement follows her main body. Her hair bounces up when her body goes down, and down when her body goes up. Her backwards walk has a creative and unique animation, as she uses her hair to walk.

Every character in Skullgirls has an extensive amount of attack moves, as the combat is the most essential part of a fighting game. The attacks can be roughly divided into some categories, such as regular attacks, mid-air attacks, crouch attacks, grab attacks and special attacks. The attacks feel responsive and are extremely fast paced, making them satisfying to repeat. Most, if not all attacks have minimal anticipation, though it can be difficult to notice given the speed of the attacks. Anticipation is particularly important in a game like Skullgirls, where any character can be played against. The attacks usually retain a good flow regardless of the order in which they are executed.

As each character has a large number of attacks, including only kicks or punches for attacks would be tedious and uninteresting for the player. For example, while Filia has a few kicks, most of her attacks feel unique and creative as they use her shapeshifting hair. Figure 16 displays some of Filia's attacks that utilize her hair as a weapon. These kinds of unique attacks increase the game's visual appeal and make the moves more memorable to players.

Figure 16. Most of Filia's attacks utilize her shapeshifting hair (Autumn Games 2012)

4 ANIMATION PROJECT

For the production-part of this thesis, the author has designed, animated and tested animations for a playable character in a 2D side-scroller type video game. The animations were created using Clip Studio Paint EX, a digital art software that can be used to create illustrations, comics, and animations. After creating the animations, they were tested in the game engine Unity to determine whether they can work in a video game or not.

4.1 Character design

When designing a character that will be animated, many aspects must be considered. This is especially true for traditional, frame-by-frame 2D animation, where the character should be designed in a way that is suitable for animation. For example, a complex character design will be much more difficult to animate than a simplified one. Furthermore, character design influences how a character will move. Two characters who look different are expected to move differently as well. (Besen & Hallett 2008, 66-67.)

Additionally, the color palette of the character should be carefully considered. A few thoughtfully chosen colors are often a better choice than a huge number of colors. In addition to making coloring the character easier, it makes it less likely that the character will blend in with the background. (Culhane 1988, 75.)

Even a solid, well-crafted character design might not work in animation. Therefore, the only way to ensure that the character works in animation is to try animating it. For example, the animator might find out that an aspect of a character, such as their limb or accessory, which looked fine while static, looks strange when animated. In this case, the only option may be to revisit the character design and fix problematic aspects. (Culhane 1988, 76.)

Although the importance of a strong silhouette in animation has been established in previous chapters, it is worth noting that the character design plays a huge role in determining whether a silhouette is readable or not. A strong character design tends to be recognizable even as a silhouette.

Therefore, the objective was to create a character whose design is simple yet recognizable, with a limited color palette and a strong silhouette. Figure 17 is the final character design: an energetic teenage girl who is supposed to be a fighter on the streets of a city. Her outfit is meant to reflect the fact that she is athletic and tough, and it is additionally inspired by street fashion.

Figure 17. Final character design

The style is cartoon-like, and the design itself is simple. It consists of four main colors: green, orange, gray and black. Seafoam green and orange are complementary colors, so they create a good contrast. Grey and black are neutral colors which balance the design, as green and orange are very saturated.

To achieve a strong silhouette, some parts of the character had to be exaggerated. For instance, her shoes are quite large, and her hair is spiky, and her wristband is thick. All these elements make her silhouette much stronger. Her weapon is a baseball bat, which is clearly visible in the silhouette.

4.2 Animation process

The animation process varies depending on many factors. For a fully rendered animation created with the pose-to-pose technique, the process will look roughly like this: sketching key frames, then sketching inbetween frames, then lining, then coloring and lastly rendering. However, before any of these steps, the software needs to be selected. Alternatively, the animation can be done traditionally on paper. Another approach is to sketch, or even sketch and line the animation traditionally and then finish the animation digitally.

There are some other aspects that should be considered before proceeding with animating. For example, sprite size is a rather important aspect that should be known beforehand. Sprite can refer to any 2D asset in a video game, and a player character animation or image is often called a player sprite. There are many things to consider when deciding on the player sprite size. For example, especially in the early days of video games, hardware limitations were often the reason for small sprites. Nowadays there are fewer limitations, and sprite size can be decided based on the desired amount of detail. Small sprites still have their advantages; for instance, a small player sprite can enable a greater focus on the environment. However, a larger player sprite can show a greater amount of detail in character design compared to a small sprite. (Cooper 2021, 276.)

4.2.1 Animating in Clip Studio Paint

Clip Studio Paint was chosen as the software for animation for multiple reasons. While Clip Studio Paint is most well-known for the creation of illustrations and comics, it has a built-in animation tool as well. Additionally, the author has years of experience with this software. Lastly, Clip Studio Paint tends to be more affordable compared to most other alternatives.

It is important to note that there are multiple different versions of Clip Studio Paint, each with different amounts of features. The version used for this production is Clip Studio Paint EX, which contains the most features. This version is ideal for animation as well, as it allows for unlimited frames. (Celsys, n.d.b.)

