DEPARTMENT OF INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Bachelor's Thesis in Informatics: Games Engineering

Applying Methods of Game Feel

Robin Ostner

DEPARTMENT OF INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Bachelor's Thesis in Informatics: Games Engineering

Applying Methods of Game Feel

Anwenden von Methoden von Game Feel

Author:	Robin Ostner
Supervisor:	Prof. Gudrun Klinker, Ph.D.
Advisor:	Christian Eichhorn
Submission date:	February 11, 2019

Declaration

I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Ottobrunn, February 11, 2019 _____

Place, Date Signature

Table of Contents

1. Introduction

Why are some games more fun than others? A very common answer to this question is that some games have better gameplay. It is true that at the base of every good game is good gameplay, but gameplay is not something completely independent of all the other things in a videogame. For example, would *Super Mario 64* still be so much fun, if instead of controlling Mario with all his animations and sound effects, you would control a rectangular box? Or if the characters in *Street Fighter* didn't show any signs of damage taken apart from the health bar on the top? All the game mechanics could still be in place, but you wouldn't feel as though you are playing the same game. Apparently, there is something other than pure gameplay, that takes effect on how much fun it is to handle a game. Some people call this *"*Game Feel*"* or "Friction", but nobody actually understands how it works or what causes it. The game designer Steve Swink first coined the term "Game Feel" in his book "Game Feel: A Game Designer's Guide to Virtual Sensation" (Swink 2008), which has been the bible for this topic ever since it was first released. He tries to dissect this topic and explain the phenomenon. Even though it doesn't completely explain everything, it is the best attempt at doing so to this day. Other people have talked about this and tried to explain some parts of it, but most game designers elegantly try to avoid this topic. This is due to the fact that it is very hard for a human being to describe what they feel when playing a game, because most of it happens subconsciously. Why does it feel so different to play *Mario Kart* compared to *Gran Turismo*? You could say that one handles more realistically than the other. But the cars in *Mario Kart* still feel like cars. There is no way for us to accurately describe what we feel when playing videogames. We simply don't have the vocabulary for these things. The best thing we can do is to compare certain games to other games with similar mechanics, but that is not a very precise solution as everyone has a different opinion on games. This is why it is so hard to talk or even write about it.

Nonetheless there is a need to talk about this very big topic, as it is connected to the root of good game design. It is important to talk about and analyze it so that maybe one day we don't have to rely on comparisons to other games when we want to describe how a game feels.

1.1. Motivation

When talking about game design most often it is about level design, game mechanics and the rules of the game. But one very big topic which is often not discussed at all is game feel, even though it is the most important aspect of the moment to moment action in a videogame. Of course, some game genres don't rely so heavily on this, for example a puzzle game, but even those can really benefit from having satisfying controls and

giving the player good feedback about his actions. *Candy Crush Saga* would never be as popular, if it was simply colored boxes that disappear without any animation, sound effects or particle effects. So why does nobody talk about it? The biggest reason is probably that there is no accurate way to describe and test it, but that doesn't mean that we shouldn't try to establish some common terminology. To get to this we first have to start talking about the topic in general and not only as a side effect when trying to describe why a game is good or not. Game designers should actively try to create good feeling games instead of trying to create the perfect game mechanic. These things obviously go hand in hand, but game feel is often neglected as being a fundamental part of the game experience. Steve Swinks book about this topic was a huge step forward in this direction but it wasn't enough apparently. Some game developers have given talks about it as well, but these are very few and far between. If you search for "Game Feel" on Google, almost all of the results show the book of Steve Swink. This is not good because young game designers are not aware of this at all, which is why there are so many new mobile games on the market which simply feel limp and lifeless. Everyone would benefit from focusing more on the feel of their game and establishing new guidelines to creating good feeling games.

1.2. Goals

This is a huge topic and trying to explain all of it in a 50-page thesis is doomed to fail. Therefore, the goal is to simply talk about various aspects of game feel and to test the established guidelines and methods on a simple 2D platformer game. The first part will be about what game feel is and what causes it. To help others talk about this topic there is a need for vocabulary and to get to that we first have to understand what it is we are talking about. Secondly, we will talk about certain aspects and mechanics of game feel. To do that we are going to compare other games to one another and look at the things they have done to create this good feel. This will also be about what makes games feel bad and why. Another very important step towards creating better feeling games is having a way to test if a game feels good or not. We will be looking at methods other game developers have used to test and create the feel of their games. After we have established what game feel is and how we can test it we are going to apply the results to a very simple 2D-Platformer game made specifically for this purpose. The players will have to complete a short level five times with different effects applied to the game. This game will be tested, and the results will then be compared and analyzed. This will show that it is indeed possible to test a game's feel and that it has a surprisingly big impact on the game. But the biggest goal for this thesis is to start the conversation about game feel in general. Game designers should think about this as well as all the other things, which are already so deeply connected to game design.

2. Game Feel

2.1. Defining Game Feel

As I already said during the introduction, we are going to start by trying to dissect this huge topic of game feel by looking at the knowledge we have so far, as well as comparing different approaches to creating good feeling games.

So, what exactly is game feel? Why does the combat in *Street Fighter* feel so good? Why does dodging an asteroid in *Asteroids* make your fingers tingle with excitement? Why is it so much fun to just jump around in *Super Mario 64* even if there were no goals? And why does the infamous *E.T.* game feel so bad? There is almost no way to describe the sensation of playing these games without referencing personal experiences and comparing it to other games. St. Augustine's comment, when he tried to define time, seems very fitting:

> *"What, then, is time? If no one ask of me, I know; if I wish to explain to him who asks, I know not." (Saint Augustine 2003)*

Something so abstract and embedded in our subconscious that as soon as we try to make sense of it, it disappears into nothingness. It is as though you were trying to explain the colors of the rainbow to a blind man. What does red look like? You know exactly what it is, but if you were to describe it, you wouldn't know how. It is the same for trying to explain game feel. At the moment the best way to describe a games feel is by experiencing it, but that doesn't mean that we have to give up on this quest.

One of the biggest problems when talking about game feel is the vocabulary. Even the name "Game Feel" itself is not very good, since it can be very easily confused with feelings in an emotional sense. Emotions probably have next to nothing to do with a game's feel. The simple act of controlling a character doesn't make you sad, does it? It is much more connected to feeling in a physical sense. The sensation you have when touching an object or driving a car. A rock in your hand feels rough and cold, whereas fur feels soft and smooth. Running in shoes feels a lot different to running on bare feet. This shows that the term "Game Feel" might not be the best, which is why some people use a different term. Tim Rogers calls this phenomenon *Friction* (Rogers 2010), which at least has something to do with the physical sensation of touching something. But in the same way "Game Feel" is too broad I think that Friction is too narrow for describing this phenomenon as a whole. It can lead people into thinking that it has only to do with how much the characters slide. Another term I have read about when researching this topic was "Kinaesthetics". The games critic Chris Franklin first started using this term in combination with games. Kinaesthetics is the study of body motion and in particular the

study of one's own body motions and how we perceive it. It is why we know exactly where all our body parts are, even if we are not looking at them. This is getting a lot closer to what the topic is about, which is why Chris Franklin started using it. The problem with this term is that the kinaesthetics of a game are only a small part of the whole, but we will look into that in more detail later. So even though Franklin acknowledges this fact he still thinks that it might be a better term for it. In one of his video essays he states:

> *"While I love Swinks book, I find the term 'Game Feel' to be unwieldy. The phrase 'Kinesthetically pleasing' to me has a far better ring than 'This game has really good Game Feel' " (Franklin 2012)*

Even though I can agree with most of the things he said about the terminology I have to stick to the definition of Steve Swink. This doesn't mean that the term "Kinaesthetics" is bad, but since I am simply trying to start a discussion about the topic, I have to use the most common term, which is "Game Feel". Finding an appropriate name for it might be an issue but it is not the most important. After all, there are tons of topics in the game industry, which don't have a proper name or definition. Think about the term *Social Games* for example, which usually only describes the business model of the game, but not the game itself. This is due to the fact that videogames are a very new topic and a quickly evolving one at that. A much more important problem with game feel is trying to define and understand it.

Let's begin with the most obvious choice for a definition first, which is the definition Steve Swink used in his book:

> *"Real-time control of virtual objects in a simulated space, with interactions emphasized by polish" (Swink 2008)*

He dissected this topic in what he calls "The Three Building Blocks of Game Feel" (Swink 2008), which are Real-Time Control, Simulated Space and Polish. Real-time control meaning that the player has to be in full control at all times, or at least feel as though he is. This means that for every input the game has to show the changes instantaneously. For the average person to perceive these changes as instantaneous the whole process cannot take longer than 100 ms, according to Swink. This means that the computer has to continuously receive and process inputs and display the results at a constant rate of at least 10 frames per second. This enables the player to make small corrections to his inputs in the same way you would adjust the steering wheel when making a turn in a car. It is all a continuous flow of control, without any interruptions. It feels as though you are perceiving the changes and acting on them in the same moment. But being able to control anything, you first need some context as to what you are doing. This is where

the Simulated Space enters the stage. Imagine controlling Mario in a completely white room. There are no shadows, no textures and no objects. The screen is just white and Mario's character, nothing else. All of his moves and jumps would lose all meaning because you have no context for his surroundings. The space around the character is just as important as the character itself, since it gives meaning to everything the character does. Using this we can create a "tactile [and] physical sense of interacting with virtual environments the same way we interact with our everyday physical spaces." (Swink 2008). This also means that there has to be a way to actively perceive the surroundings. If there are many objects in the space around the character, but the character cannot bump into them or interact with them in any way, they lose all their meaning. So, the Simulated Space also includes the collision detection associated with these objects. The last building block of his definition is Polish, which is basically everything that is not essential to the game itself. This includes animations, audio, particle effects, screen shake and everything that is simply there to add appeal to the game. Polish can be very easily confused with the simulation since it is simply there to make the simulation seem more sophisticated and real. Videogames probably would have never taken off as one of the biggest industries in the world if it wasn't for Polish, since the player wouldn't enjoy the experience of playing the game nearly as much, even if the underlying simulation was exactly the same. Using these three building blocks Swink was able to create his definition of game feel. As I said earlier this is to this day the most sophisticated attempt at defining this very important aspect of games, but it has some problems.

