Karelia University of Applied Sciences
Bachelor of Culture and Arts

The Essence of Movement

Animation Principles and Fundamentals for Game Character Production

Väinö Eemeli Koivumaa

Thesis, November 2023

www.karelia.fi

	THESIS **November 2023** **Degree Programme in Culture and Arts** Tikkarinne 9 80200 JOENSUU FINLAND + 358 13 260 600 (switchboard)

Author
Väinö Eemeli Koivumaa

Title
The Essence of Movement: Animation Principles and Fundamentals for Game Character Production

Abstract
The process of producing animations for a video game includes multiple stages. The principles of animation and fundamentals of game animation are utilized in order to bring clarity to animation production. Their implementation is a balancing act between visual feedback, responsiveness and the non-linear story telling form of video games.

To test the effectiveness of the fundamentals of game animation and how the implementation of principles of animation could appear in game animation, they are used in the production process of a playable video game character. The animation process was documented, and the documentation demonstrates how problem solving in game animation is conducted by following the game animation fundamentals. The animations were tested within a game engine for their practical application. This practical test was intended to indicate how incorporating problem solving, production methods, and game animation fundamentals can lead to quality results.

The results display the challenges to be considered in future productions and the game animation fundamentals and the animation principles. Flexible use and interpretation of the animation theories can be creatively used to produce intended results for game development.

Language English	Pages 68 Appendices 0 Pages of Appendices 0

Keywords
Animation, character animator, technical animator

OPINNÄYTETYÖ
Marraskuu 2023
Media-alan koulutus

Tikkarinne 9
80200 JOENSUU
+358 13 260 600 (vaihde)

Tekijä
Väinö Eemeli Koivumaa

Nimeke
Liikkeen ydin: Animaatioperiaatteet pelihahmon animaatiotuotannossa

Tiivistelmä

Animaatioiden tuotantoprosessi videopeleihin sisältää useita vaiheita. Animaation periaatteita ja pelianimaatioiden perusteita käytetään selkeyttämään animaatioiden tuotantoa. Niiden toteutus tasapainotetaan visuaalisen palautteen, reagointikyvyn ja videopelien epälineaarisen tarinankerrontamuodon välillä.

Animaatio perusteiden tehokkuuden ja käytön testaamiseksi pelianimaatiossa, niitä käytettiin pelattavan videopelihahmon tuotantoprosessissa. Animaatioprosessi dokumentoitiin ja dokumentaatio osoittaa, kuinka pelianimaatioiden ongelmanratkaisu tapahtuu pelianimaatioiden perusteita noudattaen. Animaatioita testattiin pelimoottorissa niiden käytännön soveltamiseksi. Tämän käytännön testin tarkoituksena oli osoittaa, kuinka ongelmanratkaisun, tuotantomenetelmien ja pelien animaation perusteiden yhdistäminen voi johtaa laadukkaisiin tuloksiin.

Tulokset osoittavat tulevissa tuotannoissa ja pelianimaatiossa huomioitavia haasteita ja animaation periaatteet. Animaatioteorioiden joustavalla käytöllä ja tulkinnalla voidaan luovasti tuottaa haluttuja tuloksia pelin kehittämiseksi.

Kieli Englanti	Sivuja 68 Liitteet 0 Liitesivumäärä 0

Asiasanat
Animaatio, hahmo animaattori, tekninen animaattori

Contents

Terms and abbreviations

2D	Two dimensional. Refers to video games where two axis of motion are possible
3D	Three dimensional. Refers to video games where three axis of motion are possible
Inbetweens	Frames in between keyframes to make motion smoother.
NPC	Non-player character. An avatar in a video game which the player does not control. Often run by code to accomplish tasks, such as attacking the player.
On twos	Animation frames where every other frame is a new drawing.
PC	Player character. The avatar which the player controls. A graphical representation of a character through which the player interacts with the game.
Rig	A system of digital bones and joints used to animate 3D models.
Sliding	In game animation this means the PC is not moving or stopping at a proper speed related to the ground.
Sprite	A two-dimensional graphic that is integrated into a large scene in a game engine.

1 Introduction

In this thesis, animations are produced to demonstrate game character animations adhering to the relevant principles of animation and fundamentals of video game animation. To test the effectiveness of the animations in gameplay, the aim is to import and code them into a game engine as a functional playable version. After the testing in a game engine, the results are discussed, and possible improvements are suggested.

The practical part of the thesis covers the documentation of the examples, methods, and arguments for professional development methods in game character animation. The key concepts in game design that correlate with game animation are included in the scope of this study. In addition, character design, gameplay, and environment design are discussed and explained. However, the sources focus solely on the working methods and challenges viewed from the perspective of an animator.

The perspective of a technical animator, an animator which implements animations into gameplay and does not necessarily create them, is also studied. The skills of a technical animator require skills to consider the usefulness of the animations from the gameplay perspective and how the animations work in practice within the game. This is researched in the practical testing portion of the project where the chosen methods of animation are viewed from the technical animator’s perspective.

The results and methods can aim to show the animator and readers how to improve their productions and skills. What mistakes can happen and how to avoid them is demonstrated. Examples of how an animator should approach problem solving in animation are presented. The detailed overview on how animations can be designed within a project strives to keep the merging of animation and gameplay as a core aspect.

2 Theory of animation principles

In the book The Illusion of Life: Disney Animation by Frank Thomas and Ollie Johnston (1981), the writers reflect on their experiences at Disney. They explain their learning process and discoveries in animation production, which at the time was a new industry. The present animation standards were defined by these early pioneers working at Disney Studios. The 12 animation principles are known as the following: squash and stretch, anticipation, staging, straight-ahead and pose to pose, follow through and overlapping action, slow in and slow out (ease-in and ease-out), arcs, secondary action, timing, exaggeration, solid drawing, and appeal were born during the development and research stage into the animation art form within the studio.

For the development of video game animation, some animation principles are less relevant than others and are not covered by the thesis. The principle of solid drawing, pose to pose, and straight ahead are covered by other theories presented within the production in a more detailed manner. Therefore, explaining them was seen as redundant.

The principle of staging is mainly designed to assist animating in a linear storytelling form. Because video games represent a form of nonlinear storytelling, the staging principle theories do not apply to gameplay serving video game animations. The staging principle can still be applied to cutscenes in video games, which are linear story-telling moments within a video game. However, due to cutscenes not being a part of this production, the staging principle was left out.

2.1 Appeal

Appeal (Johnston & Thomas 1981, 69) is the act of making the design and motion of the animated character pleasing to look at. Pleasing, in this context, refers to qualities a person prefers to see in a character. What these qualities are depends on the emotion the character wants to evoke in the viewer. The qualities could be, but are not limited to, fear, charm, joy, or strength. (Johnston

& Thomas 1981, 69–70.) The principle of appeal does not only contain animation, but character design and color theory as well.

Appeal means that instead of having morally appealing features, the character should have interesting and entertaining ones. This principle encompasses the design, the writing, the personality, and the colors of the character. It is the ability of the animator to portray a believable character. (Cooper 2019.)

Appeal as a subjective concept and a term within this thesis must be noted, however. The personal preferences of the author of this thesis must be taken into consideration when arguments made for appeal are made. When the term appeal is used, it does not only refer to the subjective opinion of the author, but animating motion and character which the viewer interoperates in the way the animator intended (Cooper 2019).

2.2 Squash and stretch

Shapes or objects that have flexibility often compress and stretch as they move. Objects change their shape according to their movements and the force exerted to accomplish actions with them. *Squash and stretch* (Johnston & Thomas 1981, 47) refers to the compression and the elongation of an object to accentuate its speed, momentum, mass, and weight. (Johnston & Thomas 1981, 47–48.) The amount of squashing and stretching also determines the firmness of an object. The more the objects squashes and stretches, the softer the objects seem. If there is little squash and stretch, the object may seem harder or stiffer. (AlanBeckerTutorials 2017.)

According to Johnston & Thomas (1981, 49) the concept of squash and stretch was referred to as “the half-filled flour sack.” If a half-filled sack was dropped, the contest would spread inside the sack stretching to its fullest form. If the sack was picked up by a corner, it would elongate to its most vertical shape possible. Yet the weight or the amount of content within the sack never changes. Meaning, if the object stretches vertically, a corresponding squash horizontally must occur to maintain volume. (Cooper 2019.) This change in shape brings clarity to the movement and force by distorting the established shape of the object.

2.3 Anticipation

Unless there is a planned sequence of actions that leads the viewer, following an animated scene or an action tends to be challenging. A preparation for the next movement should be made so it can be expected. (Williams 2001, 274). This animation of *anticipation* informs the viewer of the upcoming major movement. (Johnston & Thomas 1981, 51–53.) It also brings power to an action as it aims to depict reality and follow the laws of physics we are accustomed to. As supported by Johnston & Thomas (1981, 53): “Few movements in real life occur without anticipation. Without it there would be little power in any action.” An action can have multiple layers of anticipation to bring even more clarity to the movement (White 2006). These can be referred to as different levels of anticipation such as *second anticipation* (AlanBeckerTutorials 2017).

In Cooper’s (2019) experience: “anticipation is a controversial subject for game animation.” Anticipation requires a character to hold a pose or prepare an action for a certain number of frames. If this anticipation is started after the player has pressed a button, the character might be late to achieve the player’s wanted outcome. This may lead to an unresponsive avatar, reducing the feeling of control. (Cooper 2019.)

Anticipation should be included in some form. Otherwise, movement can feel weightless and unnatural. However, long anticipation can also be used to create engaging gameplay. If the player needs to take the long anticipation of a move into account, they must wait for the right moment to attack by observing the enemy and actively learn their moves, as explained by Daniel Floyd from New Frame Plus (2020a). *NPCs* (non-player character) receive longer anticipations for this reason. To give players reaction time to NPC actions, such as attacks, their anticipations are increased, and motion exaggerated. (Cooper 2019.) This is to give the player enough time and visual feedback to react accordingly.

A long anticipation is not the only way of giving people enough reaction time. In some games, like *Star Wars: Jedi Fallen Order* by *Respawn Entertainment (2019),* players only have a few frames to press a button to counter an attack, according to Nathan Dunlap, lead animator at Respawn Entertainment, during his lecture at AnimationExchange (2021a). This means that a long anticipation is not a sufficient way of giving the player visual feedback to react to an attack. They can tell an attack is coming, but not exactly when. To address this, the enemies were given a second anticipation right before the attack. This second anticipation would signal the player when they need to counter. Anticipation can also be used to trick the player by not delivering the attack when the anticipation ends creating more variety and challenge to a game. (AnimationExchange 2021a.)

2.4 Follow-through and overlapping action

Not all parts of the character arrive at the destination at the same time. They move at different rates as the body catches up with itself. Some parts continue moving after the main action has been completed and can still interact with the character. This is referred to as *overlapping.* (Johnston & Thomas 1981, 60–62; Williams 2001, 226–230.)

Follow-through (Johnston & Thomas 1981, 60) means the continuation of movement after the desired stopping position has been reached. Natural movements rarely stop on the exact spot where they attempt to settle. Instead, they continue forward before returning to the desired settling position. Without the follow-through, movement would seem unnaturally accurate or stiff. (Johnston & Thomas 1981, 60–62.)

Follow-through is a way of expressing the weight of a character and the power of an action. This is because the main action has caused the character or object to move further than intended through the force of the action. It seems to have fewer gameplay limitations since the main action the player wanted to accomplish with a button press has already been accomplished. However, it can still result in an unresponsive character if the follow-through is too long. (Cooper 2019.)

The solution for unresponsiveness stemming from follow-through is to give the player control before the follow-through animation finishes. This cancelation provides an option to move to the next action fluently. It should be noted that follow-through returns the weight of an action, which was lost due to lack of anticipation (Cooper 2019; New Frame Plus 2021). Meaning in game animation, weight and power should be delivered through follow-through rather than anticipation when possible.

Overlapping could be used to demonstrate which part of the body is driving the motion and giving the action the force. The body parts that move first, such as the hands or the torso, are the source of force within the motion. Other body parts like hair, clothes, and the head arrive to their destination later. This delay signals that they were not the source of the force. They were dragged along by the body giving the action more power as it moves all the other body parts with it. (New Frame Plus 2021.)

2.5 Slow in and slow out (ease-in and ease-out)

Slow in and *slow out* (Johnston & Thomas 1981, 62; Williams 2001, 50) is a way to depict more natural movement. Most motion in nature starts off slow and speeds up before slowing down. Slow out means that the motion begins slower and eventually speeds up to full acceleration before slows in, meaning the motion slows down before it stops. (Johnston & Thomas 1981, 62.)

Character movement seems to have more life when it does not move at a linear speed from one pose to the next, but instead lingers on key poses by slowly moving out of them and slowly arriving at a new key pose, according to Danie Floyd (Johnston & Thomas 1981, 62; New Frame Plus 2020). This may be because of the laws of physics that movements in nature abide by. Slow in and slow out should not only be implemented to specific body parts of a character, but their key poses as well. Allowing a character to linger in key poses through the motions can make the action more readable and clearer with correct amount of ease-in and ease-out. (New Frame Plus 2020.)

Cooper (2019) views slow in and slow out as a necessary principle but reminds of its harmful effect on responsiveness: “Objects that burst into full speed immediately can look weightless and unrealistic, so again there is conflict between the gameplay desire to give players immediate response versus the artistic desire to give weight to a character” (Cooper 2019). It is the animator’s task to add weight at the end in follow-through and find a balance in the slow in & slow out without reducing responsiveness. Depending on the project slow in and slow out needs to be finely tuned to serve the gameplay.

2.6 Arcs

Movements in nature follow circular curved paths referred to as *arcs* (Johnston & Thomas 1981, 62). They do not move linearly between positions. Hence, most movements should follow a clear arc path on their way to their destination to achieve an appealing depiction of reality (Johnston & Thomas 1981, 62–63). It should be taken note that this principle can be broken if the animator wants to create unnatural movements. Robots, as an example, do not necessarily follow arcs, but precise movements with little ease-in or ease-out.

Cooper (2019) has a different interpretation of this principle: characters can seem floaty if every element on them has been animated to follow a clean arc. (Cooper 2019.) When arcs are broken on specific parts of the motion it can add interesting movement and stability to a character. Such as: “the head breaking from the arc after a punch” (Cooper 2019.) Carefully studying movements in nature and human movements can reveal when the body breaks the traditional arc path. These breaks should be depicted accordingly to avoid weightless feeling characters. There does not appear to be a single rule as to when arcs should be broken. By studying reference footage of different movements, a moment for breaking an arc may be found and applied to the animations within the production.

2.7 Secondary action

A secondary action (Johnston & Thomas 1981, 63) is an action that is animated to support the main action to bring more appeal and interest to an action. Secondary action is often something small compared to the main action and is registered subconsciously. An animator should be careful not to make the secondary action too distracting from the main action. (Johnston & Thomas 1981, 63–64.) Yet it should not be easily distinguishable. A clear unification between a secondary action and the main action melds them both together. An example of a secondary action are expressions that reveal the emotions of a character when the rest of their body is used for the main action (Sloan 2015).