4.2.2 Setting up the animation file

Clip Studio Paint offers multiple options when creating a new project. One of the options is animation, as seen in Figure 18. When creating a new animation file, there are several useful options, such as frame size and frame rate. Most of these options can be edited later in the settings, so it is not necessary to have them all figured out immediately.

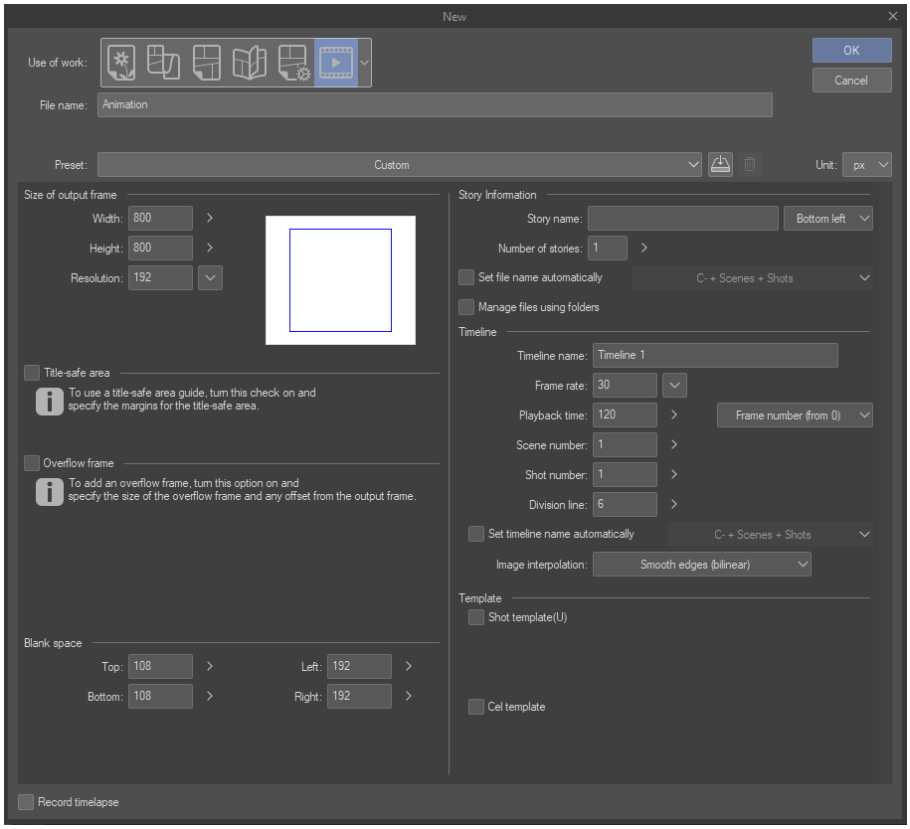

Figure 18. Clip Studio Paint's animation file creation (Celsys 2001)

At this point, the sprite size should be figured out, as it is synonymous with the file size. While it is possible to change the size or the resolution of the file afterwards, for the best result it should be figured out before starting the animation.

The timeline is one of the most important features regarding the animation process. It is an industry-standard feature for animation software that makes rearranging and adjusting the speed of animation cells simple and easy (Celsys, n.d.a). As seen in Figure 19, the timeline shows the order and duration of frames.

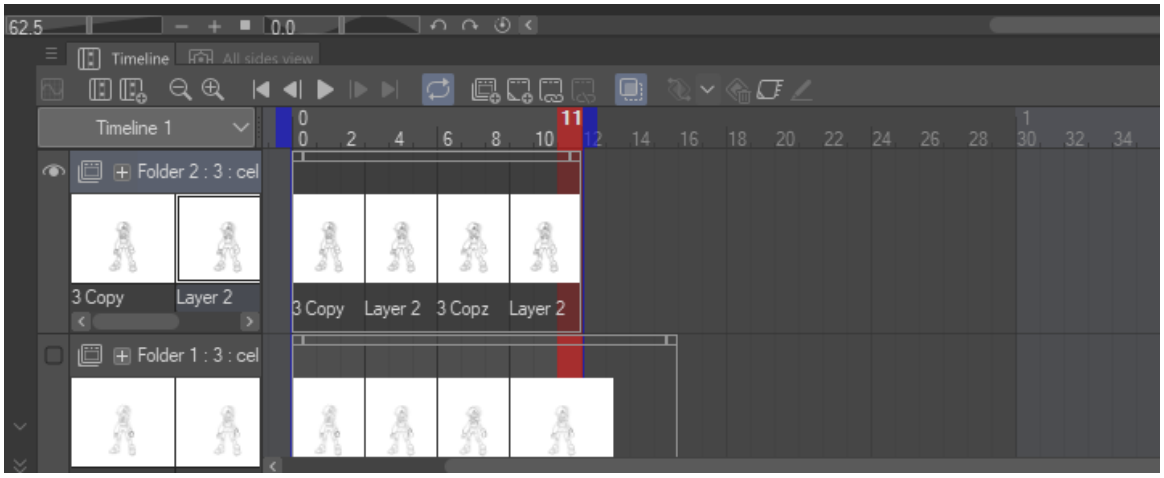

Figure 19. Clip Studio Paint's timeline for animation (Celsys 2001)

One of the other important tools for digital 2D animation is onion skin. Onion skin means displaying a number of frames before and/or after the current one (Cooper 2021, 292). Therefore, onion skin can be a useful tool, especially when creating inbetweens after key frames. Figure 20 displays onion skin used in Clip Studio Paint; the more visible the frame is, the closer it is to the current frame. In Clip Studio Paint, multiple aspects of onion skin can be adjusted, such as the transparency and number of frames.