The first thing I would like to mention is that this definition applies to more than just games if you take out the word virtual, which might not be a problem at all. But this makes it look like the definition is not precise enough, for what it is trying to say. Driving a car for example fits perfectly within this definition, since it is happening in real-time, has a space in which it is happening, and is emphasized by polish. Polish in this case would be the power steering, or the soft seats, again anything that is not essential to driving the car. Another problem is that he excludes every game which is not played in real-time. His definition basically says that a game like *StarCraft* doesn't have game feel, which a lot of people would probably disagree with. The actions on the screen are playing back in real-time but the interaction of the user with the game is happening at discrete time intervals. If the clicking of the mouse in that game would be delayed by just 100 ms the players would feel disconnected with the action on the screen, thus making it feel worse. This is a huge chunk of games which would be excluded by his definition, even though they clearly have something, which makes them feel good or bad. Swink acknowledges this problem in his book but he decided to include only games with what he calls "true game feel", but he also says that these other kinds of feel are

also important. This makes the whole definition a little wonky, since it doesn't draw a clear line between games that have game feel and games that don't. He even brings up the example of *Diablo*, where the line of real-time control is even more blurry than with *StarCraft*. So there clearly is a problem, which might be solved by calling the general topic Kinaesthetics or Friction and then defining subcategories, with one of them being game feel. But all of this won't be solved by a 50-page thesis, which is why there is a great need for some proper research and discussion about this topic.

2.2. Metrics of Game Feel

Defining what game feel is and what the correct or best terminology for it would be is one problem, but another big problem is measuring it. All the talk about what it is would be worth nothing if there is no way to apply it. This is why Steve Swink tried to find ways in which to measure game feel, which he calls "Metrics for Game Feel". This section will cover all of his definitions and guidelines and look at ways they could be improved.

The problem with trying to measure game feel is the same as with defining it. Talking about it and describing a game's feel is nothing concrete and always leaves room for interpretation. Vocabulary and terminology are one of the biggest obstacles we have to overcome to achieve this goal. Since there are no established measurements for game feel, designers often have to use very simple terminology and have to rely on comparisons to other games just for casual conversation. Derek Daniels, a designer for the *God of War* Series, sums it up quite nicely in one of his blog posts about why some games feel better than others:

> *"Because you have to reinvent the wheel each time out, you get people who either don't care or simply don't have the knowledge of what it takes to have their characters feel good when a player first picks up the controller." (Daniels 2006)*

This leads to the problem that most of the time designers try to recreate the feel of certain games instead of creating their own unique feel for their specific game. Incrementally improving the game's feel and trying to get closer to the feel of another game is the safe and easy route but it leads to uninspired works most of the time. There is nothing fresh about those games, since all they do is copy the work of someone else. The hard way is to actually think about how the designers of *Super Meat Boy* got to this satisfying control scheme, or where the feel of *Super Mario 64* came from. In the case of *Super Mario 64*, it is safe to say that they could not have cloned any other work since it was the first of its kind, so how did they do it? To understand this, we must look at all the interconnected parts of these games and how they affect each other. We have to identify the pieces, which contribute to a game's feel and find out how they all work

together. Looking at the big picture and wrapping our heads around the system as a whole and what experiences the player has when interacting with it.

According to Swink there are six key metrics, which we have to observe, when tackling the problem of game feel. These metrics are all things the designers can influence directly. They are linked to the definition of game feel, which includes the real-time control, simulated space and polish. The key metrics are *Input*, *Response*, *Context*, *Polish*, *Metaphor* and *Rules*, and we will go over each one and how they contribute to a game's feel according to Steve Swink.

2.2.1. Input

First on the list is the Input metric. This covers everything which is strictly concerned with how the player can tell the game what to do. Everything from a keyboard to a PlayStation gamepad and from a standard button, which only has two states, to a thumbstick, which essentially has an infinite amount of possible states. Swink says that the type of input can heavily influence the feel of a game. Just as an example think about how playing a realistic racing game would feel like, if you would play it on a regular keyboard. It doesn't feel as good as playing it with an actual steering wheel and pedals. This is because the intention of turning the car doesn't translate very well when playing with simple two-state buttons. A steering wheel has a much higher sensitivity and range in inputs. Also, the input type of the steering wheel is rotational, whereas the input of a standard button is linear. It just doesn't feel right. There are many different kinds of input and how the properties of the device translate to the action they are trying to represent is imperative for the player's feeling of being in control.

All of this is the micro-level of Input, but we also have to consider the bigger picture. This means we have to look at the tactile properties of the input device as a whole, which includes weight, material and the quality of the buttons. Just compare the feel of playing a game with an original Xbox-Controller with using one of the cheap knockoffs. The weight is off, and handling it just feels really cheap. The first PlayStation 3 controllers had a big problem with this issue. It was said that they were too light and felt really cheap, which is not something you would expect from a 60 € input device. One thing, which might need some more explanation, is the button quality. What Swink means by that is the "feel of spring resistance". This includes everything from how stiff the spring in the button is to how much travel the button has. Even porting a game from one console to another can lead to problems, where the players report that the game feels "just not good" on different platforms. So, creating a good feeling game requires some consideration of the input device, apart from the regular control scheme for the game. It is the only way for the player to interact with the virtual environment, so we have to make sure that this connection feels right.ResponseThe second metric Swink proposes to be

very important is Response, which is how the game interprets the inputs and calculates the results. So basically, the output of the system. The first thing to consider is how the inputs are mapped to changes in the game. Is the value of the input mapped directly to a parameter in the game or does it rather control the rate of change? How all of these mappings are connected to one another and how they are modulated is the main concern of the Response metric. The movement in the original *Super Mario Bros.* for example is not mapped directly to the input. It much more controls the acceleration of the character, which contributes a lot to the feeling of weightiness of Mario. For the real-time control of the game you will always have the inputs linked to the movement of the character in some way. This movement can be defined in many ways, including the type of motion, being linear or rotational, or the dimensions of motion, which could include any of the three spatial axes. When looking at the types of motion the character has it gets easier to choose a mapping for the inputs, since the input devices share some properties, like the rotation of a steering wheel for example.

Swink also goes into ADSR envelopes, which are usually used to describe how a sound changes overtime, but it essentially describes the modulation of a parameter. This method can be very useful when describing the relation between input and game parameter. ADSR stands for Attack, Decay, Sustain and Release. Attack being how fast the input ramps up to its target value, when the button is pressed. Sustain is the designated level for the value as long as the button is pressed. The time it takes for the value to drop from its initial peak or attack phase to the sustain level is called decay. After the button is released the value drops back to zero. This phase is called release.

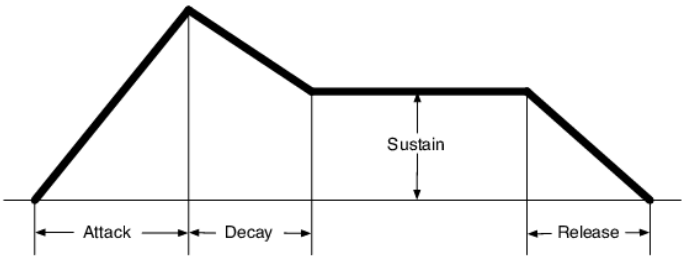

Figure 1 Elements in an ADSR Envelope, Kvifte

Using this we can look more specifically at which envelope types evoke a certain feel. If the attack time is longer than the previously defined 100 ms, the controls feel sluggish and unresponsive. If the attack time is too short the game can start to feel twitchy. Modulating these envelopes and trying different approaches can lead to very different feeling games and can be very useful when trying to perfect a game's feel. For designers they even serve as some kind of template for creating certain types of feel, without having to start from scratch for every game. They can also be quite useful when

comparing the inputs of two games or when you want to analyze how the feel of the game has been created.

Another important aspect to consider is how the state of the game influences the way the inputs are mapped to the values of the game. In the *Super Mario* games there are two different states for being on the ground and in the air. For both of these states Mario controls a little bit different, which makes the move-set of Mario a lot more expressive. The combination of different inputs can also be something which can lead to a different feel. Pressing the *A* button can have one result and pressing the *B* button can have another, but when combining the two inputs it could yield a third response, which ultimately creates a huge number of possible combinations for the player to use. All of this influences how the game's system reacts to the player's input and I have only been scratching on the surface of what's possible. There is an immense number of ways in which we can manipulate the inputs of the buttons to create a unique feeling game.