In the opinion of Sloan (2015) secondary actions, especially facial expressions, are a requirement in animation. A character often feels robotic or plastic-like if they do not change their expression when throwing a punch or committing any other physical exertion. (Sloan 2015.) It would signal to the player that the character is not feeling anything and makes the performance feel flat. Because of this, expression changes according to actions and secondary actions are considered as a requirement in this thesis.

2.8 Timing and spacing

Timing (Johnston & Thomas 1981, 64; Williams 2001, 37–38.) means the number of frames between poses, which translates to frames per second based on the format being used. Daniel Floyd (New Frame Plus 2019a) describes it as "the measuring of change over time" or "the speed or tempo at which an action takes place". Timing can affect an action's speed, force, impact, subtlety, and weight. Characters with different weights, shapes, and feelings move at different paces and require more frames to complete an action. (White 2006.)

Timing is utilized to convey the weight and size of a character or object (Cooper 2019; New Frame Plus 2019a). While it is linked to speed, the time it takes for an object, or individual limb, to move or rotate a distance affects the feeling of weight. A large object moving slowly seems heavy while a small fast-moving object appears light. A large heavy creature standing up could have slow movements taking place during several frames to convey the size and weight of the creature (New Frame Plus 2019a).

Timing also affects the emotion and intent behind actions. As an example, a character quickly turning their head can seem surprised or frightened compared to a slow turn which may seem attentive or tired. More frames can slow down the movement giving it a different meaning compared to a few framed fast movements. (Johnston & Thomas 1981, 64–65.)

Daniel Floyd (New Frame Plus 2019a) and Richard Williams (Williams 2001, 39) explain an animation concept relating to timing called *spacing*. Spacing describes the distance an object moves or changes between frames. The movements of the object appear quicker the larger the changes in the positioning of the object are. (New Frame Plus 2019a.) This concept of spacing is used to create ease-in and ease-out. However, it does not limit itself to the idea of spacing the objects only one specific way but describes the action of spacing the objects in a multitude of different ways to give the movement variety. The object will still have the same timing, meaning the first- and last frame of animation will end at the same position (Williams 2001, 37–38.)

2.9 Exaggeration

Exaggeration (Johnston & Thomas 1981, 65) is the concept of embellishing movement, deformation, and expressions to create appeal. The amount of exaggeration depends on the animator. Yet it would seem the general rule in animation and exaggeration is to depict reality in a more noticeable form. However, Tony White points out that *extreme poses,* where the form of a character is stretched to the limit, should only be used in fast, sudden, or impactful actions. It should also be noted that these exaggerations or extreme poses should be registered subconsciously by the viewer, meaning they should not be held for too many frames. (White 2006, 220–221.)

The human brain seems to register the motion and how it feels without realizing how much the original form of the character was broken. The difference can be noticed by viewing playbacks of animation with and without extreme poses while comparing them. Analyzing animation depicting exertions of force frame by frame reveals how much exaggeration and deformation takes place. This proves how many extreme poses are not consciously noticed by viewers and how they subconsciously affect our perception of motion.

The goal of animation is to create the most appealing version of reality. In video games, extra care must be taken with exaggeration due to the ability of the player to view this action from all angles. It must be ensured that the amount of exaggeration remains equal throughout the project. (Cooper 2019.)

3 Theory of game animation fundamentals

In addition to the general animation principles, Jonathan Cooper (2019) suggests a set of fundamentals unique to game animation. Instead of replacing the original 12 animation principles, he aimed to compliment them. These *five fundamentals of video game animation* are meant to clarify the methods and goals of video game animation. (Cooper 2019.) The fundamentals function as a foundation for game animators to follow and are referred to as:

1. Feel
2. Fluidity
3. Readability
4. Context
5. Elegance

The fundamentals of game animation are set as the guidelines and their testing is a goal for this thesis. The criterion set by the author for successful production is to create animations which serve the gameplay and follow the fundamentals of game animation. The purpose of this is to strive towards professional level game animation standards.

3.1 Feel

The main difference between traditional linear animation and game animation is interactivity. Players control and modify avatars by making decisions in the span of seconds. This forces the animator to forfeit complete authority over how the story within the game is told and experienced. (Cooper 2019.)

Uninterrupted animations block the decision-making process from the players. This creates brief moments when the player is not experiencing the interactivity of a game. These brief moments of lack of control need to be kept to a suitable duration depending on the project. (Cooper 2019.) Some games need as much control as possible while others function with the player observing the animated actions for a longer period. The loss of control can also be used to lock the player into an action they have made. However, the player should always attain the feeling of being an active participant in the game.

The illusion of the player being an active participant in the game is maintained by having a low *input response time* (Cooper 2019). It refers to the time between the player pressing a button and the character performing a desired action within the game. The input response time between the player pressing a button and the in-game action is an inherent part of how the control of a character and interaction in the game is perceived by the player. The ideal scenario is to keep the response time to as few frames as possible. It should be noted that this is conditional to the context of the action. (Cooper 2019; New Frame Plus 2020a.)

A balance between reaction time and appealing visual feedback is referred to as *feel* (Cooper 2019*)*. It dictates how controlling the character and the elements within the game feels to the player. The appealing visual feedback must retain the action and the character. This means that sacrificing all weight of a character to make everything as responsive as possible tends to lead to floaty unappealing movement, which is difficult to follow and respond to. Therefore, making a character as responsive as possible by not taking visual feedback into account does not lead to a sufficient feel. Feel should be considered the main attraction and uniqueness in video games as an entertainment form. Thus, a gameplay serving approach for animation should often be assumed. (Cooper 2019.)

A connection between the fundamental of feel and the principles of timing, ease-in, ease-out, and anticipation is evident. All three principles relate back to how many frames are spent on animations, and directly affect the response time and the feel. Taking precautions to design each animation through these principles may ensure the optimal feel for the player controlling the avatar.

Ensuring the accuracy of the timing of animations is not enough to achieve the perfect feel for a character. The feel of the actions the player and the PC perform are constructed through designing accurate and appealing visual feedback for the action. (Cooper 2019). The purpose of the animations is to tell the player what is happening around them and what their avatar is experiencing. If the player is not informed that they have been damaged through any visual feedback, the death of the avatar may feel unsatisfying, since the player had no way of reacting (Cooper 2019). Thus, damaged animations and other UI elements are used to give the player information. The visual feedback transmitted through animations should always tell information to the player. Such as the power behind a thrown punch or how heavy a crate is when a character lifts it. A short animation can transmit information to the player. This information could be something relating to interaction or how the avatar experiences the environment.

Since many of the feel fundamentals are connected to the process of controlling the character and how the response time of the player is affected by the animations, a process of demoing each animation within a controllable version in a game engine before finalization is required. This grants the ability to observe the functionality of the feel and that the animations fulfill the criteria to reach a proper feel. Through this demoing process, the project does not only work on a theory base, but also has a practical element introduced by applying the animations into a controllable character.

3.2 Fluidity

Early 2D & 3D games *snapping*, meaning transitioning directly from animation to the next without any link- or bridging animation, was a common occurrence. In joint-based animation today, it is possible to blend transitions between these animations by using numerical values. This tends to make the actions more visually appealing through the flow of movement. (Cooper 2019.)

Cooper states that video games are formed of several animations playing incessantly in patterns by moving and overlapping between each other (Cooper 2019). Dan Root in his video essay from Video Game Animation Study (Video Game Animation Study 2020c) defines these animations as *Loops, Links,* and *Linears. Fluidity* refers to the change between these animations and their functionality in unison (Cooper 2019).

Looped animations are regularly used in video games due to the amount of repeated actions (Cooper 2019). Loops are animations of continuous motion which must return to their starting frame seamlessly to create a loop within the motion (Video Game Animation Study 2020c). In addition, the momentum on each body part must be preserved and the animator should refrain from giving rhythm-breaks to the movement. Rhythm-breaks refer to details within the movement that break the flow of motion and bring attention to a certain part of the movement. This could make spotting the loop too apparent and break the immersion for the player. (Cooper 2019.)

Links are transition animations that bridge two animations together. They can be created as their own separate animations, which play in between animations if certain conditions are met, or they can be generated through numerical values. Linears are animations that play once from beginning to end without maintaining momentum for a perfect loop or sometimes not returning into the starting frame at all. (Video Game Animation Study 2020c.)

Settling (Cooper 2019) is an animation where the character returns to their idle pose after an action without removing the control from the player. With settling animations fluidity can be preserved because the snapping back to an idle pose will be avoided. The settling animation does reduce the feel of the character, because settling animations can be interrupted by the player to activate a follow-up action. If the player does not want the avatar to do any follow-up action, this is free space for the animator to fill in with a settling animation. (Cooper 2019.)

Recovery animation, described by Daniel Floyd (New Frame plus 2020a), is a follow through animation to return the force, weight, and momentum of the action. Additionally, recovery animations are often defined by the inability to cancel them or begin any other action with the avatar until the recovery animation has finished. This creates an opportunity to add gameplay mechanics where the player needs to be aware of losing control of their avatar for a set amount of time, opening themselves to counterattacks. This implementation of recovery animations into gameplay may encourage players to observe hostile NPC animations and wait for the right moment to activate an action so that their recovery animation does not leave them vulnerable. (New Frame Plus 2020a.)

3.3 Readability

With the introduction of 3D games, the control of the camera was further removed from the animator and the ability to view the character from any angle was given to the player. This means that it is not enough to create an appealing animation from one angle. In 3D games, action must be appealing from all angles.

Readability is the fundamental of creating understandable and clear actions from all viewable angles. The goal is to ensure the viewer understands what is happening. Where the traditional animators had full control of how the viewer perceived the visuals, video games have had to adapt. (Cooper 2019.)

There appear to be several ways of ensuring an animation is appealing from all angles. One of them is ensuring your poses are dynamic and utilize all directional axis. Limiting the limbs to only travel on one or two axis could produce flat or stiff movements. (Video Game Animation Study 2020d.)

Another way of ensuring that poses are dynamic is to draw *lines of action* through the poses. Lines of action are lines drawn through a single pose to assist with visualizing and clarifying the motion. The creating lines of action that do not align with any directional axis and contrast each other between poses may promote clear readable motion. (Cooper 2019.)

The center of gravity may also improve readability (Video Game Animation Study 2020d). Center of mass refers to where most of the character's weight is located. This is often found in the ligament the character is supporting themselves the most with. (Cooper 2019.) Knowing where the center of mass is, helps the animator pose the character clearly and in turn improve readability. (Video Game Animation Study 2020d.)

Breaking the silhouette of a character may create clear movement, according to Dunlap (AnimationExchnage 2021a). Any part of a character breaking away from the silhouette draws attention since it moves in the prereferral of the viewer. This can lead to a way of giving the player feedback on when an attack is about to land, or a different action is about to take place. (AnimationExchange 2021a.)

The readability fundamental is relevant in all game animation but based on the explanation of Cooper (2019) it leans towards 3D. This production is done in 2D. There is no need to consider alternative camera angles when producing animations other than the one set at the stage of drawing the sprite. Therefore, in this production, following dynamic poses, center of gravity, and clear silhouette is enough to achieve proper readability based the definitions of Cooper (2019).

3.4 Context

Context is the fundamental of keeping the motion appealing for the environment, story, and emotions (Cooper 2019). If the character is running with the default running animation while being chased by a wave of lava, the action can feel unfitting for the scenario. This may then lead to the unfitting animation taking the player out of the experience. Ensuring that animations satisfy the character, environment, and action should be maintained by the game animator.

When creating motion for a character the animator should know if it is for a specific character or several characters, both playable and none-playable. This approach is referred to as *Distinction vs Homogeneity*, defined by Cooper (2019). If the animator knows the animation they are producing is for a specific character, they should strive to imbue the personality of the character into the animation. If the animation is going to be used on multiple characters or shared by a group of NPCs, the animations should be much more generic. This appears to help the playable avatar character to remain distinct and unique. It also saves time in animation production. (Cooper 2019.)

A single character was used in the production solely designed for the testing purposes of the animation principles and game animation fundamentals. The backstory, personality, and traits were in full knowledge of the animator. This granted the animator the ability to imbue personality into the movements of the character. The animator had full control of the context of the environment, story, and character, which is not always possible in game production with multiple developers.

3.5 Elegance

Elegance refers to the idea of using what animations you have for the game to create the most fluid seamless experience possible. Every unique aspect of character-based gameplay requires a system to play back animations, be this either for navigation or interaction. Consequently, creating a system which allows the utilization of the same animations repeatedly. (Cooper 2019.)

Root (Video Game Animation Study 2020b) points out that a system in this context does not only mean the animation playback program or animation blending programs. It also refers to the overall design of objects, environments, and characters in the game. For example, if a character is frequently picking up objects. Same *picking up animation* could be used on many objects if their size is generally the same in-game. Similarly, if the character climbs and grabs ledges this could allow the level designers to create ledges and walls of only a few certain heights. Then animators only need to create a few climbing animations that may be used to traverse the entire map. (Video Game Animation Study 2020b.)

Since this production is not being constrained by a set deadline the fundamental of elegance may be minimally relevant. It is however not to be discarded. Using animations and visual assets again on different actions should be kept in mind. It assists in maintaining a work ethic that is representative of the current working methods in game animation. Since it is one of the goals of the author to work in this field it should be practiced.

3.6 Technical animator

The second major point for implementing a coding portion into the production is the growing need for *technical animators* in the gaming industry. During the visiting lectures on the YouTube channel AnimationExchange (2021b), Matt Courtois, senior gameplay animator at Infinityward, defines a technical animator as a person who codes a premade motion into a game engine and implements the animation into the gameplay. The opinion of Courtois is that an animator should be responsible for the gameplay animation throughout the entire production. This means they should design the gameplay aspects behind the animation, animate the motion themselves, and then code it into the game engine. (AnimationExchange 2021b.) It should be noted that the definition of technical animator changes depending on development studios. The tasks of a technical animator also depend on which studio they are employed in, but the general idea of a technical animator approaching animation from gameplay and coding perspective appears consistent. (AnimationExchange 2021c.)

Courtois (AnimationExchange 2021b) explains that the traditional production line from the perspective of game animation professionals typically moves in this order: gameplay designer, animator, technical animator, or programmer. The strength of this structure is the number of suitable candidates for each role and the ability to focus your resources on the necessary areas. However, Courtois notes a few issues with this traditional production line. He states that this method leads to less time to create gameplay since the animators are concentrating more on the actual production of animation, disregarding the perspective of gameplay. The implementation of these animations also takes longer since interdependency is required. This same requirement may lead to losing the intention of the animation.

During his lecture at GDC (2021), Ville Ruusutie, principal animation programmer from Remedy, also believes approaching animation from the perspective of a coder may lead to better gameplay. In their game *Control,* they strived to improve on their gameplay which led to them prototyping their animations with *motion matching,* a system that mathematically determines the best way for a character rig to follow a set path. (GDC 2021; Ubisoft Montreal 2020.) Motion matching and other coding-based animation systems were also used in *Half-Life: Alyx* by *Valve Corporation* (2019), according to Joe van den Heuvel (2021) a developer at Valve. Explaining these animation systems is not a part of this thesis but recognizing the need for game animators to practice their coding skills has led to it being addressed in the production stages of the thesis.