Figure 20. Onion skin in Clip Studio Paint

4.2.3 Animation phases

The animation process for each animation began with rough sketches. The keys were sketched first, and then some in-betweens were added. The animations should be tested in-engine as early as possible, so the rough sketch animations were imported to the game engine and tested. After that, more refined sketches were created, and a few more inbetweens were added if they were needed. The animations were then lined, colored, and rendered. Figure 21 displays every step of this process.

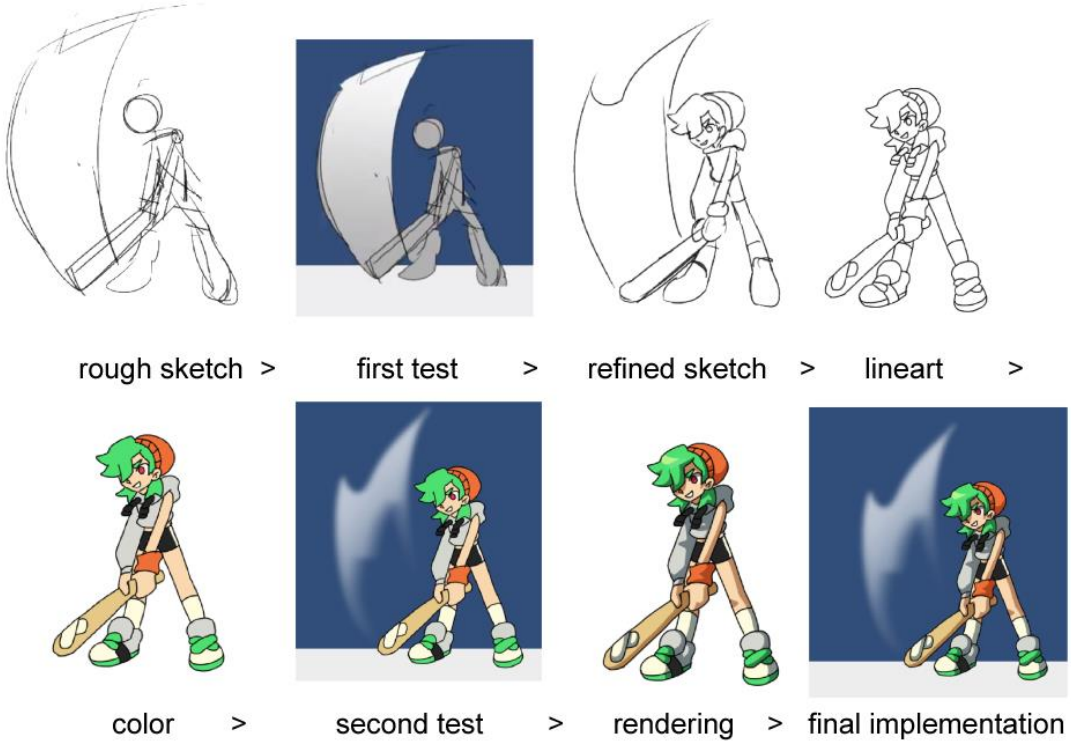

Figure 21. Animation workflow for this project

4.3 Designing basic moveset animations

Every animation for this project had to be designed with multiple aspects in mind. The animations not only had to be fluid and game-ready, but they should reflect the personality of the character as well. Therefore, this section explains the design process for each of the animations, along with descriptions for the final results.

4.3.1 Animating idle animation

The idle animation was the first animation which was designed, so it could be used as a guide for the other animations. Every other animation should work with the idle, as the idle animation is the default animation for the player character.

Most video games, excluding fighting games, have rather simple idle animations. Usually, the idle animation only includes subtle movement to indicate that the character is breathing, though sometimes there might be some subtle movement in their arms or in any other extremities. The character of this project is supposed to have an energetic personality, so the idle animation should reflect that personality while still being simple.

Figure 22 displays the completed frames of the idle animation. The animation only has four frames (three unique frames), as subtle movement can be achieved even with a few frames. The character squashes a bit, and the arms and hair move in the opposite direction.