2.2.2. Context

Context is another metric defined by Swink, which could be very useful for measuring the feel of a game. He describes it to be everything, which is involved in the simulated space aspect of the game. Most often this is about how the objects in the world are spaced to one another and how the physical interaction between objects is handled. He additionally divides it into three subcategories: High, Medium and Low-level context.

The highest level of context is the impression of space, speed and motion. Let me give you an example for the impression of space and how it can affect the game's feel. Compare the maps of the *Battlefield* series to one of the *Call of Duty* series. In *Battlefield* you usually play on big open maps, where you have to walk a considerable distance to reach the far end. This creates a very open feel of space, which greatly contributes to the strategic part of playing *Battlefield*. The maps from *Call of Duty* are pretty small in comparison and can sometimes feel a little claustrophobic, which is not a bad thing. Having a tighter space to play in creates much more tension and leads to a more reckless and spontaneous play style, which is what these games are trying to achieve. By simply comparing the space of these two games the difference becomes very clear. This can also affect how the player experiences the sensation of speed and motion. Since there is no standardized way of calculating the speed of a virtual object, the speed of an object given by the game doesn't mean anything, when being compared. Everything is simply measured relative to the environment of the game. This means that if there is nothing to compare the speed to, the movement of the object becomes meaningless. The spacing of the objects, their size and relative motion to the character are the most important aspects for the player to determine how fast he is moving. This effect can be enhanced by using effects like the doppler-shift and motion blur, but those

would fit more under the Polish metric. Depending on these parameters the movement will seem faster or slower.

The impression of size is another interesting aspect of high-level context. Swink explains this by using one of my favorite games *Shadow of the Colossus*, so I will follow his lead and use it as well. When you play this game you really get a sense of how gigantic the colossi really are. Team ICO achieved this incredible sense of size with animation and a few clever perspective tricks. When you are far away from one of the Colossi it seems to be moving rather slowly but when you move closer you realize that it is not moving slowly but that it is simply huge. If the Colossi were to move at normal speed like a human being would it would seem really off. By slowing down the motion you create a feeling of weight and size. A lot of games forget about this very important aspect of the scale of objects, which in turn completely breaks the suspension of disbelief. In *Shadow of the Colossus* they used the camera and perspective to emphasize this impression of size. The camera always tries to frame the character and the colossus in a way that the colossus is towering over the character.

Figure 2 Shadow of the Colossus, Team ICO

This little trick together with the stunning animations, and polish effects like the floor cracking underneath their feet and making the screen shake with every move, really hammer home the fact that the character is tiny in comparison to the colossus.

At the medium level is the “feeling of immediate space and object avoidance” (Swink 2008). This is connected to the spacing of the objects around you at an immediate level. At the higher level of context this was referring to the spacing between the different camps in *Battlefield* or *Call of Duty*. Here it is more about how the turning radius of a car is meaningless without a track to drive on. You have to compare the number and size of the objects and the distance between them in relation to the speed of the character if you want to look at the medium level context. This effect is even present in real life. If you think about driving your car at 100 km/h on the highway it usually feels rather slow, but the same speed would feel ludicrously fast if you were to drive on a piece of twisting

mountain road. This can be used to a great extent to tune the challenge of games, by increasing the challenge of navigating the environment.

When you look at the level of tactile interaction you have arrived at the low-level context. This is mostly concerned with the physical interaction between objects. Again, I will be using a simple comparison to show what is meant by this. Looking at the collision detection in the *Mario Kart* series it is very clear that the interactions of the cars with other objects is not very realistic. They bounce off the walls and barely slow down when scraping them. The collisions usually feel very stiff, like two blocks of metal banging into each other, where there is barely any friction. Comparing this to the more realistic game *Race Driver: Grid*, where just scraping the walls could lead to a fatal crash and your car looking like it came straight from the junkyard. The collisions are much rougher and malleable than the ones of *Mario Kart*, which leads to a more realistic feel. This is essential for the interactions with the game world to feel right.

2.2.3. Polish

The fourth metric is Polish. This is essentially everything, which is added on top of the simulation without actually having an effect on it, to artificially enhance the game experience. When a car bounces off a wall in *Mario Kart* this is not polish, since it is directly driven by and interacting with the simulation. Polish is everything that is non-essential to the underlying simulation.

Earlier I mentioned that *Shadow of the Colossus* uses some polish effects to enhance the players experience of scale, which I will now take as an example. When the colossi move their feet the whole screen shakes uncontrollably, which makes the impact a lot more powerful. The floor cracks underneath their feet and spawns a lot of particle effects in the form of dust and debris to show that the simple act of moving around involves a lot of force, so you better not get underneath their feet. The animations themselves can also be considered as polish, and the same applies to all the sound effects and the music. The movements of the colossi are greatly enhanced by the deep rumbling noises and the sounds of debris crashing into the earth. All of this is done to make the colossi feel more massive.

Polish is crucial for the player to learn how to navigate the virtual environment and how certain objects react to an action. It is essential for the player to receive precise feedback about his actions. Our whole lives we have spent perfecting this with our bodies and our brain is trained to look for clues about the physical properties of an object. Without polish the players would lose their footing, which they need to cope with a virtual environment. As I said earlier polish includes many different things, but it can be categorized into five types.

Animation is the first on the list, since it is one of the first types of polish to make its way into the games world. One of the first games to feature animation to enhance the players experience is *Pac-Man*, but since then we have come a long way. Most games nowadays use the same principles of animation as traditional film animators, like “squash and stretch” or “follow through”, to really sell the physicality of the characters and objects involved. One game series that consistently nails this is the *Super Mario* series. Without the animations these games would not feel even remotely as satisfying to control. The game could still work in the same way but what really sells the movements of Mario are the animations.

On top of the animations these games also use a lot of the second type of polish, which is Visual Effects. All the little dust particles at the feet of Mario and the tiny sparks when collecting a star are visual effects, which are there to enhance the animations. They can be applied to all kinds of things and are mostly used to highlight the interactions of objects. The spray of debris when crashing through a billboard in *Burnout: Paradise*, the small light trails behind characters in *Ori and the Blind Forest* and the blood or goo splatters in *God of War.* All those effects can be categorized as visual effects, but usually they are referred to as particle effects.

The third category Sound Effects is much like Animation as it has been studied quite extensively and is usually referred to as Foley in the film industry. Sound Effects can be used to great effect, if done right. The game *Hyper Light Drifter* is a great example for good game feel and one of the reasons for this is the exceptional sound design. Slashing your sword through enemies has a nice crunch to it and gives the player perfect feedback. The sound of the sword alone gives the player the impression of wielding a fast and powerful weapon. Picking up an item is enhanced through a very satisfying clicking and surreal tinkling. Just about every action is underlined by some sort of sound, which adds a lot to the immersion.

Jan Willem Nijman gave a great example of how a sound effect can completely change the feel of a game. He talks about how a Senior Designer at BioWare was told by the investors to fix the gameplay of a gun in one of their First-Person Shooters. He simply opened an audio-editing-software, applied a bass boost to the sound of the gun and put it back in the game. After he changed the sound the investors said that the weapon now feels a lot better and that he has indeed fixed the gameplay (Nijman 2013). Akash Thakkar, the lead audio engineer for *Hyper Light* Drifter, has now published a book about this very topic titled “Maximizing Game Feel Through Sound: A Composer/Sound Designer’s Approach to Upgrading your Game’s Impact”.

Another type of Polish which has been adopted from the film industry are Cinematic Effects. Steve Swink describes it as “things like screen shake, changes in view angle, motion blur and Matrix-style slowdown” (Swink 2008). The indie game studio Vlambeer is widely accepted as the king of screenshake. All their games wobble uncontrollably on explosions and gunshots. It really emphasizes the impact of certain actions, which is further emphasized by a small time freeze with each hit. This is when the action of the game is stopped for a really short amount, so that the player can barely notice it. Games from the *God of War* series or *The Legend of Zelda* series have used this many times to great effect. Motion blur is often used in racing games, because it gives the player a nice sense of speed. There are many tiny things you can do with this, but you always have to be careful to not overdo it when putting it in your game, as you can get “addicted” to it and don’t feel the impact as much.

The last thing on the list of types of polish are the Tactile Effects. Most players experience this as rumble in the controller. It can be used in the same way as screenshake, but it can also give the player valuable feedback about the game status. Racing games often use it to indicate to the player that he is losing grip, but those games have another great way of giving tactile feedback. The racing wheels for simulators are almost always fitted with some sort of motor, which can give the player force feedback. This means that the wheel is actually pushing against the actions of the player, which is also referred to as haptic feedback. By using this force feedback, the players can get an extensive amount of information about the car’s status and how much grip they have in a corner.

There is a lot more that you can do to enhance the simulation of a game and we will be looking at a few games in a little bit more detail in the coming section. Polish is a really powerful tool if you know how to use it effectively, but everything that is done, serves only one purpose: To enhance the simulation. There cannot be a game made purely from polish, since there must always be a foundation first for it to work.