4 Conceptualizing gameplay

Preceding the initiation of animation production, the conceptualization of foundational gameplay is recommended. An idea is only that until it gets written down or drawn as a simple image stated by Cooper (2019). Creating animations without knowing their function within the game or the character will lead to homogeneity (Cooper 2019). Challenges would arise in the production method of drawn 2D animation, since they do not utilize rigs which could be transferred to 3D models. Drawing a character to complete an action without any context offered to the animator may yield outcomes characterized by homogeneity, thereby rendering the animations unusable for the requirements of the project.

Comprehensive conceptualization of all gameplay features may not be a requisite at the beginning of animation production for video games. The animator may find knowing a portion of gameplay sufficient. For instance, understanding the combat mechanics suffices for animating relevant sequences, without necessitating foreknowledge of puzzle gameplay could be added later in the production. Given the potential lack of interaction or conflict between the animations of these gameplay segments, beginning animation production on one gameplay feature before its full finalization could save time.

4.1 Core gameplay

In pursuit of professional results and to ensure the animation principles and game animation fundamentals are used, the genre of 2D action platformer was chosen. Arguably the gameplay within this genre of games has enough variety in animations to offer extensive results. Furthermore, it also allows the author to place the animations within their portfolio to demonstrate their animation proficiency through the variety of combat between player avatar, NPCs, and platforming.

For gameplay reference, games such as Sekiro (2019), Hollow Knight (2017) and Crowsworn closed demo (2023) were analyzed. A combat system where the focus on observing animations was conceptualized. The player deflects hostile NPC attacks at the right time and attacks when the NPC will not be able to hit the player avatar in time until their control of the character is returned. This was inspired and is similar to Sekiro.

Deliberate emphasis on player engagement through observing animations affords the animator an opportunity to create exaggerated animations. In adhering to established animation principles and fundamentals, the animator is encouraged to infuse expressiveness. A gameplay loop wherein animations are played in full at times, rather than interrupted for immediate input responsiveness, may grant the animator a chance to further plan animations to serve an overall fluid visual feedback for the player.

4.2 Environment

The environment and the context of the environment matters for the animation as stated by Cooper (2019). The environment can affect the emotion and feel of PC’s actions. For example, if the player is surrounded by a scary or dangerous environment it may seem confusing if the PC has a joyful attitude in their animations. The context of the environment and the animation may not match the intended way creating confusion. Therefore, the environment of the game should be established and informed to the animator to achieve correct context in animations.

The term world is defined as the complete thematical look of the game’s environments not just the immediate changing environment of levels the player moves through. The world is a postapocalyptic radiated world with lush vegetation growing back over the remains of human civilization. The inhabitants of the world environment are insects capable of using tools and forming societies with culture. Resources are scarce and creatures are fighting to survive. This offers some context for emotional depiction in our animations.

Establishing a ruinous environment can also influence character design. The clothing, weapons, personality, and morals of the PC and the NPC can be constructed in further detail as the environment is established. In return, defining the character design further affects animation.

4.3 Story

In order to give the animator more context of the world and characters in animation production, establishing a story is beneficial. The story may promote distinction in animation creation instead of homogeny, possibly bringing more personality and exaggeration to the animations. This allows the principles and fundamentals of game animation to be more visible and easier to observe.

The story will be about a mantis warrior named Apterous who sets out to destroy a dam constructed up the river of their village, to bring an end to the drought. On their way, they face bandits and other dangers. An established story with a goal gives motivation to the character. Motivation and a goal, in turn, give additional context to the animator on why a character performs the actions allowing the animator to imbue more emotion into the animations.

5 Designing a charter

Character design is done before animation to establish the visual features and forms of the character. This section covers only the key aspects essential to animation to animation in character design. These aspects are narrowed down to shape language, symmetry, and silhouette as possible paramount aspects.

The favoring of design traits for animation may not always be a goal in game development. In a full-scale production, where multiple individuals may work on a single character, favoring features for animation is not always possible. Design choices are likely going to lean towards serving the overall project to cover story and gameplay not only animation.

5.1 Early concept

The author has more familiarity drawing the human anatomy compared to other members of the animal kingdom. This subjective preference was noted and kept in mind during the character design. To ensure the animator, in this case the author, would have the highest chance of success during the animation process the character design would resemble the human body, meaning be humanoid. In conclusion, the PC insect needed a humanoid body structure.

Cooper (2019) states that a good video reference can make all the difference between a good animation and a great one. Having a humanoid PC makes creating reference footage possible by having the animator perform the motions themselves in front of a camera. Finding references for the animation motions from the internet will also have more quantity. With this design choice, the workflow of the animator is considered and simplifies the production.

5.2 Design reference

Before beginning to draw the concept art of the PC, reference pictures should be collected and studied. The study of insect anatomy through pictures, such as in figure 1, can lead to solutions on how to modify their bodies to mimic the human body and its proportions. The main insects that are chosen to be referenced for the PC should have natural traits which complement the PC envisioned to be presented in our game. This prepared reference foundation for the animator supports the creation appeal for the PC.

The intended gameplay and its phasing will require animations with quick timing, force, precision, and intent. These motions and traits should be fitting for a lone warrior. This suggests that insects which are carnivorous and have fast movement capabilities should be favored, such as the mantis.

Figure 1. Examples of insect pictures used as reference.

5.3 Character design for animation

Humans as a species are naturally skilled at recognizing patterns and identifying outliers. (Sloan 2015.) Each shape from square, circle, to triangle, has a different meaning associated to it in our brain. Hence, many visual fields use them to build form. These primary shapes do not need to conform to their geometric ideals. Instead, they can be loosely interpreted by their definition and form. (Sloan 2015.) Mike Mattesi (2006) in his book *Force: Dynamic Life Drawing for Animators* explains that shapes form out of curved lines, straight lines, plains, and negative spaces. From the flow of the lines and the areas they close in shapes with clear favoring sides are formed. (Mattesi 2006, 103–104.) This *force* pushing the shape into its form allows viewers to understand the nature of the shape. The balance of these forces and their flow on the entire character will create their appeal. (Mattesi 2006, 1–43; 103–104). With this Matesi suggests that appealing look of a character and good character design does not come from primary shapes, but from the flow of the lines and the force they imply. When approaching shape language from an animation perspective it assists in maintaining form and clear movement. (Cooper 2019.)

Symmetry in character design has many purposes. In animation, a more symmetrical character is easier to manage due to the repeated shapes. (Howard Wimshurst 2018). In game animation, this helps when a character turns around and faces their other side towards the player. However, if a character is fully symmetrical using the same animation in mirrored form should suffice without losing clarity or feel.

By removing all internal details and discounting color a *silhouette* of a character is formed. This silhouette should be recognizable for any distance and most angles. An easily recognizable silhouette often leads to a visually memorable character. (Sloan 2015.) A recognizable silhouette will lead to clear movement and add to the appeal of a character. (Cooper 2019.)

5.4 The character

Construction of the PC began by drawing the large shapes, seen in figure 2, which define the PC as suggested by Cooper (2019). The flow of the force within the PC and the shapes was kept in mind. If the force was upheld during the design and placement of the shapes, the flow may be preserved for the animations.

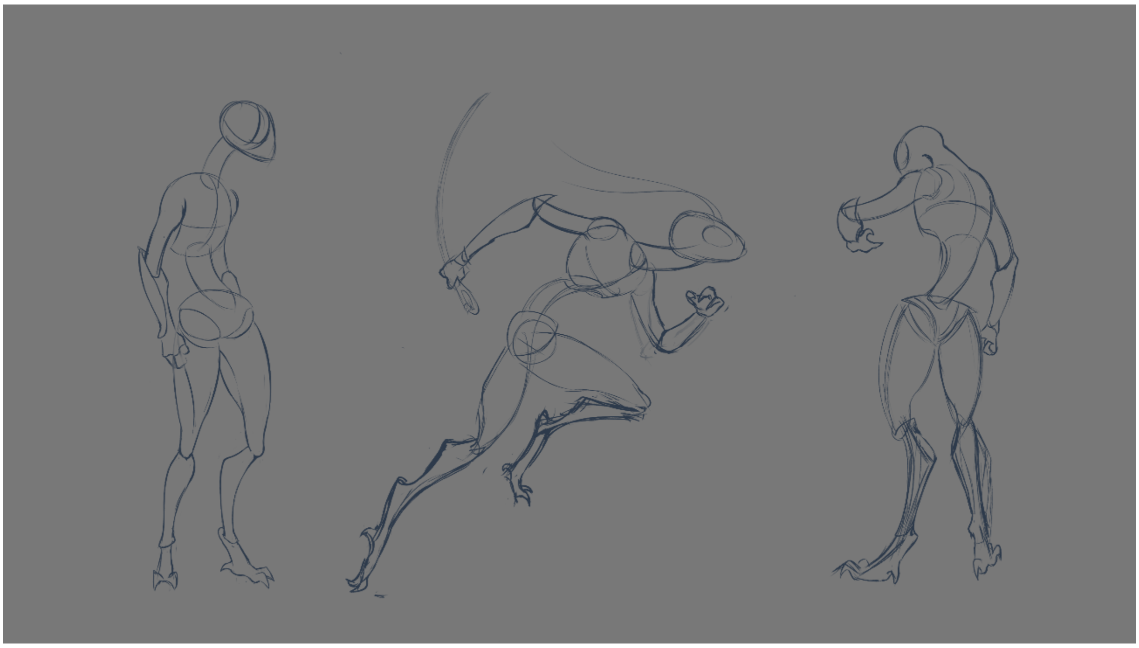

Figure 2. PC drawn in its basic shapes.

Small ligaments or sticking-out forms are difficult and time-consuming to animate. They also make movement harder to follow and not look as pleasing. (Cooper 2019.) While Cooper's (2019) recommendation might lead to good results in most character designs it was not followed with certain body parts, such as the forearms of the PC, which may have the most complex shape of the entire design. Several protruding extensions stick out from the forearms. It was a feature which was intended to complement the tibial spines, which some mantis species have attached to their arms. This design choice attempts to make the PC look more insect-like and offers an appealing detail to the PC design which can be later used in gameplay through animations.

Further rendering of the PC was drawn without clothes to narrow down how the anatomy of the creature works. A detailed version not suitable for animation depicts the anatomy of the PC (figure 3). Removing these details as the PC design progresses moves it closer to a practical animatable version where simple shapes define the anatomy. Once these shapes are formed clothes can be drawn over them to conceal them and cut the shapes hiding their full information beneath.

Figure 3. PC’s anatomy rendered to satisfactory detail to show how the anatomy of the creature works.

The clothing concealment method provides mystery for the PC. If the full look beneath the clothing, particularly on the face area, is not revealed it may create appeal for the PC. Yet, the ability to express should be maintained for the possibility of secondary actions through expressions (Sloan 2015). The concealment of the face can be used to advantage by bringing the bright eyes forward with a dark shadow concealing the rest of the face. Observing the head anatomy of a mantis it was noted that the eyes of a mantis are large compared to the rest of the head structure. Enlarged eyes are a commonly used character design feature in animation. Large eyes can make creating expressions easier, which are the most important part of an expression. (Johnston & Thomas 1981, 444–446). In the case of this production, it also follows the original anatomy of the insect.

Other design choices made specifically for the advantage of animation were the antennas. Their purpose is to be mainly used for overlapping and follow through. Since the PC will be moving in quick motions, having antennas which drag behind the motion can assist the player in perceiving what happened. This gives the animator a way to create clarity while still producing animations which fit a character with short input delay and enough feel. The risk of the antennas’ motion becoming messy and making the clarity of the motion harder is a possibility which must be kept in mind during the animation process.

The abdomen of a mantis was transformed and referred to as *pod.* This design choice would be the most insect defining feature of the PC and be the heaviest part of the design. Having the PC move while dragging this weight behind them was believed to add power and strength to the motions.

After designing the PC, it should be drawn in a character sheet format (figure 4). The purpose of a character sheet is to give an animator one easy reference image to which they can return to. By having the PC drawn in multiple different angles the animator does not have to interpret how the PC looks from a different angle, or what details are obscured.

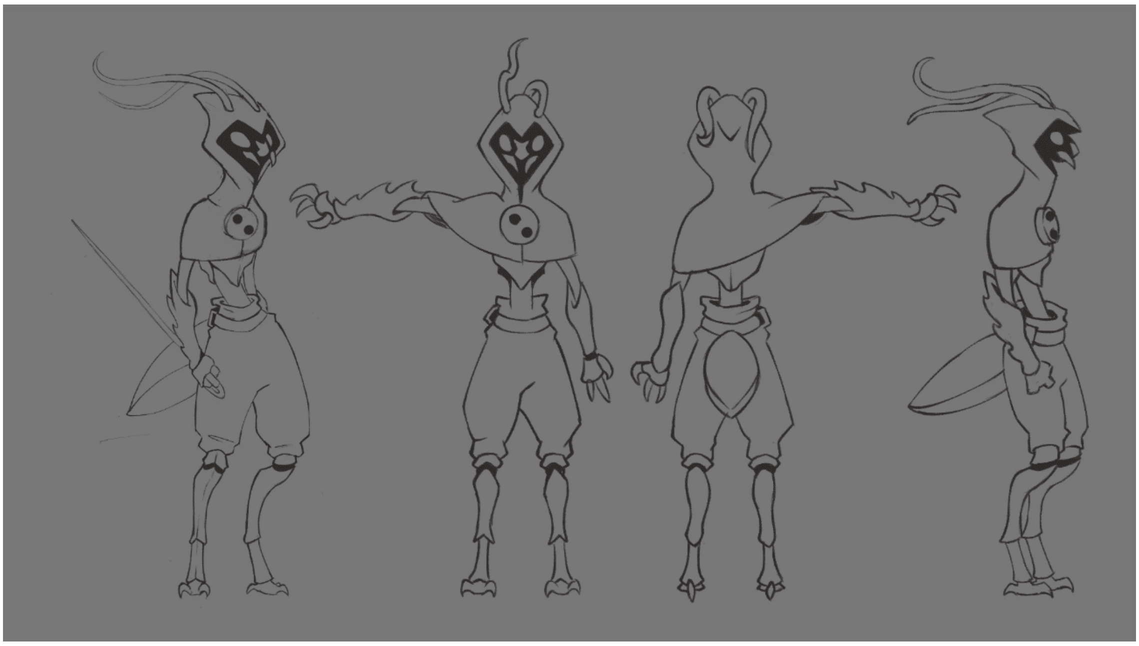

Figure 4. PC's anatomy rendered to satisfactory detail to show how the anatomy of the creature works.

6 Designing motion

Initial creation methods for game animations vary between individuals and companies. Production pipelines can be formed to meet the development needs of certain games with specific goals and art styles. (Cooper 2019). Despite this, there are methods which seem to appear consistently. Such as writing down the necessary animations which the designed gameplay would require in the form of an animation map. It can also show the reader how each animation connects to another and get an idea of how the animation will be coded within the game early in the project. An animation map is not solely a tool for an animator, but also a tool for game designers and other members of the development team. A game designer, for example, can use this as a method of communication with the animator on what is required for the project. It also assists in perceiving the scope of the project.

Figure 5 depicts an animation map for the PC used in this production. It shows the number of animations required to create a character for the envisioned gameplay with the proper amount of context and fluidity. The connection of animations between each other is also shown through arrow pathways. For the purposes of this thesis, actions which were deemed the most essential for the core gameplay; not taking actions, basic movement, and attacking, were created animations for. This means the necessary animations would be: Idle, Run, and A-1–A-5.