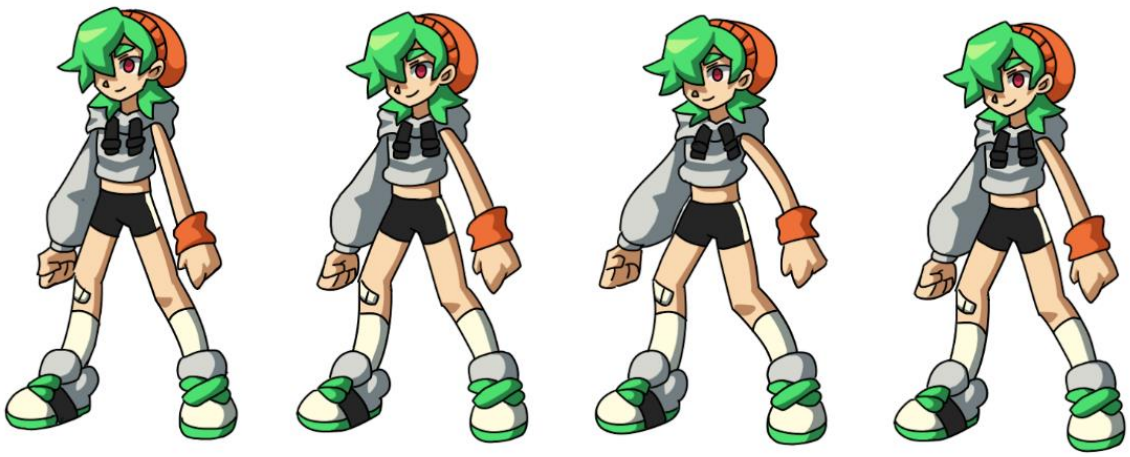

Figure 22. All the finished frames of the idle animation

4.3.2 Animating running animation

When it comes to walking and running animations, there tends to be a great amount of advice and resources available, as walking animations are common in every form of animated media. The advice and resources were helpful when animating the running animation, as well as the fact that the author already had experience creating walking animations.

There are multiple aspects that must be considered when animating walking or running, such as how the body goes up and down, how the arm swings, and when the feet make contact with the ground. When the leading foot is about to hit the ground, it's called the contact position, and when the feet are about to pass each other, it is called the passing position. Additionally, between the passing and contact positions, the character tends to move up or down, resulting in the up and down positions. (Williams 2001, 102-103, 106.)

The running animation for this project's character is supposed to reflect the character's personality, just like the idle animation. The character is energetic, so the run should feel energetic too, as opposed to feeling dispirited. The character takes large leaps in the slightly exaggerated running animation. Figure 23 displays the completed frames of the running animation, with each contact, down, passing, and up position marked.

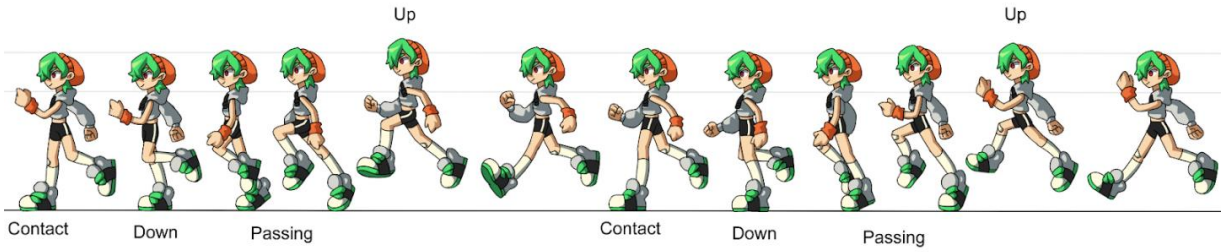

Figure 23. Finished frames of the running animation

4.3.3 Animating attack animation

Compared to walking and running animations, there are not as many resources for attack animations. Therefore, references were crucial when creating the attack animation. The first draft of the attack animation was created by heavily referencing an actual baseball player swinging his bat. However, it soon became apparent that this version would not work in a game. The anticipation phase of the attack simply lasted way too long, and the rest of the animation could have been shorter as well. The second draft was more inspired by other video game animations, such as Hollow Knight's attack animation, and it was immediately clear that the second draft would work better in a video game compared to the first draft. Additionally, the author used their own video footage for reference.

The objective was to create fluid attack animation which feels energetic. Figure 24 displays all the frames of the attack animation. The attack animation uses the first frame of the idle animation as a guide and as an entry and exit point. The attack starts with the character stretching her body upwards, preparing to draw the weapon (anticipation). Then the character does a large swing, squashing her body downwards. To ensure that the swing is readable as an attack, there's an effect on it. This effect additionally displays the range of the attack. As the character puts her weapon away, her arms swing a bit forward (follow-through). The objective is for the player to feel the force and power of the attack. While the idle frame is displayed in Figure 24, it is important to note that it is not actually a part of the attack animation.