2.2.4. Metaphor

In the introduction of this thesis I have talked about why the cars in *Mario Kart* still feel like cars, even though they clearly are not trying to be realistic. This comes down to the fifth metric, which is Metaphor. It can be divided into representation and treatment, where representation is the idea of what something is supposed to be and treatment is the “cohesive whole formed by visual art, visual effects, sound effects, tactile effects and music” (Swink 2008). The representation in *Mario Kart* clearly communicates to the player that he is controlling some sort of car, but the treatment is not trying to explicitly define what exact model it is supposed to be. By not trying to be realistic the players expectations are set to a lower standard than if the cars would look photorealistic, like

in *Gran Turismo* for example. This is the reason why the cars in *Mario Kart* still feel right. There is no disconnect between what the player is expecting from the games simulation and what it actually does. Most games get this right at least to a certain degree, but there are barely any games trying to leverage this aspect of game design. Some games even get this completely wrong, which leads to a game that just doesn't feel right for some reason.

The game *Super Street: The Game*, a racing game from 2018, is one of them unfortunately. The visuals show a rich and realistic environment and the cars are rendered in a photorealistic way and crashing into stuff deforms the car, while little pieces of it fly through the air. But even though crashing into stuff looks phenomenal it still doesn't feel satisfying. It even feels a little odd. The sound that accompanies all those crashes doesn't match the visuals at all. Instead of hearing all the pieces crashing into each other and having a big satisfying impact, all the game gives you is a tiny pathetic screeching noise, which cuts out after not even a second. The game's soundtrack gets muted a little bit as though it wanted to highlight the big crashing noise, but unfortunately this never comes. It feels a little as if the developers forgot to enable the sound effects for it. This is just one example, where a mismatch between the metaphor and the rest of the game had a negative influence on the game's feel.

Swink also goes into a concept by Scott McCloud, which defines the metaphoric space by using a ternary diagram.

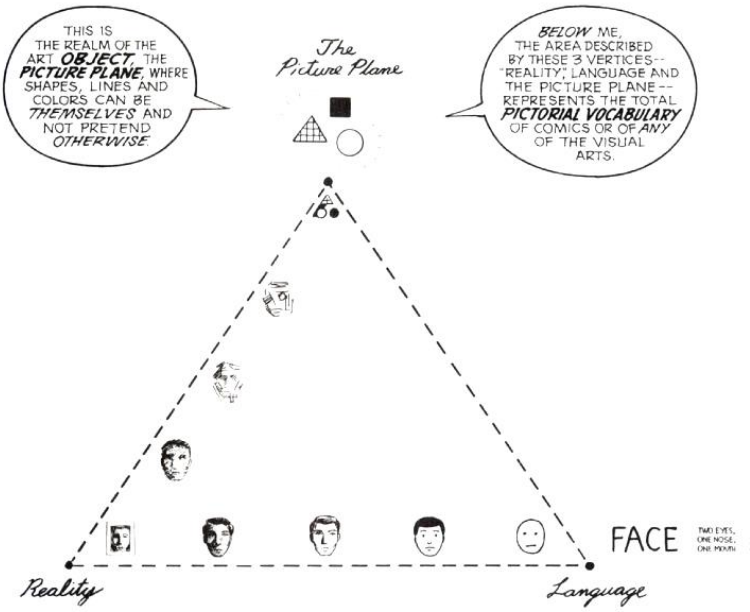

Figure 3 The Picture Plane, Scott McCloud

One end of the spectrum for metaphor is realism, which is placed in the bottom left of the diagram. A photo or a photorealistic image of a face would fall under this category and it tries to resemble the real thing as much as it can. On the other hand, you have

the more iconic images, like in a cartoon, which are focused on conveying meaning to the viewer. This can be found in the bottom right corner, where the face is reduced to the most important pieces of information: Two eyes, one nose and one mouth. McCloud calls this dimension of the diagram “levels of iconic abstraction” (McCloud 1994), because the images are being abstracted from the actual source but try to convey the original meaning. The maximum level of iconic abstraction would be written language, as this reduces the original image to pure meaning, but doesn’t look anything like it. The second dimension is pure abstraction, which is different to the previous iconic abstraction. He calls the point of extreme abstraction “The Picture Plane”, which consists purely of shapes and colors. When all three of these extremes are connected, you have a diagram of “every possible form of visual art and communication” (McCloud 1994). This can be applied perfectly to games as well. *Mario Kart* would be located more on the bottom right side, but not completely in the corner. The *Gran Turismo* games are trying to get as close to the realistic side of the diagram as they can. A game that could be categorized at the top point of pure abstraction is *Geometry Wars*. All these games stick to the metaphor they are trying to convey, which is one of the reasons they are so successful.

2.2.5. Rules

The last metric on this list is Rules. Swink defines rules as “an arbitrary, designed relationship between parameters or objects in a game” (Swink 2008). This means that every relationship between any object or parameters, that has no apparent reason for it to be the way it is, is considered a rule. In *Super Mario 64* you have to collect 100 coins to receive a star and in the *FIFA* Series you must place the ball inside the opponent’s goal to score a point. As you can see this applies to real life games as well and one could argue that rules are the most important thing about games. Think about the difference between Rugby and American Football for a minute and you will notice that the only thing that separates the two are the rules, but only one has millions of international viewers each year. Everything from the shape of the ball to what you can do with it, is a rule. When you look at the rules by themselves, they seem to make no sense at all, because they were ripped out of the context of the game. But they are deeply connected to the core of the game, which is why they are so important.

Rules can be again divided into three different types: high, medium and low-level rules. High level rules are implemented by the designer to guide the players towards a certain style of playing the game or to using certain mechanics. It is the designer telling the player “Hey, you should play it like this, because it is the most fun that way!”. Most often this is done through rewards for certain actions, by which the players are manipulated into playing a certain way. By saying that collecting 100 coins will give you a star, the

player immediately recognizes them as being important. Since coins lead to stars and stars lead to unlocking doors and new areas, which ultimately enable the player to beat the game, this rule influences the overall style of playing the game.

Medium level rules are directly linked to the action in the game. In *Hyper Light Drifter* you become hyper aware of the things and enemies around you as soon as you reach a critical health level. All you are trying to do is to get away from the enemies to use a health pack. This happens all because of an arbitrary rule that says you can only get hit four times before you die. It has an immediate effect on the way you play the game.

The lowest level of rules are how many hits it takes for an enemy to die or how fast you can fire your weapon. One game series that is known for the use of its rules is the *Dark Souls* series. Each enemy takes multiple hits and the bosses take even more. Sometimes a boss fight can last more than 10 minutes, which is quite a long time for just one enemy. By doing this the player immediately recognizes the importance of the fight, when he encounters one of the bosses. Once you have upgraded your character far enough, some of the enemies only take one hit to defeat them. This feels satisfying, because the player can now breeze through a horde of enemies, which previously would have taken him multiple tries to get through. The same applies to the way these games handle the different attacks, which the player can choose to perform. A light attack feels very different to the player than a heavy attack, which is emphasized by the rules. A heavy attack feels very weighty because of the animations and timing of it, but it wouldn't feel like a hard-hitting attack if it would do less damage than a regular attack. All these rules are arbitrary decisions, which were made by the designers to enhance the players experience.

2.3. Classifying Game Feel

After this last section it must be acknowledged that game feel is not the be all and end all of game design. It is just one method, which can be used by game designers to tell their story. At the center of each game has to be a game that is fun at a very basic level. You can try to hide the fact that the game is boring behind stunning visuals and effects, but it won't be the cure for bad gameplay or bad game mechanics. So where does it fit within this big system, which we call game design?

The paper "MDA: A Formal Approach to Game Design and Game Research" tries to break down the process of designing games and how they are consumed into smaller, more understandable components. They claim that the fun in a videogame is based on the systems of the game, which are again based on rules. By translating them to their design counterparts, which are aesthetics, dynamics and mechanics, they were able to describe the fun in games in a formal way. They also specify that the designer and player

both have their own perspective on these three components. A designer would first look at the game's mechanics, while the aesthetics are the first thing the player has contact to. According to this paper, one part of the aesthetics of a game is "Sensation: Game as sense-pleasure", which seems to fit the description of game feel. Unfortunately, there is nothing that further specifies what is meant by this term, so again we are left with an incomplete definition. This shows however that game feel or "Sensation" is part of a bigger picture, located somewhere in the realm of aesthetics. (Hunicke, LeBlanc and Zubek 2004)

Another more formal approach to analyzing games was made by Michael Nitsche in his book about "Video Game Spaces" (Nitsche 2008). He proposes a dissection into five "main conceptual planes" for analyzing video games. These planes are defined as follows:

1. *rule-based space* as defined by the mathematical rules that set, for example, physics, sounds, AI, and game-level architecture
2. *mediated space* as defined by the presentation, which is the space of the image plane and the use of this image including the cinematic form of presentation
3. *fictional space* that lives in the imagination, in other words, the space "imagined" by players from their comprehension of the available images
4. *play space*, meaning space of the play, which includes the player and the video game hardware
5. *social space* defined by interaction with others, meaning the game space of other players affected (e.g., in a multiplayer title).

When looking at this definition it becomes very clear that game feel plays or rather should play a pretty big role in game design, since it is directly linked to four out of these five planes. The rule-based space aligns perfectly with the Rules metric defined by Swink. Polish and Metaphor are what defines the mediated space and contribute to a big part of the fictional space as well. The play space can be seen as a combination of the Input and Response metrics. Of course, with each one not completely aligning with Swinks definitions, but the connection cannot get any more obvious. And yet we still have barely any knowledge about this topic.