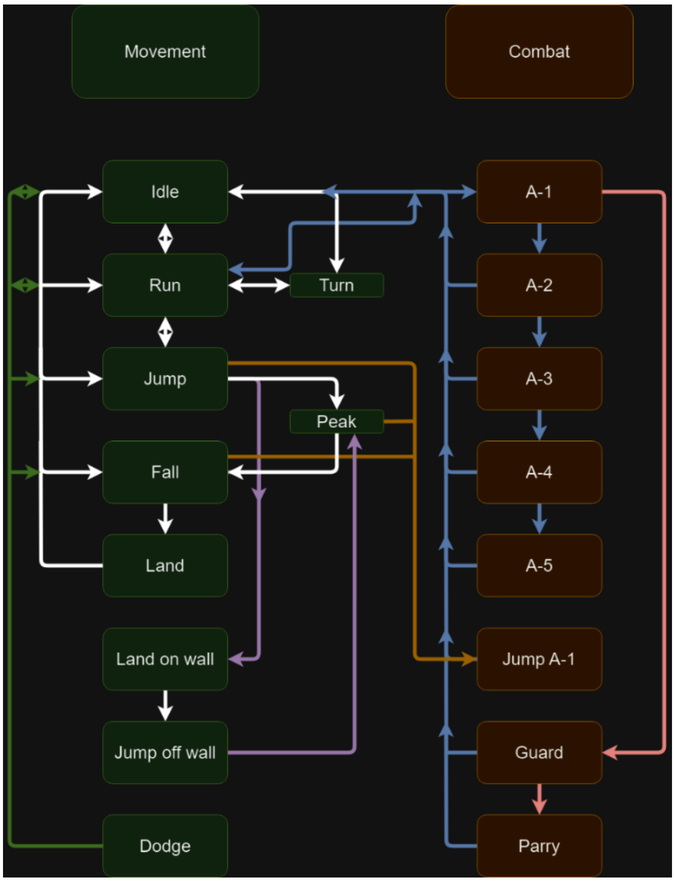

Figure 5. An animation map for the player avatar referred to as *Apterous* during the production.

6.1 Feel of the gameplay

Analyzing other games can assist in forming the feel of the gameplay for new games. Crowsworn and Hollow Knight have fast attack animations mainly consisting of keyframes and extreme poses. Input response times in these games are short and support fast paced combat interchanging between dodging and attacking. To give the animator more leniency to demonstrate the fundamentals and principles of animation, the gameplay should be similar, but with slower phasing. This can be created with intentional input delays or through animation. Animations can use recovery periods and anticipations to completely change the phasing of the game. Gameplay phasing, being mainly defined by animations, could encourage players to observe animations used by NPCs to wait for the proper openings to attack.

The early interpretation of feel can also be formed of rhythm. Music and acting out the motions in person were used to create early concepts of what timing the animations should have. *Mr. Blue Sky* (1977) *by Electric Light Orchestra,* particularly the base cords, were played in the background as different motions were performed to the rhythm. This led to determining that the main movement animation would be taking a step on each beat of the bassline, approximately every 10 milliseconds or 10 frames when played at 24fps, while the attack animations would mostly occur during every other bass cord, meaning attacks would happen at the base of 20 milliseconds in a combo. This should lead to gameplay, where engaging in combat halves the speed of the gameplay further encouraging the player to begin concentrating more on the timing of the animations.

6.2 Feel of the character

With the pre-established story and environment, the animator has context to imbue into the animations. These established elements assist in forming the first concepts of motion which will gratify the overall character the PC will become. Combining this with the knowledge of the animation map and the gameplay rhythm references the animator can begin performing the motions themselves to create references.

The references are observed while sketching out rough keyframes for animation, referred to as *a rough pass*. Creating a rough pass (figure 6) of multiple different actions early in the animation production gives the animator a chance to see which motions work best for the intended result. An overall visual feel begins to take form. This process is safer than animating single complete animations before knowing what the other motions of the PC are going to look like. These rough animations are useful for the overall development process since the rough animations can be placed into a game engine to run in a gameplay test with code implementation. (Cooper 2019).

Figure 6. PC’s anatomy rendered to satisfactory detail to show how the anatomy of the creature works.

7 Animating movement

7.1 Core motion pillar

The PC’s run animation was finalized first before any other motion was taken further from the rough pass stage. This is because horizontal movement across the screen is intended to be the most issued action by the player. Therefore, the run animation is the most frequently played animation in the game.

The frequency at which the run animation is played raised the need for proper appeal, timing, energy, and force within it to be as fitting as possible to serve the envisioned gameplay. Since the run animation is the most frequently played animation, it also becomes paramount for the gameplay. For these reasons, the run animation is referred to as a *core motion pillar*.

A core motion pillar means it is an animation which defines much of the timing of the gameplay and its animations. It is continuously referenced when timing and merging all other animations to it. Additionally, it can often be an animation which the PC returns to and flows out of continuously. Similar working methods have been used by Team Cherry in Hollow Knight where the developers animated and coded the jump and run of the PC first. (Milner 2018.)

A core motion pillar can be recognized and determined early in the production by observing an animation map. In figure 5 of the animation map, the run animation has several pathways converging into it which tells the animator the run animation is constantly played before and after actions. This tells the animator the run animation should serve fitting flow for the other animations and these animations in turn should have a fitting flow back into the run animation.

It is not recommended to use the idle animation of a character for a core motion pillar. The idle animation plays when the player is actively not controlling a character, making the fundamental of feel largely absent. Therefore, the idle animation should not be used as a core motion pillar despite the idle animation having similar flow pathways as the run animation on the animation map.

Working with the core motion pillar method assists the animator in maintaining fluidity between animations. Weight and speed at which the PC is moving need to be considered when the PC accelerates or decelerates to a new action commanded by the player. The momentum, which the run cycle gains within the game, must be maintained to create fluidity for the animations.

7.2 Running animation

Figure 7 depicts the first version of the 8-frame running animation. Colored lines show arc paths viewed from the camera perspective. The displayed arc paths belong to the closest leg, the furthest hand, and the tip of the sword.

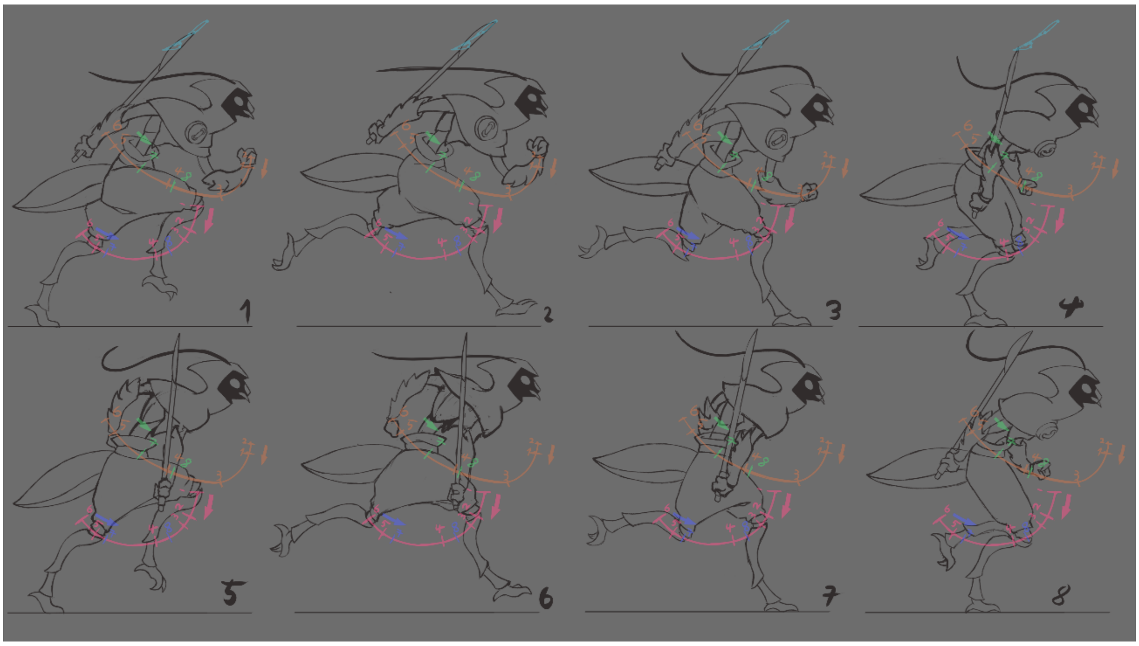

Figure 7. The first version of the running animation with frames separated and arcs drawn in.

The arcs in this version are *flat,* meaning it is a single arc line which the limb follows through the loop. The pink arc, showing the path of the knee, follows the line almost exactly to both peaks of the motion, meaning the ends of the arc. In turn, this causes the motion to feel mechanical, stiff, and lacking in dynamics.

Maintaining dynamic poses can be done by ensuring the arcs move on all axis; x, y, and z. (Cooper 2019). The purpose of the arcs is to depict reality where things do not move linearly. If the limb follows an arc too perfectly it may feel like a mechanical motion. Most of the time the path of action is either wavelike or the form of the shape of number 8. (Williams 2001, 90.) The shape of number 8 arc path from a side perspective the PC is being animated in mimics motion which moves on all axis x, y and z. Hence, the PC's arcs need to be adjusted to follow the shape of number 8.

There are six frames where one of the legs is contacting the ground. Moreover, the frames where both legs are in the air have the landing foot relatively close to the ground. This makes the running motion too stable and requires little force.

When people run in real life their legs stay in contact for a short time with the ground until kicking off from it. Additionally, the legs keeping contact with the ground increases the chance of *sliding.* Sliding in game animation means the PC is not moving or stopping at a proper speed related to the ground. (Cooper 2019). Causing an effect where the PC seems to be running on ice and sliding on the surface.

As the PC moves in a game engine, their position on the screen is updated in every frame by the defined fps. However, the animation itself changes to a new frame with every other frame. This is called animating on twos. (Johnston & Thomas 1981, 65). This combination causes the PC's legs to move to a new position relative to the ground without any mechanical motion happening in the animation. To avoid this, having as little ground contact as possible during the running animation may help.

With the final version of the running animation frames separated (figure 8), the arcs now have a resemblance towards shape of number 8 and are no longer flat. The spacing of the limbs between frames also favors the sides more clearly lingering in poses before taking a step. This creates anticipation where the PC is charging up an energetic step to push themselves off. These acts of holding poses simultaneously function as a follow through after a step. Both legs interchange between the anticipations and follow throughs, meaning when the other leg is anticipating a step the other one is following through a step.

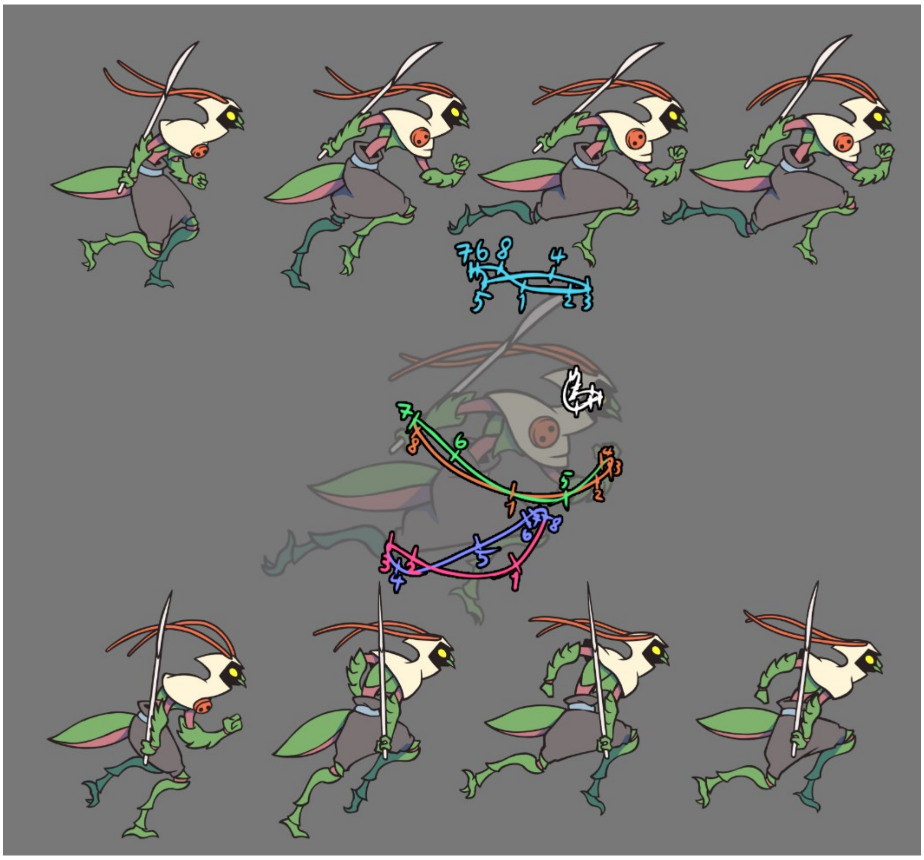

Figure 8. The final version of the run animation frames separated.

This pose lingering action can also be observed through the theory of ease-in and ease-out. The lingering is used to imitate reality and make the movement more appealing. (Johnston & Thomas 1981, 62; New Frame Plus 2020). It also adds readability to the animation by allowing the player time to register the poses. (Cooper 2019).

The white arc line follows the head (figure 9). The white arc line is notably smaller compared to the other body parts. This size difference stems from the lack of movement the head makes during the run animation. The head only moves up and down with ease-in and ease-outs remaining mostly stationary. The intention with this lack of movement is to make the PC seem focused and determined. The PC does not remove their gaze from the direction the player is moving. They stare forward towards the point of interest the player is directing the avatar towards. It can make the PC seem focused on objects or enemies the player is intending to interact with. This design choice of the movement adds feel and possibly correct context relating to the environment as recommended by Cooper (2019). However, this context partially depends on the player and if they are directing the PC towards a point of interest.

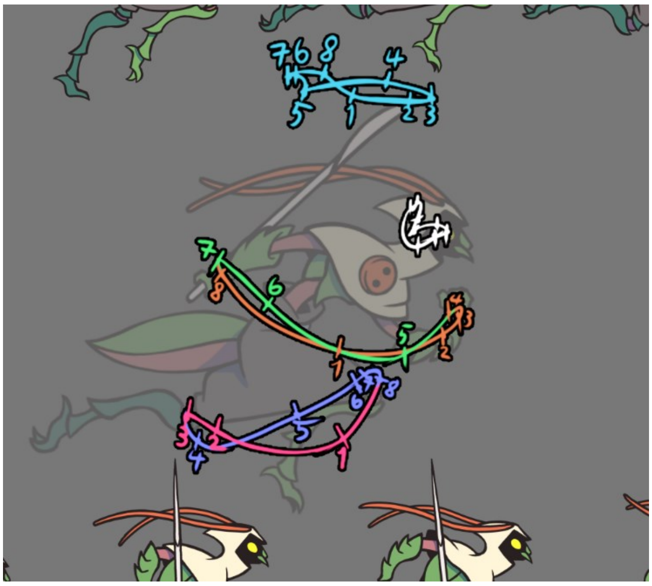

Figure 9. A close-up of the PC run animation arc lines.