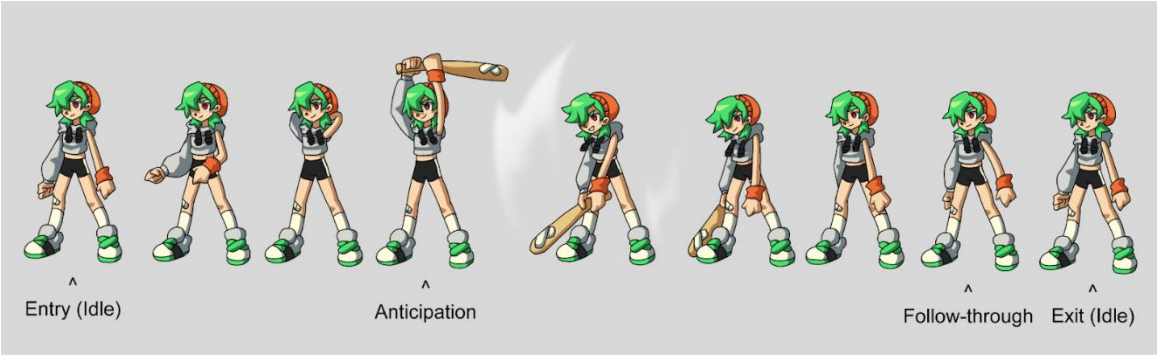

Figure 24. All the finished frames of the attack animation, except for the first and last frames which are part of the idle animation

4.4 In-engine testing

The game engine chosen for this project is Unity. Unity supports 2D video game development with a wide variety of tools, and many of the 2D video games examined in this thesis (such as Hollow Knight and Cuphead) have been developed using Unity. Additionally, the author has a good amount of experience with Unity and is familiar with the interface.

The animations were first tested in-engine after the sketches for all three of the animations were finished. Editing the animations is easiest when they are still sketches, so testing at this phase was crucial. The animations were first exported as individual images and then compiled into a sprite sheet. This means that all the images meant for an animation were in a single image file, and in the engine they could be separated again.

There are multiple ways to make animations work in Unity. First, animation clips should be created. This involves putting the sprite sheet's images in the animation clip as frames. When all the animation clips are created, Unity's Animator tool can be used. Unity's animator tool is useful when figuring out the states for each animation to be entered or exited. For example, the idle animation is usually the entry animation, meaning it plays when the game is started. The walking animation plays when the player moves the character, and the attack animation should play when the player hits a specific button or key. Figure 25 displays the animator view for this project.

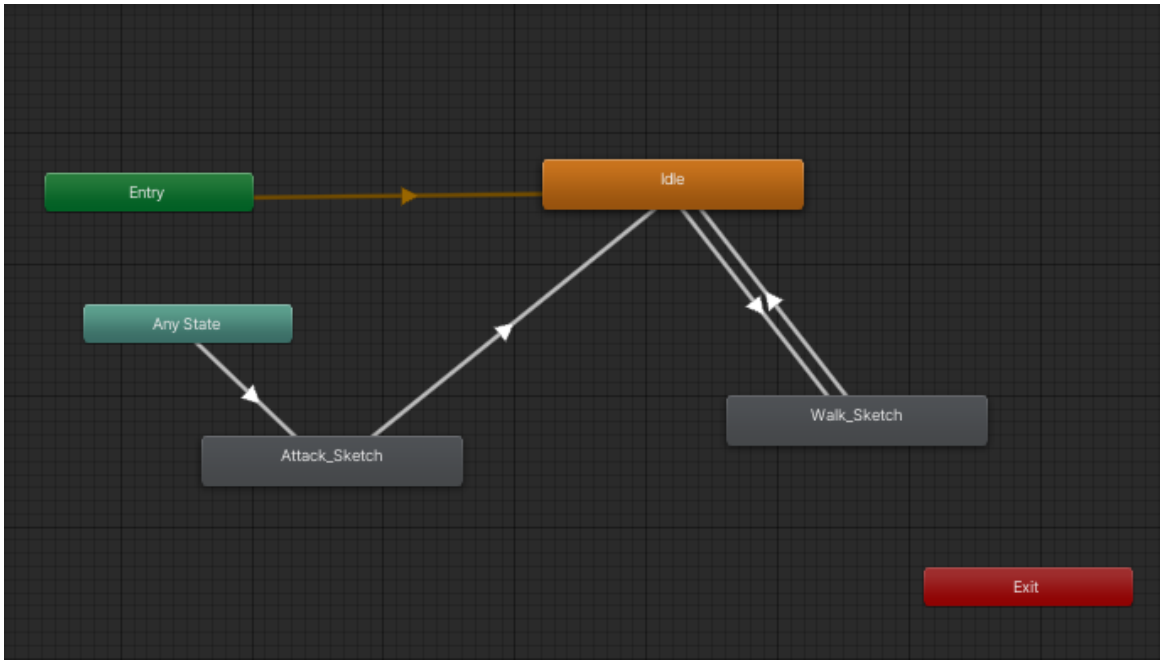

Figure 25. Unity's animator window (Unity Technologies 2005)

To test the animations, a simple player controller script needed to be programmed. The player needs the ability to move to left and right to test the walking animation, as well as the ability to use an attack to test the attack animation. The programming language used by Unity is C#, so the scripts were written in C#, with help of free online resources such as tutorials.