Even though game feel seems to have influence on a lot of parts of the whole experience of playing a game, it doesn't really seem to be connected to one particular topic. At least not if you strictly look at the definitions. I'm talking about the holy grail of game design: Flow. Up to this point I have tried to not talk about Flow, but now that I am looking into the placement of game feel within the big topic of game design, there really is no way around this one. As a quick reminder, Flow is the "feeling of complete and energized

focus in an activity, with a high level of enjoyment and fulfillment" (Chen 2006) and was defined and researched by the psychologist Mihály Csíkszentmihályi. For a person to experience flow the activity must perfectly balance the challenge of the activity and the skill level of the person. If the challenge is too high we feel stressed out and become anxious, but we get bored, if the challenge is not high enough. The characteristics of Flow, for example losing the sense of time, being able to focus on the activity completely and forgetting all the problems and troubles of life, are also the characteristics of fun. (Chen 2006)

It seems that Flow plays an integral part of having fun and is therefore very important in game design. But this is also the reason why I would say that it has nothing to do with game feel. At least if you look strictly at the definition, it is obvious that game feel does not influence the balance of challenge and skill at all and therefore the two subjects should not be connected to one another. Of course there are some more factors that play into being in a flow state but those are the most important ones. Looking strictly at the definitions there doesn't seem to be a direct correlation between these two phenomena apart from having to have real-time direct feedback. But it still seems like game feel can influence the state of flow or at the very least can trigger the same sensations as flow. If there is a disconnect between the feel of an object and what it is supposed to be, we notice it immediately, and in some cases this can be very distracting from the task at hand. On the other hand, if it feels really nice and intuitive to control a character we immediately get soaked up by it. When the controls are satisfying we can quickly loose ourselves in the game and forget about the time and everything around us. You can spend hours in *Burnout Paradise* just racing through the streets and smashing billboards, without ever actually participating in any of the challenges given by the game. This is what is sometimes referred to as "being in the zone" and it is a very common thing to see amongst gamers. We already know that games and flow are deeply connected to one another, but I think a big part of why that is has been mostly overlooked. The feel of a game seems to be able to change the state of flow and trigger the same behavior in the mind of the player. There are some striking similarities between the two, but we know almost nothing about the connection of these two subjects. It is not my intention to give answers to this particular question. Since there is barely any knowledge about this specific topic and I did not have the resources to take a closer look at this phenomenon, I think it is best if I don't say much about it. I will, however, come back to it later, when I am discussing the results of my own project, because the results of my own studies have shown some very interesting results concerning this topic.

The connection between gameplay and game feel is much more obvious. Even though gameplay is also a very hard to define term, it is usually used to describe the underlying simulation of a game. What you can do in a game and how the simulation is handling the interactions of different objects is gameplay. With this in mind, we can say that game feel is everything that enhances this basic simulation, the effects of which have been already covered in the previous section. It is so deeply connected to gameplay that it is often mistaken for it, which could be one of the reasons why so many people say that good gameplay is the most important aspect of a game. But really it is the combination of the two that has to be good for it to work, with gameplay being the foundation and skeleton of the game. Game feel is the flesh and skin, which makes the game juicy and aesthetically pleasing.

2.4. Methods of Creating Good Game Feel

For this next section I will take a quick look at two games that were able to use some of the tricks discussed earlier to create a very satisfying feel. The metrics defined by Swink will be used as guidelines and ways to measure the feel these games have. The first one to be analyzed is *Hyper Light Drifter* by Heart Machine. It has a very satisfying combat system, which we will be looking at in more detail. The second game we will be looking at is *Journey* by That Game Company. I chose this game because it is very unique in many ways and the feel of it greatly contributes to this.

Hyper Light Drifter is a game that is often compared to the early *The Legend of Zelda* games and to the *Dark Souls* series. The combat is very fast and unforgiving. Exploring the world creates a sense of adventure and loneliness. Sometimes the game can even seem quite depressing with its visual story telling. Overall, I think it is one of the better games of this decade, which is due to the phenomenal visuals, outstanding sound design and of course the very crisp combat. The game world is set in this futuristic, almost dystopian time, which is emphasized by the visuals in the game at every corner. There are old ruins of the remnants of an older world, overgrown by trees and bushes. The main hub of the game is a small city with very old buildings, which surprisingly have neon signs hanging above their doors. Even in this city, the buildings look like they were destroyed by the forces of nature. They are trying to give the player the feeling of mystique and hopelessness. There is a lot that could be said about this game, but I want to focus on the combat, since this is the part in which this game really excels. Your main weapon is a sword, which has a basic attack and multiple different upgrades which can be purchased throughout the game. The secondary weapon is a gun with which enemies can be dealt with from a distance, but it can also be used to solve puzzles. A very quick dash is given to the player as a way to dodge enemy attacks and to leap across gaps in the floor.

Now let's take a closer look at this combat system and why it feels so incredibly satisfying to slash through enemies. The first thing that is obvious are the incredibly fast attack times. It is possible to deal three strikes in just over a second and you have the ability to change the direction with each blow. This is emphasized by the animations as the second frame of the slashing animation already shows the full sword swing. This creates the very snappy feeling in the combat. The following frames basically just show the character in the same position as he was in the second frame, which emphasizes the power behind the attacks.

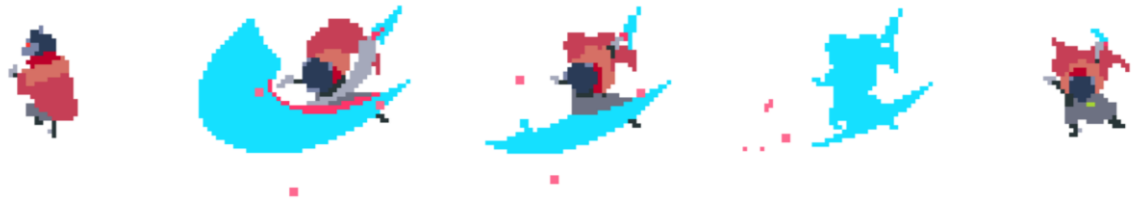

Figure 4 Attack Animation, Hyper Light Drifter

What's also interesting is that they have put in a frame where the whole character is blue. This is an effect usually seen on enemies when they are being hit, but I cannot recall a game doing this for the main character during the attack animation. It works really well for the feel of the attack, as it communicates clearly that the attack is very powerful. Even though the animation is very short it is still able to clearly show the player what is happening and it is doing so in a very elegant and satisfying way. On top of the visuals are the impeccable sound effects, which accompany each and every thing in *Hyper Light Drifter*. For the slashing of the sword they have used a variety of different sounds layered on top of each other, which accumulates to a high pitched swooshing sound. It feels very nimble but still powerful at the same time. Since it is a blue glowing sword they added something to the sound that makes it fit perfectly to the futuristic vibe of the game. The sound designer, Akash Thakkar, gave a talk about the sound design of *Hyper Light Drifter*, which was very interesting and insightful. With the animation, the quick response time and the sound effects in place it is already very satisfying to swing the sword through thin air, even when there aren't even enemies around. But the really good stuff happens when you have something that you can use these moves on. When an enemy is hit, they spray blood everywhere and the whole screen starts to shake violently. They also applied the same trick, where the character is painted entirely blue, to the enemies when they are hit. On the final blow the enemies are sent flying and hit the floor, where they are going to stay permanently. If the character is hit by an enemy, he flashes white and is stunned for a couple frames. On top of that they added a huge amount of particle effects, in the form of tiny glowing sparks, that play whenever something is hit. All this accumulates to this very satisfying feel, which makes the

character and the enemies feel like living beings that actually have weight to them. This is only the tip of the iceberg though, as there are many different attack and enemy types, which all have their own way of communicating their characteristics to the player. Going over each and every one of them would take a whole lot longer than I can afford in this thesis, which is why I will leave it at that.

In contrast to the very fast and snappy *Hyper Light Drifter* I have decided to take a closer look at *Journey* by That Game Company. It is a very short game and takes about two hours to complete. Most of the time in *Journey* is spent walking around the very linear levels. You play as an unknown figure dressed in red robes, who is able to glide through the air. The main goal of the game is to reach the summit of a holy mountain, which can be first seen from a distance, glooming over the desert. A lot of the world is focused around the red cloth in which the character is dressed. Almost all of the creatures you meet along the journey are completely made of this fabric and are unlike anything we have in our own world. What's fascinating about this particular game is that even though there really isn't a lot to do, it still feels very satisfying to simply maneuver around the world. That Game Company have done a fantastic job at creating a very unique sensation when walking and flying through the air, which is what I will be focusing on.

Figure 5 Journey, That Game Company

The first hour of the game is spent walking through a desert, where you encounter several buildings and ruins. It seems like there is a lost civilization hidden underneath the sand. Apart from collecting glowing artifacts, which enhance your ability to fly, there is nothing to do. You simply walk around and discover the world at your own pace. But this is the charm of the game. The walking feels really good and the flying is even better. You can almost feel the warm sand underneath your feet. Flying feels almost weightless and you can feel the wind flowing through the garments of your character. Obviously, the incredible visuals of the game play a huge role in communicating the feeling of walking through a barren desert, but they have added a few details to these mechanics,

which really hammer home this effect. Let us first take a closer look at the walking mechanic.