There are three main twisting motions happening during the running animation. These twisting motions are present at the sword wielding hand, upper torso, and pod. Figure 10 shows their rotation directions at their extreme poses. Due to the PC design not having many details inside the outlines of the PC and being formed of flat colors, it could be difficult to discern which way the body is twisting. The button in the middle of the PC’s upper torso serves as an additional function of being a point of interest. The player can see the button disappearing behind the torso of the PC making it clear that the body has turned its back towards the camera.

Figure 9. A figure showing the direction and locations of some twisting motions occurring during the run animation.

The rotation of the upper torso and pod adds energy to the running motion. Each step pushes the whole body, which in return twists the pod being dragged behind the body. Twists bring perspective shift to the body parts adding more reality to the movement and feel. (Cooper 2019). Not all body parts require this twisting motion even if it would occur in reality. Adding it to a few chosen locations seems to serve the animation sufficiently. Arguably more challenging body parts such as the legs can then be left without any twisting motion.

Twisting the sword hand was required to keep the arm connected to the torso. Yet, it was not given a completely similar motion to the free hand. If it were to mimic a similar motion to the point of straightening out, the axis of the sword would rotate 90 degrees or more. This appeared to be too distracting for the running animation loop. As a byproduct, this makes the sword seem relatively stable during the energetic run, signaling that the PC always remains in control of their weapon and is prepared to strike. This should imbue some of the personality of PC into the running animation from the established story.

7.3 Idle animation

Animating the idle animation before attack animations could be beneficial in maintaining fluidity. The player can begin an attack from both run and idle states and return to either state from any attack. This is why the next animation to be produced was the idle animation. The animator would have difficulties animating an attack animation without knowing what idle animation looked like.

In the first version of the idle animation, the PC is flipped compared to the run animation as seen in figure 10. Ideally, this was meant to give the PC a prepared anticipation to kick off a running or an attack animation. It did function for this purpose but returning to this pose from running or any of the attacks with a single return animation proved challenging. The animator would have had to create two sets of link animations to return the character to either animation. One return animation for returning to idle animation and one for returning to run animation. This would not only have increased the workload but possibly created programming issues, which made the initial idle animation a nonviable option.

For the reasons explained, the idle animation had to be redesigned to improve the flow of the animations. Cooper's (2019) approach states that a game animator should design their animations to serve the gameplay and not bring unnecessary workload to other members of a development team. This re-design perspective considered the needs of a technical animator and coder within a development team.

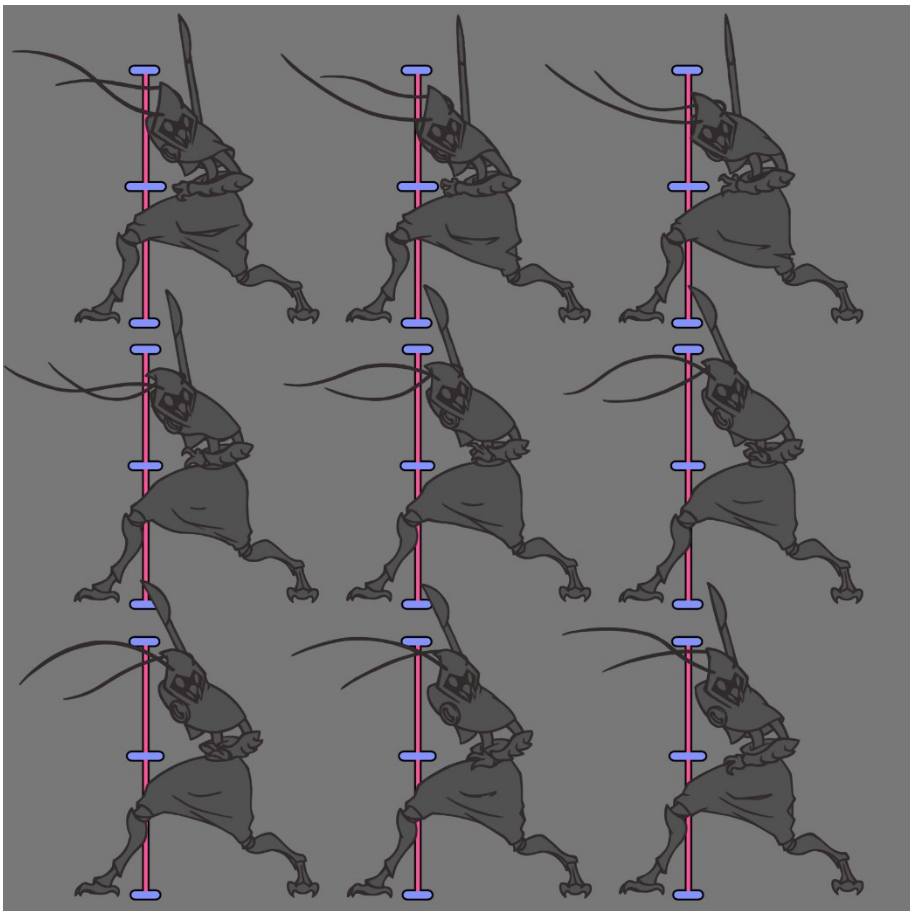

Figure 10. The first version of idle animation separated to its frames.

The idle animation (figure 10) pose was transformed to face towards the same direction as the running animation (figure 11). The legs were copied and flipped for this version from the previous one. The rest of the body parts required mostly new drawings. The upper torso and the abdomen connecting to the pants needed more lift towards the free hand. The flapping of the cloak also needed adjustments to fit the new angle, along with drawing more of the upper torso, which is revealed from underneath the garment.

Figure 11. The final version of the idle animation separated to its frames.

While reanimating the idle animation the core motion pillar was heavily followed to reference pre-established angles and forms. By comparing the first frames of both idle and run animation, the top left frames in both figures 8 and 11, some perspective and positional similarities can be noticed. The sword wielding hand is in a similar angle and position compared to the run animation. The head also has the same traits and remains facing towards the direction the player was last moving. This maintains the idea of the player avatar being vigilant and focused on their goals.

Similarities in perspective and position are needed to make the animation transitions smooth. Animated recoveries from attacks connect to both idle and run animation seamlessly with a singular return animation. With this, the fundamental of elegance is kept in mind. As explained by Cooper (2019) an animator should be efficient with whatever animations they produce. This does not only extend to the animation process itself, but game design as a whole. It can be demonstrated here by reducing the number of animations required.

Giving the PC's other hand a preparing action could increase the feel of the PC being a warrior and always on alert. Hence, the free hand of the PC was placed to cover the torso area and face towards the direction the PC was looking. The hand is intended to be prepared to deflect incoming attacks. The thick green chitin armor around the hand protects it from certain attacks and would allow the weapon to be either caught in the protruding spikes or pushed aside.

An animation depicting this deflection- or parry animation could later be produced, from which the player can flow into attacks. The animations necessary for the project did not require a parry animation, but designing the current animations with non-required ones in mind is a requisite to maintain the vision of the project. If they are not kept in mind, the practical aspect of the animations could be questionable.

Placing the free hand in its position to parry attacks (figure 12), has made it flow properly into the run animation as well. A small rotating motion was added to work as a *counter action* to the movement of the torso, which occurs naturally (Williams 2001, 230). A similar rotating movement was given to the sword wielding hand to have the sword in a readied state, yet not too stiff or completely idle.

Figure 12. The rotation path of the free hand counter action during idle animation.

Clothing flow works as an overlapping action. (Johnston & Thomas 1981, 60–62; Williams 2001, 226–230). The pants flow down the legs and extend outward between the legs with a slight delay after the legs have crunched to their maximum (figure 11). The flow action is intended to guide the players' eyes towards the ground and the feet along with the entire lower form of the PC. The PC feels more stable and planted to the ground because of this.

The idle animation presents the PC in their stillest state. During this time the player may have their attention directed to observing the PC model itself. This makes the idle animation an opportune place to give information about clothing as the attention of the player is in the PC itself and not in movement. The physics and volume of the clothing are established, such as the looseness of the pants and the weight of the cloak as it is lifted by the air. When the form of the PC and their clothes are squashed and stretched in other animations the player can recognize these actions are pushing the pants to new forms from the original creating exaggeration. (Johnston & Thomas 1981, 47.)

Overall motion in the idle animation seems energetic. The spring-like motion begins from the legs, which the rest of the body follows to the ends of the antennas (figure 13). This motion should mimic a motion of trying to start off running or attacking. This is done to maintain the flow of animations from idle- to run animation and from idle- to attack animations.

Figure 13. A partial flow path of the force within the looped idle animation

If a jump animation was to be implemented in the future, this idle animation should function as a spring-like anticipation for a jump. Additionally, a similar spring-like motion is present in the run animation ready to fulfill a similar purpose (figure 14). This further demonstrates how the animator keeps the animations outside the project's requirements within the design of the run- and idle animation. This creation thought process stems from the fundamental of elegance where the animator should strive to efficiently design and use their animations. (Cooper 2019).

Figure 14. Visual demonstration of the theoretical flow into jump animation.

8 Animating attacks

The envisioned gameplay comprises of avoiding attacks and dealing damage to enemies without receiving a counterattack from the enemy. Thus, the PC's attacks require a short enough timing to function as a reactive command in between enemy NPC's attack animations. Furthermore, the duration of the timing needs to remain long enough to give the NPC time to harm the player with their attack if the player input the attack command at an improper time. Thus, a dilemma between the timing of the animations fulfilling both criteria was created.

It would seem an attack animation can be divided into three different stages: anticipation, action, and recovery. During the anticipation-, action-, and possibly recovery stage the ability to move the PC is often removed from the player. The removal of control prevents the player from being too powerful and breaking the gameplay. Cooper (2019) states that this brief moment of the player not being an active participant in the game is necessary, but it should be maintained at a proper level to not reduce playability.

Attacks should be visually designed to require enough anticipation and force to fit the timing duration required for the intended gameplay. The visual feedback must also communicate to the player that the attack they commit demands weight and power, which cannot be canceled. The player should not be left confused when they cannot move during attacks and why performing an attack requires the necessary time allocated to it.

The personality of the PC and the context of the story should also be preserved in the attack animations. To cover these requirements, it was decided that the PC should be depicted to be in control of their body and not overshoot during the anticipation or recovery stage to an excessive amount. This should create visual feedback where, in the case of PC being hit by an NPC’s attack, the animations communicate to the player that they are the reason why they took damage. The player should feel responsibility for inputting an attack command at the wrong time and not blame the PC for being clumsy or too slow.

The attack animations must take several factors into account. The timing duration of the animation needs to serve the intended gameplay and not remove control from the player enough to create lack of feel. The visual feedback of the animation must also communicate a fitting representation of force and power to warrant the removal of control, the timing duration and create appeal for PC.

8.1 Attack timing and control

Finding a perfect timing for the envisioned game during the animation stage may be difficult. Playing the animations in a video player is different from playing them in a game engine with input mechanics activating them. This is one of the reasons why the rough versions are first tested in an early working version of the game. (Cooper 2019).

Due to the lack of coding experience of the author, testing the rough animations was not seen as a time efficient option. Instead, following the established rhythm of Mr. Blue Sky and referencing the core motion pillar was seen as a more feasible option. Furthermore, the animation timing could be adjusted afterward in Unity (2005) by moving the frames on the timeline, meaning perfecting the timing in the animation stage was not necessary.

During the recovery stage, the animation could remain uninterruptable to give more time for the hostile NPC to react. The uninterrupted animation could come to the detriment of feel, however. Removing the control from the player for the duration of the whole animation could make the combat feel stiff. Some uninterruptable recovery animations after a big attack, which the player can alternatively commit instead of a normal attack, seem to add appeal. Allowing the player to move or transfer to other actions during the recovery brings more control back to the player without losing much elegance and fluidity in the animations. Hence, it was decided most recovery animations would remain interruptible with a selected few remaining uninterruptable to bring more appeal to the PC while still serving the gameplay.

To enhance control of the PC a short pause should be hidden between the action stage and the recovery stage. This should be done to give the player time to see the effect of the attack they just performed. During this pause, the player can either activate another attack or give a different command. Activating another attack or a different command can cancel the recovery animation and begin a new animation. This pause can appear in the animation in the form of ease-in and ease-out, over shooting, or through an effect animation.

8.2 Filmed attack reference

If the animator films the performance of the action, they will be able to control the camera angles and capture how the motion within the action works. The animator should be wary of overacting their movements during the reference filming. Acting out the motion exactly as the animator sees the final version in their head, may negate the purpose of using video reference in the first place. (Cooper 2019.)

Figure 15 shows screenshots of the animator acting out the motions of the attack combo. A camera was placed to capture a wide side angle of the animator who would then proceed to perform sword swing movements with different timings. When the animator performed the action themself, they could discover how to balance their body during the swings and translate it to the animation.

Figure 15. Screen shot of video reference showing some of the key poses used in the final animations.

8.3 Delayed strikes

Some of the attacks are led by the body. The hands holding the sword remain close to the same position and follow the body after it moves forward as shown in figure 16. This technique is used to animate fight scenes and bring more weight to strikes as shown by the analysis of fight scenes by Howard Wimshurst (2018). It appears to be used in video games like Crowsworn.

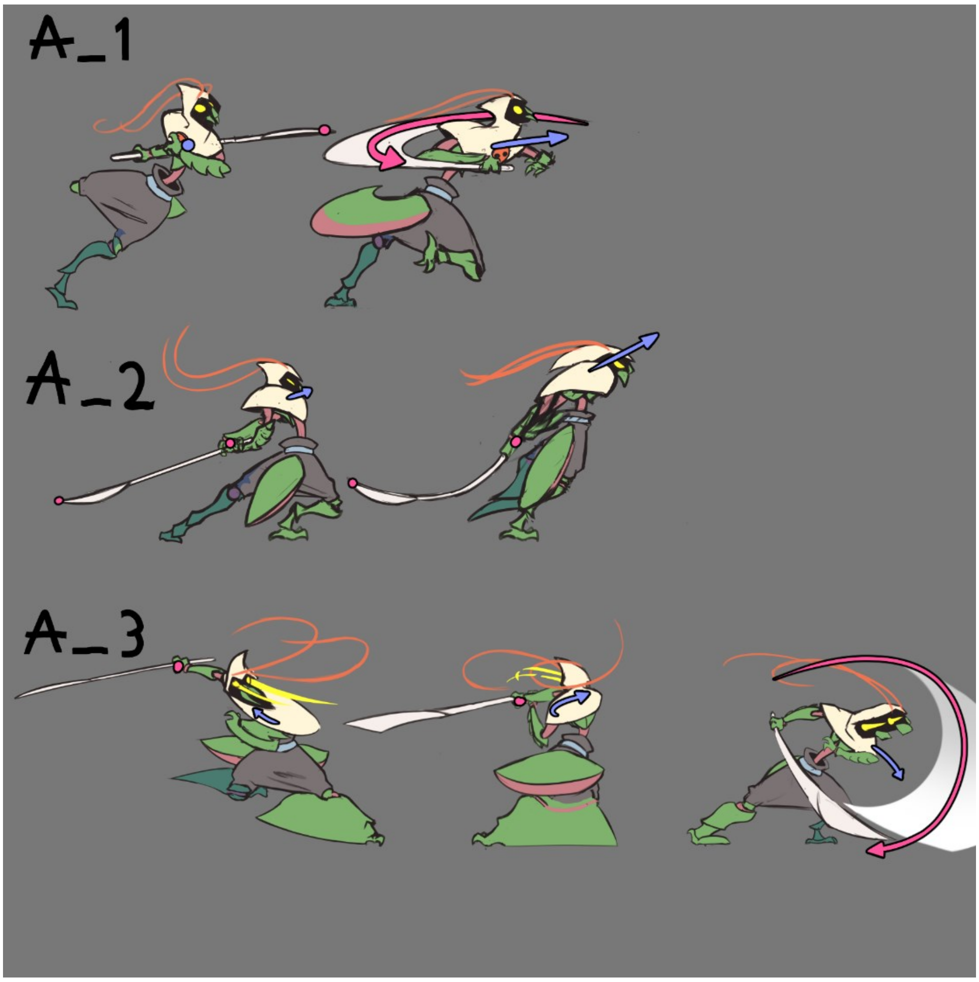

Figure 16. Examples of the hand remaining relatively stationary as the rest of the body twists and moves.