The first test displayed no serious problems with the attack and idle animations. However, there was one issue with the walking animation. The player visibly shrank when switching from the idle animation to the walking animation, which is not supposed to happen. Comparing the idle animation with the walking animation displayed that the drawings in the walking animation were too small compared to the idle animation. However, fixing the issue at this early stage was not too difficult.

The animations were tested again after they were lined and colored. The assumption was that there would be no serious issues, as there had not been any issues with the sketches that had not been fixed. The animations worked well overall, and the gameplay of the prototype felt fluid. The animations could then be rendered for the final prototype.

While the animations themselves were mostly free from problems, the character design proved to be rather problematic. The asymmetric character design was chosen because it is visually interesting, however, the asymmetric nature of the design became a problem for the animations. When the animations were flipped, they became incorrect. The only way to fix it would be to either change the character design to be symmetrical, which would have required changing all the material for character including lined animations and concept art. The other option would be to create flipped animations in addition to the original animations. This means that every frame of every animation would have to be edited for the flipped versions. Figure 26 displays two flipped characters, the one above being incorrect, as the character's clothing changes. The one below is a more correct way of flipping, as the character's clothes do not change. Nonetheless, in the example below, the character's hair still flips incorrectly. However, the inconsistency of the hair could be explained simply by the fact that hair is flexible and can move or be moved easily compared to a piece of clothing that should not change drastically.

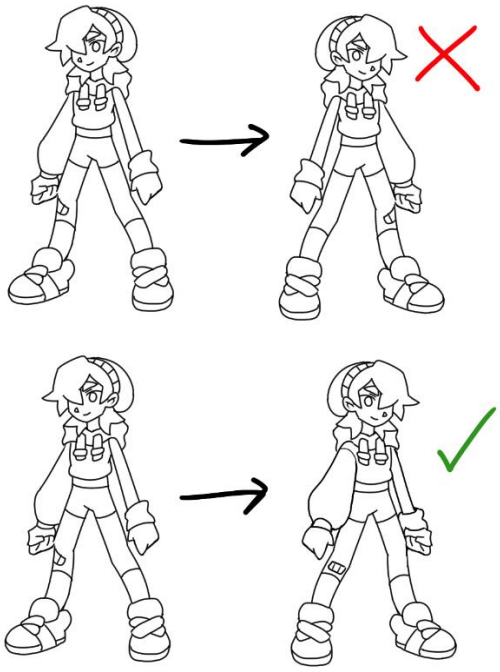

Figure 26. The above example displays incorrect flipping, while the below displays a more correct way of flipping

5 CONCLUSIONS

This thesis focuses on studying the basic animation principles, animation in video games and utilization of 2D graphics and animation in video games. The purpose was to provide enough fundamental information and instructions to be able to create basic video game animations for a potential 2D side-scroller video game. Additionally, different approaches to 2D animation in video games were explored. However, the relationship between 2D and 3D animation could have been examined further. 3D animation is undoubtedly the industry standard, and there are ways to utilize both the strengths of 2D animation and 3D animation in the same game project.

When it comes to examining other 2D video games, having access to more technical data would have been greatly beneficial. It proved to be rather difficult to find technical data, such as the exact number of frames certain animations utilized. For example, only *Cuphead*'s (2017) specific frame rate of animation was found. Additionally, it would be too difficult to try to count the frames from the gameplay, because the gameplay often runs at 60 FPS, even if the animations itself were created in other frame rates such as 30, 24 or 12 FPS.

While Clip Studio Paint provides the basic animation tools that made the implementation of the animation project possible, further analysis on whether Clip Studio Paint is a suitable software for game animation should include comparison to other relevant 2D animation software, such as Aseprite, Spine, Toon Boom Harmony, and Adobe Animate. Additionally, Clip Studio Paint still lacks some crucial features for video game animation, such as exporting animations directly as a sprite sheet. While it is not too difficult to use an external program for creating sprite sheets, it would be easier if the animation program itself allowed for the exportation into sprite sheets.

The character design for the animation project should have been better thought out. Too much focus was put on whether the character design was visually interesting, rather than how well it would work in an animation. The asymmetrical design proved to be troublesome, as it created multiple problems when the

animations were flipped in the game engine. Additionally, the character design could have been further simplified. While it was not too difficult to create three rather simple animations for this project, in a real video game there would most likely be a greater number of animations.

While the animations did their job, they could have been more polished. All the animations had at least a small number of issues with inconsistency, such as with the proportions of the character. This issue could have been avoided by more carefully planning the animation, paying more attention to the inbetweens, and tracking the proportions individually (by coloring the proportions differently from each other, for example). Additionally, the walk animation felt too stiff when the aim was for it to feel smooth and bouncy.

REFERENCES

Agala, J. & Bouthillier A. 2014. 2D Animation at Klei Entertainment. Game Developers Conference 2014 lecture. Available: https://www.youtube.com/watch?v=8_KBjd0iaCU&list=PL2e4mYbwSTbbHAJT7OdK5mv-idhlLOTl0&index=68 [Accessed 31 January 2023]

Autumn Games. 2012. Skullgirls. Video game. New York: Autumn Games.