You control the character by pushing the left thumbstick in the direction you want to walk. Even though it feels very responsive and almost instantaneous, the movement of the character is very smooth and slowly accelerates in the direction you want him to go. This creates a very loose feeling, but at the same time gives the character some weight. All of the animations are very soft and there are almost no sudden movements. Since the motion of the feet is perfectly aligned with the speed of the character, it actually feels like the feet are connecting with the sand and get buried a tiny bit. The robe of the character flaps gently in the wind and gets kicked up by the feet of the character, which adds a nice physicality to the cloth. One of the biggest achievements of this game is the way they were able to make you actually feel like you are walking on sand. As you can see in Figure 5, the sand shimmers in the bright light of the sun, which is actually not very realistic but somehow it still fits perfectly. It looks almost as though it was a sea of sand, with the slight undulations. When walking around, the sand deforms underneath the feet and leaves a satisfying trail. John Edwards, the lead programmer for *Journey*, gave a great talk about how they were able to achieve this look at GDC. Moreover, they added some particle effects of sand being kicked up by the feet, which also slightly sparkles in the light. When walking down the side of a dune the character starts sliding, as if he had some kind of sandboard underneath his feet. The movement suddenly changes and becomes even more floaty than before, but at the same time you can feel how the feet have to push against the sand. This creates a very cohesive feel, which fits perfectly to the atmosphere they were trying to create.

One of the most satisfying bits of the game is the flying mechanic. As I had already mentioned previously, the player can collect glowing artifacts, which enhance the characters ability to fly. This ability is represented by a scarf, which is attached to the robe. The more artifacts collected, the longer the scarf becomes and the longer the character is able to fly. When the ability is “fully charged” the patterns on the scarf glow in a bright white, which depletes when the player is pressing the jump button. This is a very neat way of hiding a UI element inside the game’s world. The scarf is an actual physical object and is simulated in a semi-realistic way. It gets dragged around by the movements of the character, which makes it float through the air as if it wasn’t affected by gravity. Wind can also affect the movement of the scarf, which can lead to the scarf being wrapped around the character. When the player presses the jump button the character flaps his arms underneath the robe, which is reminiscent of a bird’s wings, and all the patterns on the robe and the scarf begin to glow in a bright white. It almost seems magical. As long as the player holds the jump button and the scarf is not depleted, the

character keeps climbing at a rather slow speed. Through this you can clear some pretty big gaps, even though the game never really forces you to do this. While flying the character leaves a trail of glowing sparkles, which slowly fade out. He continues to swing his arms and does all kinds of swirls and flips in mid-air. Once the player releases the jump button the character stops flapping his “wings” and starts to glide effortlessly to the ground. This also feels like the character almost wasn’t affected by gravity any more. While in the air the movement and direction changes are even slower or rather smoother than on the ground. All the animations make it look effortless and fluid. The robe and scarf slowly flutter in the wind. When the character lands back on the ground he does a quick cartwheel, but even this feels very graceful. You can still feel the impact of landing but since the movement keeps its momentum it makes it seem very fluid. Everything I have mentioned is emphasized by the minimalistic sound design of *Journey*. Howard Shore, the composer of the soundtrack, has created some very special pieces for this game. They allowed the game to breathe and gave the sounds of the world some time to unfold. There are many quiet moments in *Journey*, where the music almost stops completely so that you can only hear the sounds of the world. You can listen to the wind blowing through the sand dunes and hear every footstep. The sound of thousands of grains of sand hitting the floor and the cloth flowing in the wind. Every swing of the characters wings, is accompanied by some very light sound effects of cloth flying through the air. All those things combine to a very visceral sensation of walking in the sand and floating through the air, which is very hard to pull off if you still want the game to be responsive at the same time. That Game Company have done a great job at combining multiple aspects of game design to create this unique feel.

Both games have a very unique way of communicating to the player. *Hyper Light Drifter* is very fast and precise, while focusing on making the player feel powerful and heroic. *Journey* on the other hand tries to create this very serene atmosphere, which is reflected in the movement, animation and sound design. Even though both games require the player to walk long distances it feels very different in both games. The soft and fluid motions in *Journey* are very immersive and lets the players mind wander off into the distance, while the snappy movements and possibility of danger around every corner in *Hyper Light Drifter* makes the player extremely focused on everything that is happening on the screen. Both games succeeded in their own goals, because they had a clear vision in mind of what the gameplay should feel like, and they executed it masterfully.

3. Project

We have already looked at the various definitions of game feel, the problems of those definitions, different ways of measuring game feel, and we have also taken a closer look at two games. Now it is time to apply the knowledge of the previous sections to an actual project. For this I have built a really simple 2D-Platforming game where you can shoot enemies with a gun. It is not meant to be the best game of the year, with the best mechanics and the best level design, but rather a simple playground to test various features. The main mechanics of the game are walking/running, jumping and shooting. You have to traverse a rather simple level where you can fall from various platforms, which are filled with some basic enemies. All the enemies do is walk from one side to the other, turn around and repeat that process. If you touch an enemy or fall between the platforms you have to start the level again. To finish the level, you have to collect two coins which are placed somewhere in the level. This isn't exactly a masterpiece of game design, but it is good enough to test how good it feels to play.

3.1. Goals

The goal is not to build an amazing game but rather a really simple playground, where you can test and feel all the mechanics implemented later on. You don't get a fancy double jump or wall jump. Neither do you get any possibility to change your weapon. You can't even make actual progress as the level can be beaten in about 30 seconds and that's all there is. But that is good because we don't want to test how good the game itself is but rather how good it feels. Anything that could possibly distract from that or create a false sense of better feel was left out. This was very important because it is already a very confusing and ambiguous topic when testing. By leaving out anything that could distract from the game's feel I tried to minimize the effect of the gameplay itself on the outcomes of the test. To emphasize this, I made the testers play the same basic level six times. With every new playthrough I added some mechanics to the game. By not changing the game itself, but only adding on top of the existing experience, the players can focus on how the game actually feels. I will from now on refer to each playthrough as a stage.

3.2. Unity

Using the Unity Game Engine was an obvious choice for this project. Most of my previous projects were made in Unity and it is the easiest to use for projects like this. Also, since the goal wasn't a realistic look or any unusual simulations but just a very simple 2D-Platformer, there was no need for anything else. Unity's Editor made changing the details of each stage very easy, while still being very flexible. Since Unity

has a built in Physics Engine it was very easy to handle gravity and all the collisions. It also made it very easy to add little particle effects, which could interact with the world.

3.3. Baseline

Why did I choose to create a 2D-Platformer instead of a Racing game or a First-Person-Shooter? The most obvious reason for this is the simplicity of the simulation. It is very easy to create a 2D-Platformer with the basic functionality, whereas a racing game would need a more sophisticated simulation to be acceptable to begin with. Also, the platformer-genre is very well known and almost everyone seems to enjoy them, but not everyone likes First-Person-Shooters. This could potentially be another factor influencing the results of the tests. When the person simply doesn't like game's genre then the changes to the game won't affect the game's feel as much. Therefore, I chose to stick with a classic arcade platformer since this is something almost every kid grew up with and knows the basic controls of.

In this section I will be discussing the baseline of this project in detail. I will explain what exactly I started with and why I chose to do so.

3.3.1. Controls

The game is played on a regular PlayStation 4 Controller. The gamepad sits fairly comfortable in the hands and almost everyone is familiar with the layout of the buttons. This minimized the effect of the players first having to learn the control scheme of the game and getting used to the controller. For the inputs I used the standard platformer game controls. Running or moving left and right was mapped to the left thumbstick. Since the thumbsticks on the PlayStation controllers send a continuous signal ranging from -1 to 1 on each axis, this already feels more expressive than using the D-pad. For jumping I used the X-Button, since this is the standard jump button in almost every game. Since there is no need to give more precise input than "Jump" and "Don't Jump" the binary input of this button was sufficient.

The one thing that doesn't have a standard mapping is shooting. Depending on genre this can be put anywhere on the gamepad. Most FPS titles put shooting on the triggers of the controller (Xbox: LT/RT, PlayStation: L2/R2), whereas some more arcade like games put it on some of the front-facing buttons. I decided to map it to the R1 Button of the PlayStation 4 controller, because the player only needs two options. Either you want to shoot or not. There is nothing in between, which could benefit from having a continuous input from 0 to 1. But I still wanted to make the players use their index finger to shoot as this gives the players the sensation of actually pulling the trigger on a real gun. Even though the game is not about realism, it still feels better. With shooting in place all of the controls are set up and the player can control the game.

3.3.2. Physics

During the introduction to this section I already stated that the players start by controlling a simple rectangular box. This is exactly what the simulation of the game is actually doing. The player character is a rectangular box, with a collision box exactly the size of what the players can see. To handle all the physics and collisions I used Unity's built in physics-system. This means that the player object is a rectangular box, with a rectangular collision box and a rigidbody attached to it. The same basic structure is applied to the enemy characters, even though they are a bit smaller than the player character. The Z-Rotation of the rigidbodies is locked since any rotation would be undesirable. The characters should never roll onto their sides. Their feet should always be planted on the floor, or at least point downwards.