The visual idea of the technique is to sling the strike forward after a short delay. The bigger mass of the body moves first to then decelerate or stop which guides the energy to the hands to catch up. The hands then overshoot forward past the body in a quick motion creating an appealing strike motion.

8.4 Overlap

Because action in the attack animations happens during one to two frames, it may help to have an overlapping action with slower timing to assist the player understand what happened. Antennas have their potential unveiled in the attack animations. During the idle and run animation the antennas served a complimentary purpose in design and motion. They serve a more critical purpose in attack animations, one of the major reasons for their inclusion in the character design phase.

The antennas are much looser than their real-life counterpart insect antennas. This was done intentionally to mimic hair or a scarf adding another more human element to the PC in hopes of creating appeal. They are animated with a similar wave or whip techniques which also are used when animating tails (Williams 2001, 328). The root of the antennas is connected to the head, which moves first along with the head. The tip of the antennas follows along straightening and curing up as they catch up with the motion of the head. In figure 17, you can see the force traveling through the antenna like a wave towards the tip while the antenna moves towards its resting position. In attack_3, shown in figure 17, the motion of the wave was quicker compared to attack_1 and attack_2, adding force to the midpoint of the attack combo. This separation from the first two attacks assists the player in telling when their combo is at this stage. Intendedly they can know a different attack, referring to attack_4, will be next if the attack command is given again.

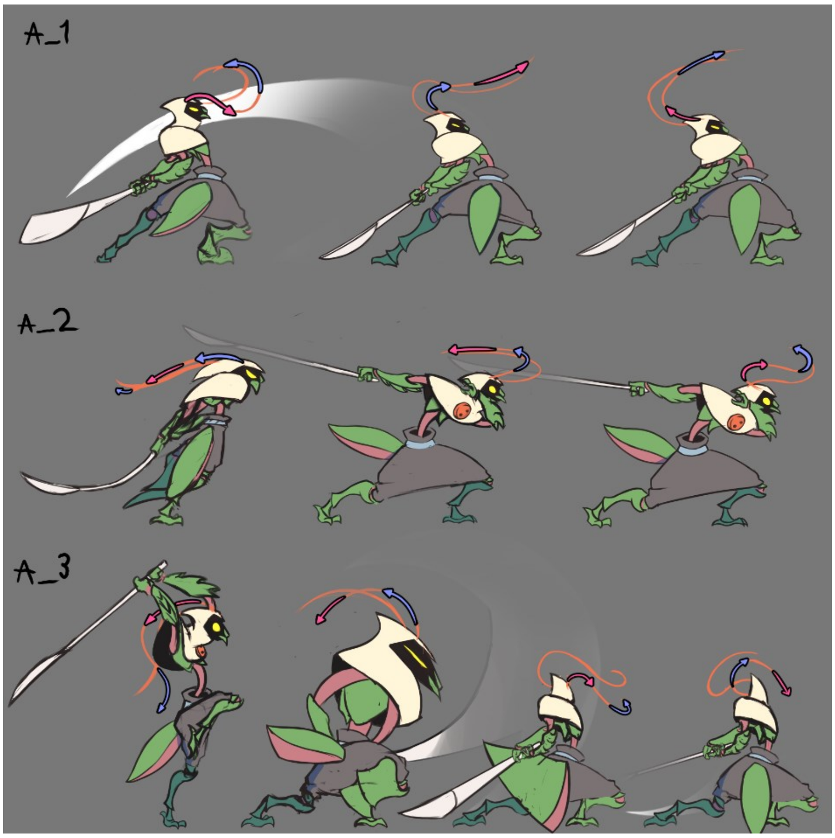

Figure 17. Flow of force moving the antennas after the rest of the body has stopped moving.

8.5 Expression

To avoid the PC from feeling robotic, some expression changes were added as a secondary action as suggested by Sloan (2015) and Johnston & Thomas (1981, 63–64.). The PC holds the same expression for the idle- and run animation to establish a default emotion the PC is feeling during basic actions. Expression changes were avoided during these animations to avoid making the loop of the animation obvious. (Cooper 2019.) The slightest change to the face of the PC such as the eyes should now stand out more clearly even during actions with few frames.

The expression changes as secondary actions also add appeal to attacks. The PC squints their eyes, as shown in figure 18, to focus on the target ahead during anticipations and then opens them again to switch between the default expression and squinting. If the PC was to squint their eyes during the entire attack animation, the expression change might not be noticeable enough for the player, thus losing most of its function.

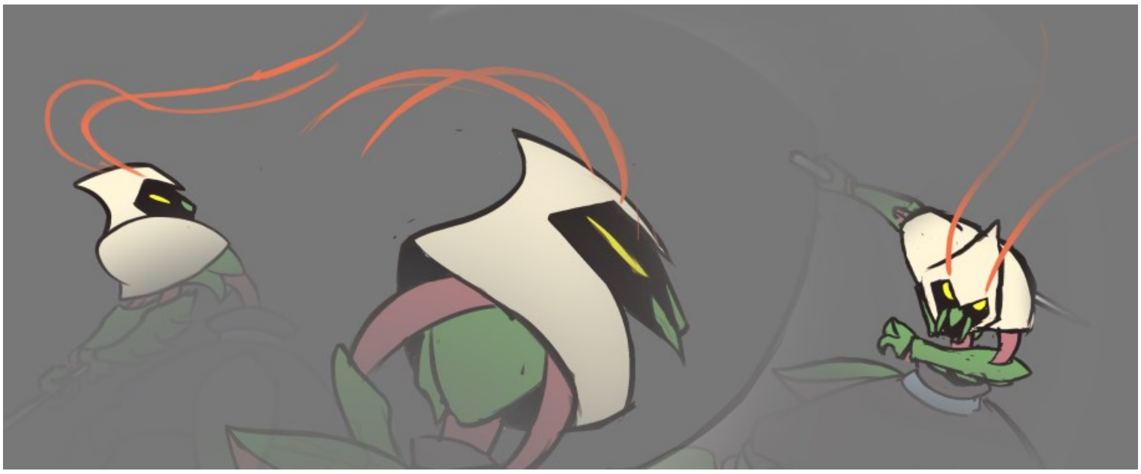

Figure 18. Examples of expression changes during attacks

During attack_3 recovery stage a flourishing sword movement is performed as a secondary action. The sword spins as it maintains a circle shape of a specific size and moves upwards towards the intended resting position. Meanwhile, the main action of the PC, which is to turn their body, is held for a short period. During this holding the torso and hand of the PC extend to show the viewer the mantis preparing their hand to catch the sword at the right moment.

8.6 Smear frames

The attack animations rely on the use of *elongated inbetweens* (Williams 2001, 96) or more commonly known today as *smear frames* as explained by Toniko Pantoja in his video essay on the Toniko Pantoja YouTube channel (Toniko Pantoja 2018). Smear frames are drawings that use different stylistic approaches such as speed lines or stretching out the shape of the PC to emulate motion blur from live action film. (Willimas 2001, 96–98). In figure 19, some of the smear frames created for the attack animations can be seen.

Figure 19. A collection of smear frames used in the attack animations.

There are many different methods of creating smear frames (Williams 2001, 92–98, Toniko Patonja 2018.) In this production, the used smear frame style has the PC’s form widened without redistribution of the mass. This means the shapes the PC is formed of do not receive squash and stretch intentionally and their volume changes, as shown in figure 19. If smear frames receive too much squash and stretch, it might make the PC appear too squishy. Particularly, doing attacks when the PC is meant to have impact if it connects with a target, having a squishy form could make the landed hit seem weak.

8.7 Breaking and returning to form

The principle of exaggeration (Johnston & Thomas 1981, 65) appears in the run and attack animation as deformation of silhouette, size changes in the PC model, and speed of movement. Temporary changes, such as the size change of the sword and smear frames, all break the established form of the PC. This form was established in the calmer animations of run and idle animations, which mostly persists in the attack animations as well. These changes are expected to not be noticed by the player consciously, but subconsciously, allowing more weight and speed to be added to the motion to create appeal.

Compared to the run and idle animations the sword is enlarged for the attack animations. The enlarged size of the sword would stick out of the silhouette too much, which is to be avoided to a certain extent according to Cooper (2019) when creating appealing silhouettes. Turning this concept to advantage in attack animations. The intent is to make the sword stand out clearly during the animations since the action of attacking uses the sword to accomplish these actions. It is arguably the focus of the action.

Returning to form should eventually be completed to allow the character to have an established silhouette and shape to break and exaggerate. This form can be maintained by referring to previously drawn frames of the PC. In this case, idle and run animation frames were placed on the background while drawing the attack animation combo. The animator could then move body parts and shape them to the size, angle, and shape of the previous frames. This method should assist the form of the PC and the exaggerated to remain consistent, which was noted as an importance by Cooper (2019).

8.8 Effect animations

Character animation alone may not always be sufficient to create enough clarity in a game. Several elements may appear on screen at the same time and cause the player to lose clear sight of their character. It may also become unclear if the input was received correctly by the character if one cannot see which animation the player avatar is playing. Character animation can be assisted by producing *effects animations*, which bring more clarity to the animation. (New Frame Plus 2019c.)

Effects animation does not only mean the art of animating elements like fire and water, but also the art of exaggerating elements in a stylized manner. (New Frame Plus 2019c.) Effect animations can appear in games in many different forms. Such as the visual elements of the game stopping still and the screen shaking when a character is hit. The pause gives the player a moment to recognize which attack hit the enemy. Effect animations can also be swirling lines which gather to a point of interest to indicate charging up something for an action.

In figure 20, some examples of effects animation can be seen. In this production, effect animations appear as white streaks to tell the player which path the sword traveled to get to its new position. The motion of the sword cutting across in front of PC does not exist in several of the attack animations. Instead, the PC moves from an anticipation frame to overshooting and recovery. This makes the action snappy fast and contains a lot of force. The white streak functions like an afterimage. It shows how the sword got to its new location and removes confusion from the player on why the sword suddenly appears in its new spot.

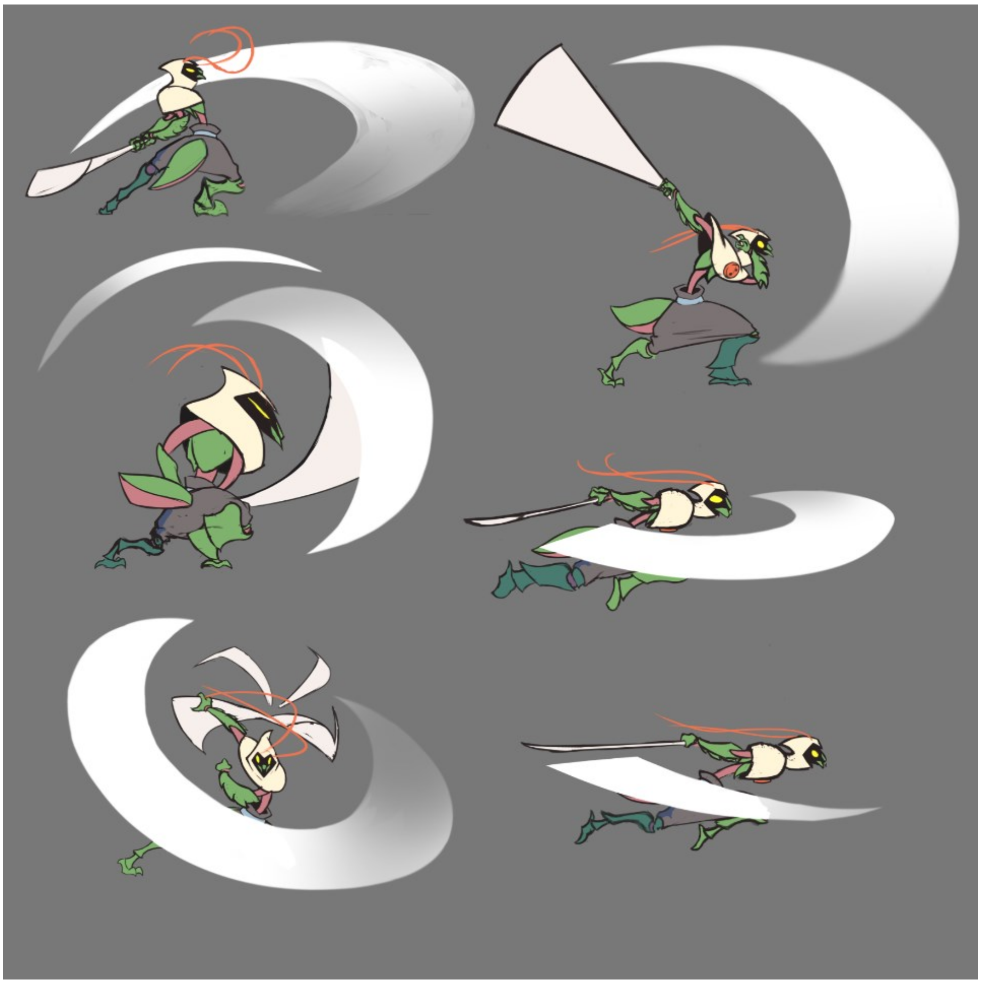

Figure 20. Effects animation appear as white streaks in the attack animations.

The white streaks attempt to inform the speed and size of the strike. Additionally, it tells the player where the hit would have landed, A hitbox, an area defined in a game engine for recognizing if an attack hit a target, could be placed in the shapes of these white streaks for their duration. This way the player can tell where the attack areas are and measure the appropriate distance from a hostile NPC. The effect animations through this hitbox could serve as a gameplay function in addition to bringing clarity into animations. With this approach, the fundamental of elegance is maintained by using animation features for multiple purposes.

8.9 Movement during attacks

The PC will move forward with each attack in the game engine. The intention is to give the player different forms of maneuvering during combat. The player does not have to stop attacking and moves the PC closer to hit an enemy with their next attack. Instead, the player can input another attack, which will take them forward, whilst possibly landing the next hit of the attack combo. This method would give the player the power to complete two actions with a single input, which is conveyed by a single animation.