Besen E. & Hallett B. 2008. Animation Unleashed: 100 Principles Every Animator, Comic Book Writer, Filmmaker, Video Artist, and Game Developer Should Know. Los Angeles: Michael Wiese Productions.

Brown, M. 2019. Why Does Celeste Feel So Good to Play? Game Maker's Toolkit. Video Essay. 31 July 2019. Available at: https://www.youtube.com/watch?v=yorTG9at90g [Accessed 9 February 2023]

Cani, M., Guay, M. & Ronfard, R. 2013. The Line of Action: An Intuitive Interface for Expressive Character Posing. ACM Transactions on Graphics, 32 (6). Article. PDF document. Available at: https://www.researchgate.net/publication/256979808_The_Line_of_Action_an_Intuitive_Interface_for_Expressive_Character_Posing [Accessed 2 February 2023]

Cartwright, M. 2014. Making Fluid and Powerful Animations For 'Skullgirls'. Game Developers Conference 2014 lecture. Available: https://www.youtube.com/watch?v=Mw0h9WmBlsw [Accessed 31 January 2023]

Cartwright, M. 2011. On Animation in Skullgirls. Blog. April 19 2011. Available: https://skullgirls.com/2011/04/on-animation-in-skullgirls/ [Accessed 27 March 2023]

Celsys. n.d.a Animation: Animate your vision. Available: https://www.clipstudio.net/en/animation/?from_features#func_01

[Accessed 10 March 2023]

Celsys. n.d.b What is the difference between Clip Studio Paint PRO and EX? Available: https://support.clip-studio.com/en-us/faq/articles/20200186 [Accessed 10 March 2023]

Clark J. 2017. Cuphead's Animation Process and Philosophy. Game Developers Conference 2017 lecture. Available: https://www.youtube.com/watch?v=RmGb-jU3uVQ [Accessed 31 January 2023]

Cooper, J. 2021. Game Anim: Video Game Animation Explained.

Abingdon-on-Thames: Taylor & Francis Group. E-book. Available at: https://ebookcentral.proquest.com/lib/xamk-ebooks/detail.action?docID=6469123 [Accessed 31 January 2023]

Culhane, S. 1988. Animation: From Script to Screen. New York: St. Martin's Griffin.

Floyd, D. 2020. Why Mario's Triple Jump Animation Is So Good. New Frame Plus. Video essay. 22 May 2020. Available at: https://www.youtube.com/watch?v=zCxpzIiUJIg&t=313s [Accessed: 9 February 2023]

Floyd, D. 2019a. How Link's Climbing Animation Works in Breath of the Wild. New Frame Plus. Video Essay. 21 August 2019. Available at: How Link's Climbing Animation Works in Breath of the Wild - YouTube [Accessed 13 March 2023]

Floyd, D. 2019b. The Effects Animation of Hollow Knight. New Frame Plus. Video Essay. 7 November 2019. Available at: https://www.youtube.com/watch?v=SIJtfr-PO4Y&t=178s [Accessed 9 February 2023]

Gibson, A. 2017. Co-Founder & Co-Director of Team Cherry. Interview. 14 October 2017. Team Cherry. Available at: https://www.framefreakstudio.com/ari-gibson-hollow-knight/ [Accessed 1 February 2023]

Gibson, A. & Pellen, W. Founders and Directors of Team Cherry. Interview. 15 October 2018. Team Cherry. Available at:

https://www.gameinformer.com/2018/10/16/the-making-of-hollow-knight [Accessed 6 February 2023]

Halas, J. & Whitaker, H. 2009. Timing for animation. Amsterdam: Elsevier.

Unity. n.d.a. Cuphead by Studio MDHR. Web page. Available: https://unity.com/madewith/cuphead [Accessed 27 March 2023]

Unity. n.d.b. Hollow Knight by Team Cherry. Web page. Available: https://unity.com/madewith/hollow-knight [Accessed 6 March 2023]

Klappenbach, M. 2022. Understanding and Optimizing Video Game Frame Rates. Lifewire. 5 November 2022. Technology information and news website. Available at: Understanding and Optimizing Video Game Frame Rates (lifewire.com) [Accessed 1 February 2023]

Mobygames. n.d. Groups > Game Engine: Z-Engine. Electronic games database. Available: https://www.mobygames.com/group/13447/game-engine-z-engine/ [Accessed 27 March 2023]

Moldenhauer, C. & Moldenhauer J. 2017. Cuphead goes platinum. Blog. 13 October 2017. Available at: http://studiomdhr.com/cuphead-goes-platinum/ [Accessed 31 January 2023]

Moldenhauer C. 2016. Co-Director and Lead Artist of Cuphead. Interview. April 12 2016. Studio MDHR. Available at: https://time.com/4123150/cuphead-preview/ [Accessed 13 April 2023]

Moreno, L. 2014. The Creation Process of 2D Animated Movies. PDF document. Available at: https://dl.jalebamooz.com/upload/2014-2015-02-4-TR_baixa.pdf [Accessed 1 February 2023]

Nintendo. 2020. Animal Crossing: New Horizons. Video game. Kioto: Nintendo.