Gravity is applied every frame at a constant rate, but I used a different value than the actual 9.81 m/s^2. Even though this value would be accurate it makes the object feel really floaty. This could have been used as the starting point for the tests, but the difference would have been too jarring. So instead of the real-world value I used the value of 20 m/s^2. Even with a value almost double the real value it still doesn't feel very good since the character jumps almost three times his own height. This is something very unusual. Using the actual value of gravity seems really odd in a videogame. It feels as though you are playing a character on the moon even though this is the same acceleration we experience in our everyday life. This effect is enhanced by the fact that most game characters can jump much higher than any human being ever could. If you look at Mario's Jump in any of the *Super Mario* games you will notice that he jumps incredibly high, sometimes even reaching five times his own height. We are accustomed to game characters jumping very high, since it wouldn't be fun if you could only jump half your own size. To counteract this, we usually use a very high Gravity factor, so that we don't experience this feeling of floating through the air.

3.3.3. Running

The horizontal input of the left thumbstick is directly mapped to the velocity value of the player character. There is no acceleration applied, but since the players are using the Left Thumbstick of the PlayStation 4 controller, there is already some smoothing of inputs happening. Even though the players don't notice that their inputs are not instantaneous, it still has some effect on the feeling of movement. It is a very small impact but an impact nonetheless. Having no acceleration means that the players can instantly change direction without having to worry about sliding off any platforms. But it can also feel a bit "twitchy", since this is something we never see in the real world. There is nothing that we know of, which can actually change direction in an instant. The maximum walking / running speed of the character is 5 m/s, or 18 km/h, which seems

like a really high value especially when you consider that the character has a total height of about 80 cm. Considering that the average running speed of humans is 9 km/h, this is a very high number, but when playing the game it feels as though the character is having a stroll down the beach. In other words: The character is still rather slow.

3.3.4. Jumping

As I have already stated in the Input section, the X-Button of the PlayStation 4 Controller is mapped to jumping. Since this is a simple binary input device the feedback to pressing the button is instantaneous. You don't have to cross a certain threshold to make the character jump, you just have to press the button. Once the button has been pressed the character jumps instantly. There is no delay and no wind up for the animations. To achieve this, I simply apply an upwards impulse to the character's rigidbody. The force value is 10, while the weight of the character is set to the default 1. These values have no actual meaning unless you give it one. I will not be doing so, since there is nothing gained by defining these values in real world terms. Once the character has left the ground he will be accelerating downwards at a constant rate until he hits the floor. While in the air the character is still able to move left and right in the same way as before. There is nothing gained or lost when jumping, at least not when looking at the movement itself. The acceleration is still instantaneous, and the maximum speed is the same at 5 m/s.

3.3.5. Shooting

For the shooting mechanic I have first thought about having the players aim in the direction they want to shoot. This would have had to be done by using the right thumbstick of the gamepad, but this would have been interfering with pressing the jump button. It could have led to problems, where the player wants to shoot in a specific direction and then cannot jump because his right thumb is already occupied by having to aim. It also would have been a rather unusual way to aim, since not many games have this kind of control scheme. Therefore, I decided to simply have the character aim in the direction he is currently running in, which is a pretty intuitive control scheme since most of the time the players want to shoot in that direction anyways. Another positive side effect of this is that the controls stay very simple. If I had added the second thumbstick I think most of the players would be pretty overwhelmed by the controls. This could have a very big impact on the games feel, which is another reason why I decided not to do that. One problem arises though, when the player has to run away from one of the enemies but at the same time wants to shoot him. There is no way to do so, which is why I decided to keep the direction the character is facing, as long as he is shooting. This gave the players the opportunity to walk backwards, while still firing their weapon.

Once the R1 Button has been pressed the weapon fires a small rectangle, which is about 1/8th of the size of the character. While being pretty big in comparison to the character it still looks rather small in the game and in turn makes the bullet seem a lot less dangerous. The bullets move at 10 m/s, which is ridiculously slow, when compared to a real bullet. On top of that the bullets are flying in a straight horizontal line, without ever deviating from this overly perfect path. Once they hit something the bullets simply disappear, without showing any signs of actually having an effect on the environment. If a bullet hits an enemy, it of course applies damage to the enemy, which is always exactly one health point, of which the enemies have four. The players are able to fire a maximum of two bullets per second, which again is pretty slow and feels really awkward when trying to make any progress. Since firing every bullet one by one could get really tiring for the fingers, I decided to allow auto firing, by simply holding down the R1 Button.

3.3.6. Enemies

As I already said earlier the enemies are made in the same way as the character is. They are simple rectangles which are a bit smaller than the one of the character. The movement pattern of them is very predictable since all they do is walk from end to end of the platforms. Once they have reached the end of the platform they turn around and instantly have the same velocity as they had before. They are moving at a constant rate of 3 m/s, which makes them a fair bit slower than the character. If the character touches an enemy the level is restarted instantly without any animations or effects. It is simply a hard cut back to the beginning of the level.

3.3.7. Winning Conditions

Since the rules of the game have an effect on the games feel I decided to go for one of the simplest forms of giving the players a goal. I had them collect two coins, which were at some fixed location in the level. All they had to do to finish the level was to collect these coins. To make it a little bit more interesting than that, I decided to add a timer, so that they had to finish the level before the timer runs out. If the timer finishes before they are able to collect all the coins, the level restarts from the beginning.

3.4. User Study

The baseline for the whole project is now finished. It is a really simple 2D Platformer, where the players experience nothing but pure gameplay.

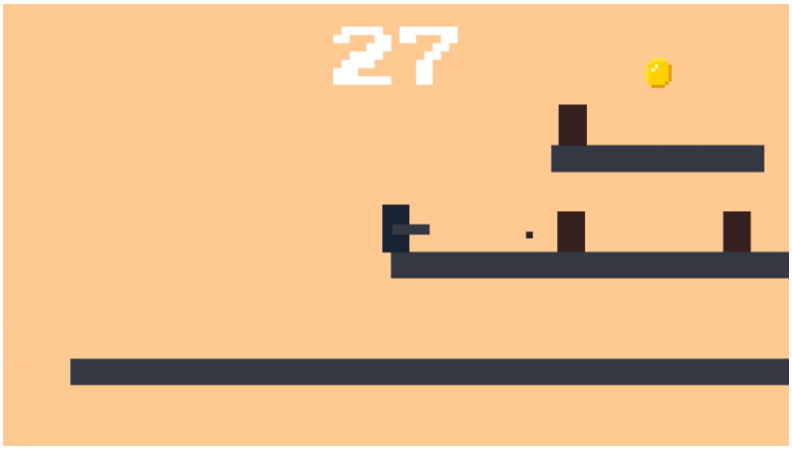

Figure 6 Screenshot of the 1st Stage

As you can see in Figure 6 it really is the simplest form of a 2D Platformer. The characters are rectangles, the bullets are rectangles, and the platforms are rectangles. Since the previous section has been all about how and why this baseline was created, I won't go over it again here. Just remember that this is what is actually happening underneath all of the changes coming in this section. I will now go over all the changes and improvements I have added for the first user study to make this game feel better.

3.4.1. Second Stage

Right now the players have no idea what they are actually controlling. It could be a human being with a gun, or it could be a bear that fires acorns from a tree trunk. There is no context to what you are doing when playing the game. This is a very important aspect of controlling a virtual object. If you give the players some visual feedback in the form of context it immediately sets the tone for all of the things that are happening. Playing the two different versions of the game I have just described would feel a lot different, since the player would have different expectations about how everything should look, sound and feel. This is why the most obvious improvement anyone would probably come up with is to add some sprites and animations.

Figure 7 Screenshot of the 2nd Stage

Since I am not a particularly talented artist I decided to go for the classic pixel art look, which sets the expectations for the game to be more arcade like and that it doesn't try to be realistic in any way.

I added a background image of a futuristic looking city at sunset. This created a generally warm and lighthearted feeling, which is good since I didn't want to create a feeling of unease in the players. Since the city should act as a backdrop, I added a slight parallax effect. This gives the impression of the city being further in the back, but it also places the level more into the world. The effect of that on the game's feel is probably rather small but it could still be interesting to look at this in more detail. The character has received a big overhaul and is now a small robot with a gun. I also added the animations so that when the character is running, he actually moves his feet and turns his body slightly. To emphasize the motion, I made him bop up and down when running. This gave the character a nice friction with his surroundings and it didn't seem as though he was sliding across the floor anymore. The animation is played back at a rather fast speed so that the speed of his feet moving actually matches his horizontal movement. When jumping, the character now actually looks in the direction he is heading and is moving his feet in a way that would make sense in the real world. The gun looks like a small rifle and the bullets are now a lot bigger and glow white and yellow. The bullets are now about a quarter of the size of the character and are almost as big as his head. This makes the bullets feel a lot more powerful even though they stayed exactly the same, if you look at the simulation. To really sell the impression of a gun being fired I added a small white circle for the first couple of frames of the bullet, which looks like the gun has muzzle flash. For the enemies I created a similar sprite with animations, but this time I only needed the walk cycle since this is all they are able to do. They are now also small robots but in a dark red, which instantly shows the player that these are bad guys and you shouldn't touch them.