The visual feedback of the attack animations needs to match the ability to move during them. If the feet of the PC remain still during the animation, the mantis will be sliding forward, as if running on ice. This is why the feet of the PC move during an attack, which is not always the case in games. A specific distance, viewed in figure 21, was measured and maintained for the ending positions of the legs in most attack animations. By having the distance of the leg movements remain consistent in the attack animations the distance the PC is moved in Unity can be copied.

Figure 21. Effects animation appear as white streaks in the attack animations.

The distance the legs move closely matches the idle animation's leg positions and the running animation's contact leg. The running animation contact leg was slightly further back to transfer the momentum of the legs to the runt animation and avoid its sudden stop. Having the leg spacing of the attack animations match other animations assists in improving flow back into idle- or run animation through a return animation.

8.10 Leaping attack

Attack_4 and attack_5 distinguish themselves from the previous attacks by performing a long leap slash (attack_4) and a final powerful strike (Attack_5). The leaping maneuver moves the PC in the game much further than the previous attacks. This is intended to give the player the option to reposition themselves during combat and possibly dodge an enemy attack. If there are multiple enemies surrounding the player, they may first attempt to attack with the first three attacks of the combo followed by possible realization that the attacks are not killing the enemies. The player can then proceed to use the fourth attack to leap through the enemies in hopes of removing themselves from the encirclement. It should be noted that despite the attacks making the PC forward movement possible it is not large enough to replace the running or jumping action during combat. Hence, giving the player the option to perform a long leap after the three attacks returns some of the control and momentum the player lost by inputting attacks.

The fifth attack is meant to optionally interrupt the leap animation and the momentum in it. The option to interrupt the animation assists the player to not overshoot from their desired location. The fifth attack does not instantly stop the momentum but continues forward with a slight ease-out from the momentum of the leap depicted by the legs still moving in the animation. The animation principle is not only applied in the animation, but the game engine itself where the visual feedback of the animation fits the physics of the game.

The fifth attack is the most powerful of the attacks and ends the chain. This is due to the twisting charge up of the anticipation, which the mantis enacts before coming to a relatively quick stop and swinging their sword. It has the longest pause between the action and recovery in addition to the longest recovery time. The inability to move or change to a new action signals the player that the PC needs to pause and recover from the exertion of the swing. It creates a long window of time when enemies can land hits without the player being able to respond. This feature encourages this action only to be used when the player is certain enemy NPCs do not have an opportunity to attack the PC during the recovery.

9 Implementing animations

Unity was chosen for the in game practical test of the animations due to its accessible and readily available animation *state machine*. An animation state machine, shown in figure 22, visually shows which animation is playing and how each animation connects to one another. Parameters, seen at the top left of figure 22, can be placed into these connections, which can be further defined and tuned in code to play at the correct times. (Unity 2023.)

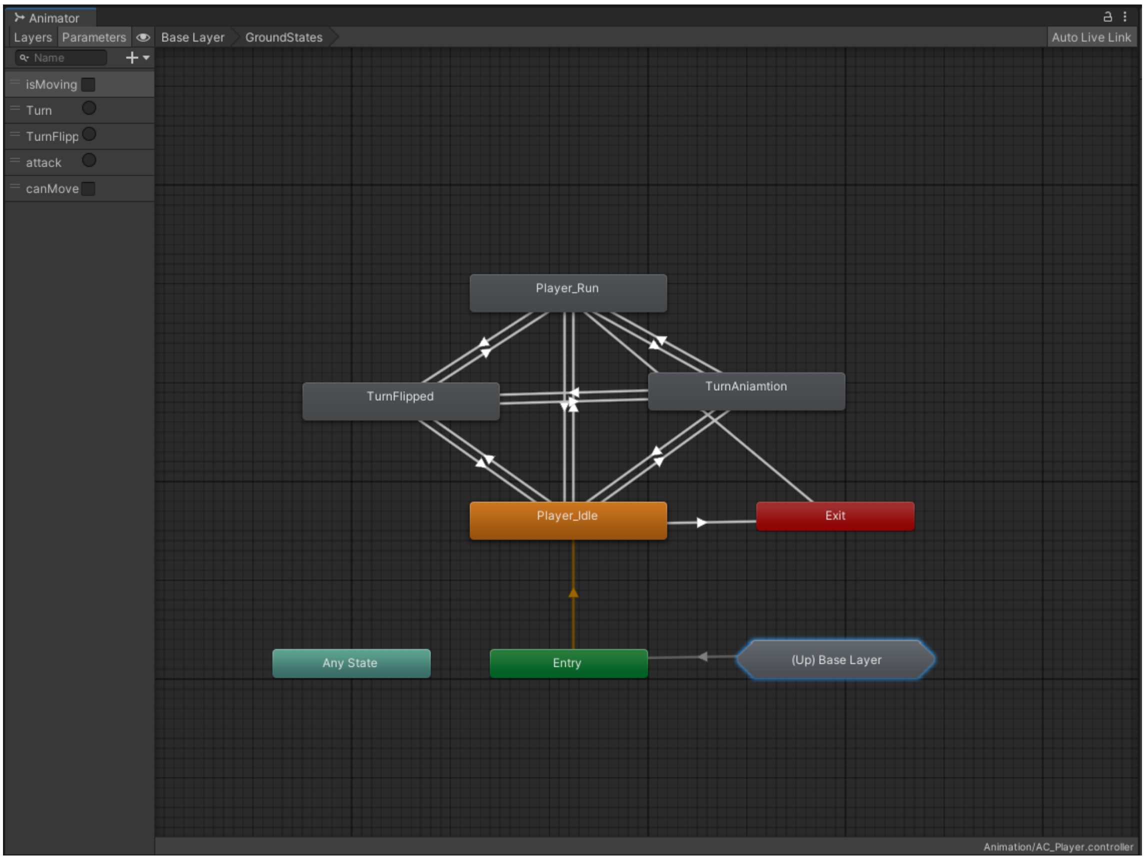

Figure 22. The animation state machine in Unity showing PC's movement animations and their connections. Screenshot from Unity. (Unity 2005).

AI (artificial intelligence) programs such as ChatGPT (OpenAI 2023) can assist in simple coding in Unity. It should be noted that a professional technical animator should not need to rely on AI-assisted programs to create character movement codes. However, using AI-produced code does fulfill viewing the animations through the perspective of a technical animator, game designer, and player despite the code being created by an AI created fully through the knowledge of the author.

9.1 Importing

To import 2D animations into Unity they first must be exported out of Clip Studio Paint (2001). Exporting was done via *image sequence,* shown in figure 23. Image sequence means each frame of the animation is exported out as a designated image file format. Each exported image was set to the scale of 1500x1500 and formatted to PNG files. PNG format makes transparent backgrounds possible with the PC being the only visible element in the image.

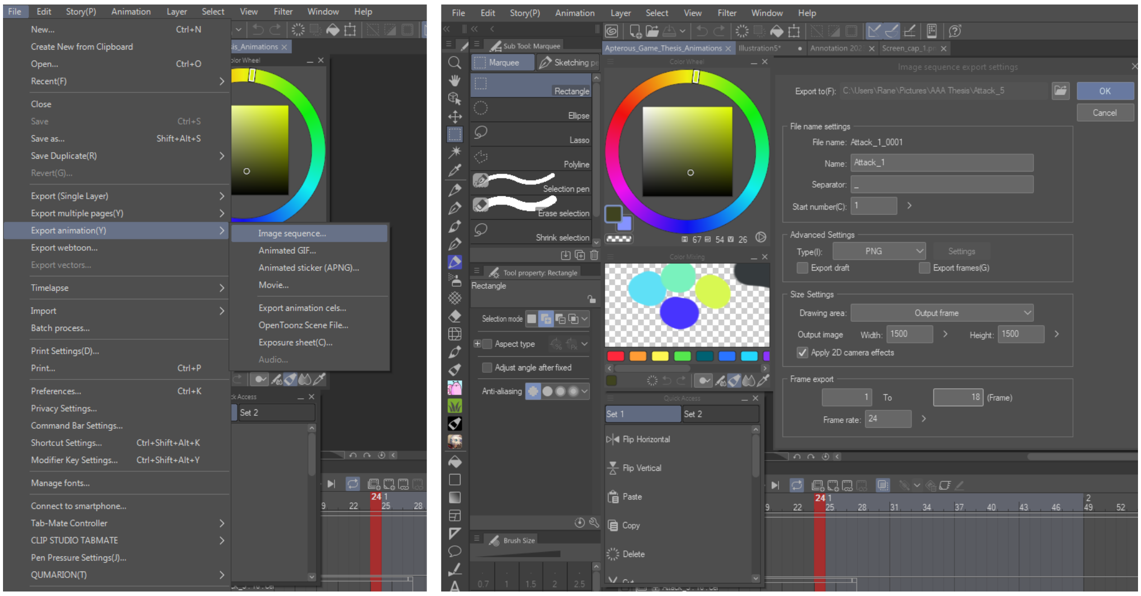

Figure 23. Demonstration figures of animation export process from Clip Studio paint. Screenshot from Clip Studio Paint. (Clip Studio Paint 2001).

Character animations are often split into *sprite sheets* to properly compress the animation sizes and make the handling of animations easier for further game production. Sprite sheets are single large images which contain most or all frames of animation the character uses. However, the sprite sheet method was not used in exporting and importing the animations into Unity. The testing of the animations did not demand the size of the files to be optimized. To save time from building a complex sprite sheet, where all the animations would fit symmetrically, each animation frame was exported into Unity separately for the test. The sprite sheet export method should be reserved for after the animations have been first tested and finalized for the game.

After the images were imported into Unity, they were placed into an animation player, shown in figure 24. The animation player was then placed into the animation state machine, which is connected to the PC. The code, seen at the bottom right of figure 22, referred to as AC_*Player controller* defines the animation parameters connected to transitions in the state machine. The parameters activate when the PC is moving, holding still, and turning to face the opposite direction.

Figure 24. Figure of the animation player in Unity. Screenshot from Unity. (Unity 2005).

9.2 Snapping visual error

A visual error appeared when the PC would turn and flip their sprite on the x axis. The visual error made the run animation momentarily unclear after a turn action and distractingly snappy. The error seems to have been caused due to the PC returning to the idle animation for one to two frames while turning.

The reason why the PC returned to idle animation for a fraction of a second was because the input response time was too immediate. When the player changed directions, they had to let go of the other directional button first. This sometimes caused the movement parameter to be unactive for a brief period of time. The code interpreted the PC to be still, causing the idle animation to trigger.

Arguably this error could be due to the coding of input response and not animation. The error could have been remedied by implementing a code which creates a slowly decreasing number after the directional key has been released. A new issue of how low responsiveness could have emerged from this instead because then the PC would not stop where the player intended. Therefore, the snapping error was remedied through animation instead.

9.3 Turn animation correction

The turn animation has two frames, which are on twos, creating an animation duration of four frames. The timing of the turn animation needed to be as short as possible to maintain an appropriate feel and not delay the directional movement of the PC in the game. This means the PC had to move in the direction the player commanded even while turning, which means the animation needs to be created to serve this requirement.

If the mantis has their leg attached to the ground whilst they turn, it may cause the sliding effect to occur. To avoid this, the mantis was animated to perform a small hop where both legs lose contact with the ground. Any frame of the run animation loop should flow into the hop of the turn animation. The reasoning for this is because the PC already is performing a similar hop during their running cycle. It can either flow into a seemingly extended hop, as seen in figure 25’s lower example, or from the legs pushing the PC into an immediate turn hop, as demonstrated in figure 25’s upper example.

The fundamental of elegance is demonstrated by designing the turn animation to work as a link animation. The turn animation is designed to fit any frame of the run animation and the idle animation sufficiently without any changes to the previous animations. This provided an opportunity to exchange the shading between the legs by having both legs be in shadow for the duration of the animation frame.

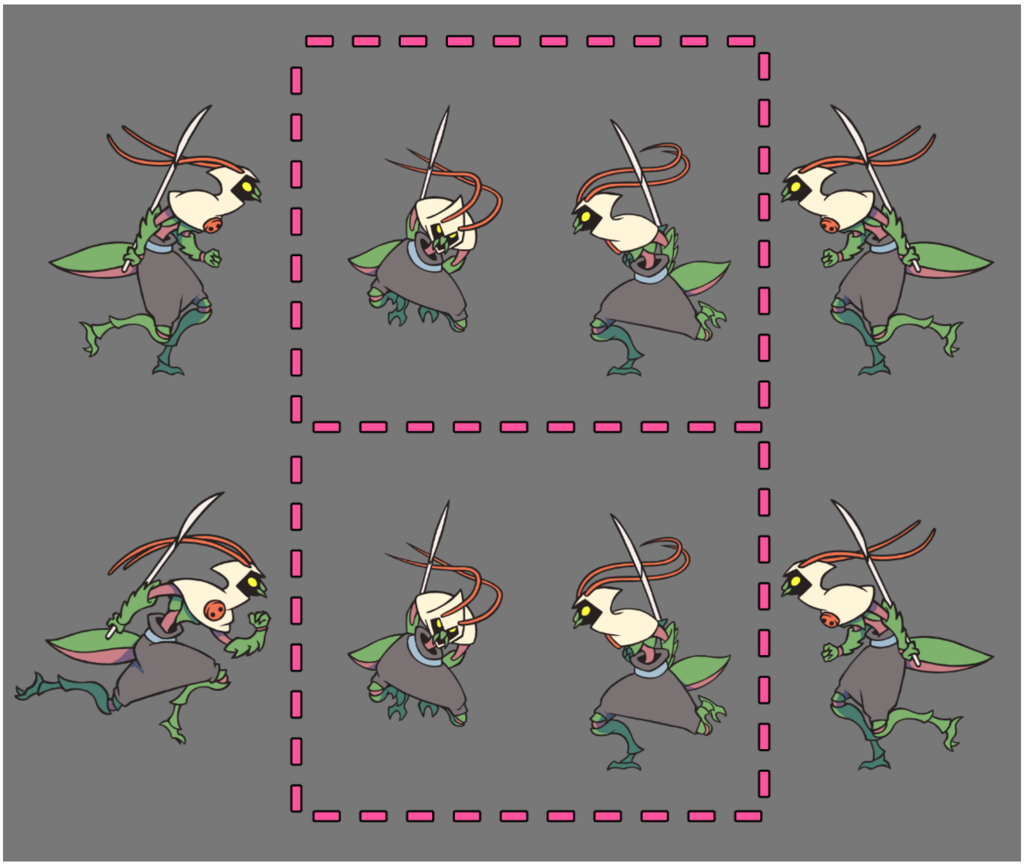

Figure 25. Demonstration of the flow from running animation to turning animation.

The PC simultaneously puts their hands behind their back to swap the sword wielding hand with a secondary action. Extra work was avoided by switching the sword behind the PC’s back. It also seems appealing to hide this quick secondary action to let the turning of the body and head be the main action. The secondary action of the sword hand switch is used to complement the rest of the animation and avoid drawing too much attention to it.

9.4 Timing adjustment within Unity

While changing the timing in Clip Studio Paint, shown in figure 26, it was noted that the feel of the animation was unfitting for the phasing of the intended gameplay. By adjusting the timing to be faster, particularly in the action stage of the attack, the animations received more force. Adjusting the timer to be faster also gave the animations faster input response time.