Nintendo. 2017. The Legend of Zelda: Breath of the Wild. Video game. Kioto: Nintendo

Nintendo. 1996. Super Mario 64. Video game. Kioto: Nintendo.

Pantoja, T. 2021. How I Color my Animation Work. Toniko Pantoja. Video tutorial. 4 January 2021. Available at: https://www.youtube.com/watch?v=eG9hBIweaOE [Accessed 28 March 2023]

Roberts, S. 2011. Character animation fundamentals: developing skills for 2D and 3D character animation. Amsterdam: Elsevier.

Root, D. 2020. Game Animation Fundamentals: Elegance. Video Game Animation Study. Video Essay. 1 April 2020. Available at: https://www.youtube.com/watch?v=ma4SoiEcKwQ [Accessed 16 March 2023]

Studio MDHR. 2017. Cuphead. Video game. Oakville: Studio MDHR.

Team Cherry. 2017. Hollow Knight. Video game. Adelaide: Team Cherry.

Williams, R. 2001.The Animator's Survival Kit. London: Faber & Faber.

LIST OF FIGURES

Figure 1. Drawing illustrating the timing and spacing of an animation. Williams, R. 2001. The Animator's Survival Kit. London: Faber & Faber.

Figure 2. Drawing illustrating animation frame positions. Williams, R. 2001. The Animator's Survival Kit. London: Faber & Faber.

Figure 3. Illustration of the "squash and stretch"-principle. Lampman. 2008. Image file. Available at: https://en.wikipedia.org/wiki/Squash_and_stretch#/media/File:Squash_and_Stretch.svg [Accessed 24 January 2023]

Figure 4. Different frames of jump animation, including anticipation. Williams, R. 2001. The Animator's Survival Kit. London: Faber & Faber.

Figure 5. Illustration displaying the line of action and lack of it. Blair, P. 1994. Cartoon Animation. California: Walter Foster Publishing. Available: https://ia600700.us.archive.org/17/items/PrestonBlairCartoonAnimation/Preston%20Blair%20-%20Cartoon%20Animation.pdf [Accessed 6 February 2023]

Figure 6. Illustration displaying the arc of motion and lack of it. Williams, R. 2001. The Animator's Survival Kit. London: Faber & Faber.

Figure 7. Knight’s attack effect animation. Hollow Knight. Team Cherry. 2017. Screenshot from: https://www.youtube.com/watch?v=SIJtfr-PO4Y&t=178s [Accessed 9 February 2023]

Figure 8. Screenshot of a lecture dia show from GDC, displaying Skullgirls character sketches and their silhouettes. Cartwright, M. 2016. Making Fluid and Powerful Animations For 'Skullgirls'. GDC 2014 lecture. Available: https://www.youtube.com/watch?v=Mw0h9WmBlsw [Accessed 6 February 2023]

Figure 9. Animal Crossing's fishing animation. Animal Crossing: New Horizons. Nintendo. 2020. Screenshot from: https://www.youtube.com/watch?v=ma4SoiEcKwQ [Accessed 23 March 2023]

Figure 10. Screenshot of a lecture dia show from GDC, displaying sketches of a Cuphead boss. Clark J. 2017. Cuphead's Animation Process and Philosophy. GDC 2017 lecture. Available: https://www.youtube.com/watch?v=RmGb-jU3uVQ [Accessed 6 February 2023]

Figure 11. Screenshot of a lecture dia show from GDC, displaying the animation process of Skullgirls. Cartwright, M. 2016. Making Fluid and Powerful Animations For 'Skullgirls'. GDC 2014 lecture. Available: https://www.youtube.com/watch?v=Mw0h9WmBlsw [Accessed 6 February 2023]

Figure 12. Screenshot from Hollow Knight displaying the animation which plays when the player starts walking. Hollow Knight. Team Cherry. 2017.

Figure 13. Screenshot from Hollow Knight displaying the animation which plays when the player is walking. Hollow Knight. Team Cherry. 2017.

Figure 14. Screenshot from Hollow Knight displaying the animation which plays when the player stops walking. Hollow Knight. Team Cherry. 2017.

Figure 15. Screenshots from Cuphead displaying the player character's special attack. Cuphead. Studio MDHR. 2017.

Figure 16. Character sprites from Skullgirls. Skullgirls. Autumn Games. 2012

Figure 17. Finalized character design

Figure 18. Screenshot of Clip Studio Paint EX's file creation. Celsys. 2001

Figure 19. Screenshot of Clip Studio Paint EX's timeline feature. Celsys. 2001

Figure 20. Animation with onion skin

Figure 21. Animation workflow for animation project

Figure 22. Idle animation frames

Figure 23. Running animation frames

Figure 24. Attack animation frames

Figure 25. Screenshot of Unity's animator window. Unity technologies. 2005.

Figure 26. Two ways of flipping a 2D character

ANIMATION PROJECT FRAME COUNTS

Idle	4
Run	12
Attack	7