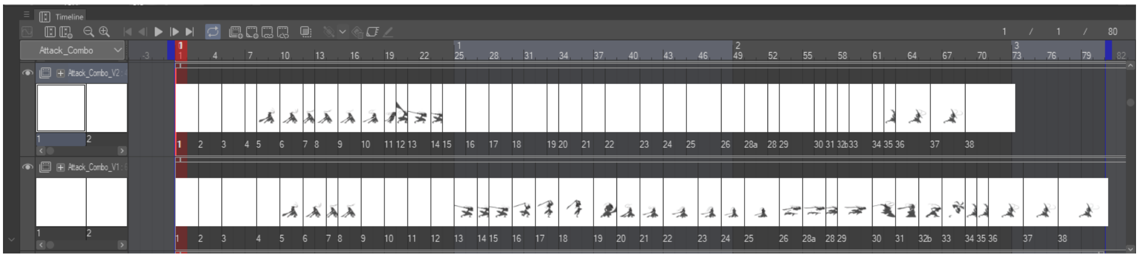

Figure 26. Timing fixes tested in clip studio paint, which could then be replicated in Unity animation player. Screenshot from Clip Studio Paint. (Clip Studio Paint 2001).

As predicted during the animation production stage the timing could be adjusted in Unity animation player to replicate the timing created in Clip Studio Paint. All drawn frames were placed into the Unity animation player and some of their timing was changed. With this adjustment, none of the previous animation was wasted but transformed the attack animations to be much smoother at certain parts.

10 Reflection

10.1 Fluidity and feel in practice

The code, which runs the PC and the animation, was not without errors and could be broken seemingly with relative ease. Despite of this, it functioned to an extent where all animations played when a command input was given. Thus, testing the feel of PC animations and gameplay was possible. This was sufficient for the project, since the goal was to test the animations, not to create a finalized playable PC.

Most animations seemed to flow into each other appealingly. The intended animation flow paths were working. Based on these findings the preparations made in the animation map should make the implementation for the rest of the animations possible.

The recovery stage of the attack animations can be interrupted by inputting a follow-up attack. The momentum seems to ease-in and ease-out appealingly between running animation and attack animations. The anticipation takes the momentum of the running animation into account appropriately and the recovery animation builds up the momentum for taking off into the running motion.

The movement speed along with the movement during attacks shows that the PC can move in the game without the sliding effect. Further adjustment and testing are required to fully remove any sliding that may occur. The testing did have some sliding occur at times during the attacks, but this could be corrected with further adjustments through the physics of the game engine.

Overall, controlling the PC feels precise and fast. The phasing of the animation actions changes when attack commands are given as planned during the gameplay capitalization. It should be expected that following the core motion pillar and established rhythm will further assist in maintaining the rhythm and feel for the full gameplay.

10.2 Clarity in practice

The turn animation improved the issue with the snapping error albeit it did not fully remove the issue. The snapping error still sometimes occurs when the PC moves from idle animation into the turn animation. Nevertheless, it seems to have improved the overall flow and readability between idle- and run animation.

The attack animations seem distinguishable from each other and the appealing features the animator intended to communicate are clear. Force, speed, and weight behind the actions is not lost from moving the animations into a controllable PC in Unity. Coding a functioning camera into the game, which follows the player, assisted in clarity during attack animations. The camera maintains the PC at a certain position on the screen preventing them from going off frame with the leaping attacks.

The white streak effect animations successfully communicate to the player where their hit would have landed. During the testing of the animations hitting a specific target in the background was tested. Because of the white streak effect animations, the area of the strike could be predicted, and the target was hit.

10.3 Elegance saves time

Following the fundamentals of elegance and designing animations along with game mechanics saved notable amounts of work hours. Designing animations to fulfill more than one action or purpose used the production time efficiently. The elegance of the animations saved time on the gameplay development side. In addition, it brought more personality and character into the animations.

Designing animation for multiple purposes in mind seems to make the work of technical animator and game designer easier, yet it also has a risk of making the coding with animation state machine challenging. Testing is required to avoid creating difficulties by trying to have multiple uses and actions for single animations. Creating roughs and proceeding to test them in the game engine should be done before animating complex animation systems which strive for elegance. A system of rules, which give the animator parameters to work with, should be established early on. This seems particularly critical if multiple artists are working on the project.

10.4 Context maintains consistency

Establishing the context of the environment and character seemed to help in creating animations with consistent feel and appeal. If these contexts were not available for the animator, the design of the PC and the animations for the PC would lack many of their features. This in turn would have led to issues in clarity, feel, and overall appeal of the animations.

By having a reference point for the animator to return to either in story, environment, or other pre-established context points in the project the animations are united. Steps to clearly communicate the necessary context information to the animator should be taken. Using animation maps, concept art, written text, and references from other games will assist in communication between the animator and the rest of the development team.

10.5 Principles serving the gameplay

The principles of animation were used as foundational rules in the animations. They assisted in designing the motion within the animations and creating appealing flow, feel, and clarity. Through improving the animations, they improve the gameplay.

By using the animation principles in game animation, an animator can have a clearer production process for animations. The principles need to be implemented at specific stages. Producing animations with these stages in mind gives the animator clarity and milestones to strive towards. The animator can also improve the flow between animations by utilizing the principles.

The animation principles can bring appeal and spirit into the game with a fitting art style. The design of the PC and the animations allowed the principles of animations to be used extensively in the project. More visually realistic games might need to utilize the principles of animation sparingly to avoid uncanny motions.

10.6 Future corrections

The symmetry of the PC brought issues of clarity. During the running animation, it was originally difficult to decern which leg was closer to the camera. This caused clarity issues within the running animation and reduced some of the energy of the motion. Shading one of the legs was used to differentiate them from each other. The shading had to be done for the legs in all animations, not only the running animation, to maintain consistency in the design. To maintain similar consistency, all future animations need to have one of the PC’s legs shaded, which is extra work for the animator to maintain fluidity.

This shading may bring issues of lighting in the game later. If the PC has too clear directional lighting established in the animation, having a different lighting direction in the environment when the PC turns could cause issues in appeal and breaking of immersion. The shading may not be usable if the game was to be finished, but for the testing portion and current version it suffices. This issue could be researched and tested further.

The code which runs the PC seems to be easily broken by the player. It is not usable for a finished product. It should be made functional and not as fragile before further animations are produced so that the animator knows what is possible in the game. This will further assist the animator to gain context and design their animations with these new factors in mind.

10.7 Technical function

Observing the production from the perspective of a technical animator displays some of its lacking aspects. A technical animator is meant to be responsible for both the production and implementation of the animations. The implementation of the animations had difficulties due to the lack of coding skills of the author. More time should have been prepared for the research and the production of the code.

Since the animator is defining much of the gameplay with their animators the technical animator shares in the responsibility of perfecting their implementation to gameplay. Most of the mistakes were not because of the animations themselves, therefore the animator has succeeded in producing material which the technical animator can proceed with. There did not appear to be any extra workload which was not intended by the gameplay’s design or the animations’ design. More research or questions from experienced technical animators should be done to discover if the envisioned gameplay is difficult to achieve. The animations could be played to mimic the phasing of the gameplay, thus it is expected the gameplay is possible to be produced with the current animations.

Despite not producing fully working code the intended function of the animations could be tested. This provided knowledge on how problem solving with animation implementation is done and what the process demands. Creating code to resolve problems often creates new problems and custom animation parameters must often be created to adjust for input response time. These animation parameters often clash with one another and create errors in code. They can also create errors in how the animations are played within the game engine. Starting with a simple code, which enables the PC's basic actions, may make more complex additions, such as movement during attacking, easier in the future.

Having a prepared template of code for character movement could make testing the animations faster. If the animator took time to prepare a general movement system code, they could simply drag animations into the animation state machines. The animations could be tested while they are being produced. This is how game production seems to normally work in larger productions with other team members. However, an animator working alone as both the animator and technical animator would benefit from preparing such code for many future projects.

10.8 Overall production and final thoughts

The overall production was ambitious in its scale. Many animations in the animation map, which would have been interesting to test, were left unproduced due to time constraints. Production of this scale could have been better suited for two or more animators to test these theories.

More time should have been designated to studying coding and learning coding. This would have enabled expanding the perspective of a technical animator. However, this does demonstrate how much skill a game animator with both animation and technical animation skills requires.

Producing the animations alone and attempting to do the work of a technical animator revealed new skills which must be acquired to become a professional game animator. Knowledge of overall basic coding is not enough, but the ability to recognize problems and find their cause through deduction is needed. To improve as a technical animator, one should test multiple solutions to a problem to see which one has the least chance of creating additional problems in the future.

Managing and estimating game animation production can now be done more accurately. The animator can divide the animation production process to clear stages and methods to achieve professional level results. Estimating how long each stage takes and which methods bring appealing results without the cost of extensive time are valuable knowledge to have when communicating with other team members or setting production goals.

The lack of an enemy NPC leaves many animations' main priorities untested. Producing an enemy for the player to fight against should be the next priority after producing a few more gameplay necessary animations, such as jump and parry. This should produce more definite results for how the animations serve the gameplay.

The animations which were tested did not all work as intended, as shown by the turning visual error, but the reason for their problems could be explained. If the reason for the error is known, it is then a matter of learning and fixing the issue. This was one of the successfully achieved goals of the testing. The aim was not to prove all the animations work flawlessly, but to see which work and do not work and attempt to find out why.

The core motion pillar proved to be a helpful working method. It ensured the quality of the animations stayed consistent and served the gameplay. However, it did not necessarily speed up the animation Process. For the future, creating rough animation versions first to be tested in a game engine before creating a core motion pillar could help in avoiding creating unfitting animations. It would also give the rest of the development team assets to work with faster.

Sources

AlanBeckerTutorials. 2017. 12 Principles of Animation. https://www.youtube.com/watch?v=uDqjIdI4bF4. 20.03.2022.

AnimationExchange. 2021a. Animation Exchange 2021 – When Will The Attack Land. https://www.youtube.com/watch?v=E-hQOq3pbCQ. 23.03.2022.

AnimationExchange. 2021b. Animation Exchange 2021 – Why Gameplay Animators Should Be More Technical. https://www.youtube.com/watch?v=TBMjqPApLxQ. 22.03.2022.

AnimationExchange. 2021c. Animation Exchange 2021 – Technical Q&A. https://www.youtube.com/watch?v=SMATCF9UWts. 25.03.2022.

Cooper, J. 2019. Game anim: video game animation explained. Boca Raton, FL: CRC press/Taylor & Francis Group.

GDC. 2021. Take 'Control' of Animation. https://www.youtube.com/watch?v=JH69g7yA7QM. 23.3.2022.

Howard Wimshurst. 2016. How to Design characters For Animation. https://www.youtube.com/watch?v=FCdWY7nHWDA. 16.02.2022.

Howard Wimshurst. 2018. The Genius Behind Naruto's Fight Scene Animations – Norio Matsumoto. https://www.youtube.com/watch?v=i6Z9m_0-DVo. 22.10.2023.

Joe van den Heuvel. 2021. Character Locomotion in half-Life: Alyx – SIGGRAPH 2021. https://www.youtube.com/watch?v=RCu-NzH4zrs. 23.03.2022.

Johnston, O. & Thomas, F. 1981. Disney Animation: The Illusion of Life. New York City: Abbeville Press.

Korshak, I. 2020. Illustrator's Eye: The Language of Shapes. Medium. https://medium.com/@irynakorshak/illustrators-eye-the-language-of-shapes-88b93cd259a1. 28.03.2022

Mattesi, M. 2006. Force: dynamic life drawing for animators. Boston: Focal cop.

Milner, D. The Making of Hollow Knight. Gameinformer. https://www.gameinformer.com/2018/10/16/the-making-of-hollow-knight. 17.10.2023

New Frame Plus. 2021. FOLLOW THROUGH & OVERLAPPING ACTION – The 12 Principles of Animation in Games. https://www.youtube.com/watch?v=rYtrV1IChsA. 22.03.2022.

New Frame Plus. 2020a. ANTICIPATION – The 12 Principles of Animation in Games. https://www.youtube.com/watch?v=28s1Hv3Zqlo. 22.03.2022.

New Frame Plus. 2020b. SLOW IN & SLOW OUT – The 12 Principles of Animation in Games. https://www.youtube.com/watch?v=3jNiNctcQ4c. 23.3.2022.

New Frame Plus. 2019a. TIMING – The 12 Principles of Animation in Games. https://www.youtube.com/watch?v=rHEJZXvFc5I. 23.3.2022.

New Frame Plus. 2019b. SQUASH & STREATCH – The 12 Principles of Animation in Games. https://www.youtube.com/watch?v=1kFRU_xBZnE. 24.11.2021.

New Frame Plus. 2019c. The Effects Animation of Hollow Knight. https://www.youtube.com/watch?v=SIJtfr-PO4Y. 24.10.2023.

OpenAI. 2023. ChatGPT. https://chat.openai.com/. 25.10.2023.

Sloan, R. 2015. Virtual character design for games and interactive media. New York: AK Peters/CRC Press.

Ubisoft Montreal. 2020. Introducing Learned Motion Matching. https://www.ubisoft.com/en-us/studio/laforge/news/3NWqwcMU9IMfumgfPAEfka/learned-motion-matching#:~:text=Motion%20Matching%20is%20a%20simple,create%20new%20transitions%20between%20states. 25.03.2022.

Unity. 2023. Animation State Machines. https://docs.unity3d.com/Manual/AnimationStateMachines.html. 25.10.2023.

Toniko Pantoja. 2018. Animation Smears – The Hand Drawn Motion Blur. https://www.youtube.com/watch?v=KXuP2_BCmV0. 24.10.2023.

Video Game Animation Study. 2020a. Game Animation Fundamentals: Fluidity. https://www.youtube.com/watch?v=ZutXSJhBTi4. 4.11.2021.

Video Game Animation Study. 2020b. Game Animation Fundamentals: Readability. https://www.youtube.com/watch?v=ynL_3zTiom4. 4.11.2021.

Video Game Animation Study. 2020c. Game Animation: Context. https://www.youtube.com/watch?v=w9IM-RPXmvs. 4.11.2021.
Video Game Animation Study. 2020d. Game Animation: Elegance. https://www.youtube.com/watch?v=ma4SoiEcKwQ. 4.11.2021.
White, T. 2006. Animation from Pencils to Pixels: Classical Techniques for Digital Animators. London: Routledge.
Williams, R. 2001. The Animator's Survival Kit. London: Faber and Faber.

List of games, songs, and programs

Clip Studio Paint. 2001. Celsys.
Crowsworn closed demo. 2023. Mongoose Rodeo.
Half-Life: Alyx. 2020. Valve Corporation.
Hollow Knight. 2017. Team Cherry.
Mr. Blue Sky. 1977. Mr. Blue Sky. Electric Light Orchestra.
Sekiro. 2019. FromSoftware
Star Wars: Jedi Fallen Order. 2019. Respawn Entertainment.
Unity. 2005. Unity Technologies.

Appendix

Testing the animations within Unity-video

A separate video which displays the animations being tested within unity. Additionally, the video contains the animations in MP4 file format.