Visually the game already looks acceptable at first glance, but something every game needs is sound. I added some 8-bit sound effects, which I have created myself with a program called SFXR. With this I created sounds for walking, jumping, shooting, collecting coins, enemies being hit and enemies dying. All of these sounds are not really perfect, since I am not an audio engineer, but I think they get the job done. For the walking and jumping sounds I tried to emulate the sounds of classic arcade games. The shooting needed a little more substance to it and should feel powerful, which is why I gave that sound a little more bass. For the enemies I essentially had some noise at different levels. The sounds are rather short so that it actually sounds like something is being hit. The only sound I am really happy with is the one of collecting coins. It really reflects the delight of picking up a coin and gives you a smile on your face. It reminds me a lot of the old *Super Mario* games and simply works perfectly for what it tries to communicate to the player. The last thing I added was some background music. For this I used an 8-bit track by Komiku off the internet (Komiku 2018). It is fast and makes the player want to go fast, which is exactly what I wanted to achieve. It also works really well in combination with all of the other sound effects.

Overall the sound and visuals now show a kind of unison and both work in the same direction. Everything tries to give the player the feeling of a lighthearted arcade game. At the same time they give the player more context about what is actually happening on the screen.

3.4.2. Third Stage

The next level contained the only actual changes made to the underlying gameplay. They are all very simple and don't actually change what is happening it just gives the player more freedom of expression and gives everything a lot more physical weight. These changes are important to sell the impression of controlling a physical object and not something that floats across the floor.

Something that is very obvious is the changes made to the movement of the character. Until now the character was moving at a constant speed with the value straight from the input of the left thumbstick. It was a direct mapping with only a speed multiplier applied. Now I changed it so that the player would control the acceleration rather than actual velocity. So, when the player would press the thumbstick in one direction it would take the character approximately 150 ms to reach his top speed. This gives the impression of weight because the character first has to accelerate. Even though 150 ms is not a long time at all it is still very noticeable. The same thing would happen when the character wants to decelerate. While the character is in the air this value would be a little bit lower, so that it would now take him about 200 ms to reach his top speed. By doing this the players can feel a difference between running on the ground and moving in mid-

air, which is very important for the response of the game. By doing this, the game clearly communicates the two different states of the game. Unfortunately, the changes made to the acceleration create the side-effect of the character sliding a little bit. This introduces some unpredictability, which is not exactly ideal, but it is a good tradeoff, if you want to sell the impression of weight, at least in my opinion. There is no ultimately perfect way to balance this since everyone has a different opinion about what this acceleration value should be. On top of the changed acceleration the character is now running at a speed of 8 m/s. This is important, since before the character felt quite slow and sluggish. Giving him the ability to run a lot faster means giving the player more control. If the players have to wait for something to happen it can get very boring so the goal should be to always have them do something. Of course, this depends on the type of game and the context of the environment.

For jumping I implemented the same methods used in *Super Mario Bros.,* where the player can decide how high they want the character to jump. So, if they press the jump button really quickly the character would only hop a tiny bit, but if they keep pressing the button, he jumps a lot higher. Adding this creates a great feel when jumping since you are always in control of what the character is doing. Before I added this, you would commit to a jump all the way, even if you only wanted to jump a tiny bit. This means taking control away from the player which is almost always a bad idea. I achieved this by applying a higher gravity to the character of about three times the regular value. So, gravity is pulling him down at 60 m/s^2 as soon as the player releases the jump button. This is obviously not realistic, but it feels correct to the player. I only apply this value if the character is still travelling upwards, so most often it is a rather short period of time. In addition to giving the player more control in mid-air, I added a stronger gravity when the character is falling. For this I used a factor of 2.5, which results in an acceleration of about 50 m/s^2. This creates the feeling of the character really trying to reach the most height before plummeting down again. This is the reason for players pushing the jump button super hard in the *Super Mario* games just to reach a tiny bit higher. They have the impression that the character is actually struggling upwards, so maybe pressing the button harder gives him an extra boost. It also gives the character a lot more weight, since you can feel how gravity is pulling him down. Again, all those values would need a lot of testing to get exactly right, so I used the values from *Super Mario Bros.* and changed those slightly to suit the game.

Since all of the changes focused on changing the physical properties of the character, I decided to also add a small kickback to the gun. This means that when the player fires the weapon, I would apply a small force opposite to the direction of the shot. When standing on the floor this moves the character by only a couple of pixels. It almost isn't

noticeable, but it still has an effect on the overall feeling of firing the weapon. When in the air the effect is a lot stronger since there is no friction with the floor which could slow the character down. It gives the shot more weight, which results in the weapon feeling more powerful and more connected to the character. In the same way, I added a kickback to the enemies when they are hit with a bullet. The force for this is twice as big compared to the one applied to the character. This improves the feeling of the shots actually having an effect on the enemies. Before adding this the player had no way of telling if the bullets are damaging the enemies. To further improve the feel of the weapon I made the fire rate a lot higher. Whereas before the weapon could only fire 2 bullets per second it could now fire 10 per second. Of course, this has an effect on the gameplay as well, but it immediately feels more powerful. Adding to this I decreased the accuracy of the bullets and made them faster. They are now flying at a speed of 30 m/s and can have their direction changed by 5 degrees in each direction. This deviation from the regular flight path is done randomly, which further sells the impression of the weapon being powerful. The character seems as though he can barely control the weapon. It also makes the shots fired more unpredictable and therefore gives the eye something more interesting to look at.

All of the changes done in this step are centered around the physicality of the objects involved in this game. By giving the character acceleration and making the shots have an effect on the position it creates a sense of weight, which is essential for the feel of the game. They are all rather subtle changes, apart from the fire rate, but when combined they really sell the impression of weight.

3.4.3. Fourth Stage

For the fourth level I focused on getting the camera right. At the moment the camera is always centered on the character and never moves away from that. This creates a very stiff feeling when moving the character around the level. To improve this, I made the camera look in front of the character. The camera focuses on a spot 2 meters in front of the character which places him at a third of the horizontal width. Placing objects of interest at a third of the image is a rule used in photography and film, which makes the overall composition a little bit nicer. Having the camera look ahead of the character enables the player to see more clearly what they are trying to shoot. The player doesn't want to look at his own character, but he much rather wants to see where he is going. This makes the placement of the camera a little bit better, but it still feels really stiff, which is why I added smoothing to the camera's movement. This lets the camera glide gently between the different positions. This also means that the camera is lagging a little bit behind the character, making it a little bit harder to accurately follow the movement of the character. Getting this absolutely perfect requires a lot of testing but getting the

general feeling right can be done relatively easy. The speed at which the camera is moving is still pretty high but it's just slow enough to give it a smooth look.

Now moving around the level feels a little bit more natural and not so stiff any more, but we can improve the feeling of shooting enemies as well. The biggest and most obvious change here is the screenshake. Having the screen shake, when hitting something, is essential, if you want it to feel correct. To do this I had the camera shake a tiny bit whenever a bullet hit something and made the camera shake quite strongly when an enemy died. This gives the player great feedback about whether an enemy has died or not.

For the screenshake itself I used a method described by Squirrel Eiserloh in a Game Developers Conference talk (Eiserloh 2017). In his presentation he talked about various ways to improve the camera movement, including a better way to apply screenshake to your camera, instead of applying random offsets to the position of the camera every frame. Instead of having an event trigger a certain amount of shake he instead suggests using something he calls *Trauma*. This is a value which directly indicates the screenshake amount of the camera. Different events, for example when an enemy died, can then simply add a value to the existing Trauma. This Trauma level is clamped between zero and one and always decreases at a constant rate. Using this system means that when many things happen at the same time or quickly after one another the screen starts to shake a lot more, since all of the events add their value on top of the existing Trauma. This is a very natural way of giving the player feedback about the intensity of the situation. To make the shaking itself feel better he suggests using Perlin Noise instead of assigning a random value each frame. To use this, you have to use the current time as the seed for the Noise and add a constant value for each axis, because you need different values for the x and y axis. You then simply apply the offset of each axis to the position of the camera, but first you have to multiply the values with the Trauma level first. To create a more natural falloff he also suggests using a quadratic function so instead of simply multiplying the offsets with the Trauma level you first square it and then multiply the values. Eiserloh also suggest applying a rotation to the camera as well as simple offset. This greatly improves the feeling of natural shaking, since anything that shakes in the real world usually also has some random rotation. This is also a method used in cinema and can really improve the overall cohesion of what you are presenting to the viewer. Using this method creates a very natural looking screenshake. If you were to apply a random offset each frame the camera would jump around without any connection between the frames, but by using Perlin Noise you mitigate that because now the camera smoothly changes direction but it is still unpredictable. (Eiserloh 2017)

Applying screenshake is always a rather difficult task since you get used to it and start to think that you could apply a little bit more of it, even though the screen is already shaking uncontrollably. This is why it is very important to have unbiased persons to test this, since in this case the developer really is the worst judge. Now that this system is in place, I can just add a value to the Trauma to make the camera shake every time an enemy dies. This makes a huge difference to the weight of the bullets and gives the player the feeling of controlling a powerful weapon, since every shot really hits home.

Another small addition to the camera, or the game itself rather, is having the game freeze for just a split second every time an enemy dies. I got the idea for using this technique from a talk called “The Art Of Screenshake” by Jan Willem Nijman (Nijman 2013), but it is a very well-known method and is used extensively by games like *The Legend Of Zelda* and *God of War*. All you do to create a more satisfying impact is to have all the motion freeze for about 50 milliseconds. This is just enough for the brain to notice but it doesn’t interrupt the flow of the game and doesn’t break the real time control of the player.

For this stage that is all I have changed and even though it isn’t a lot and doesn’t have the biggest effect on the games feel, it would still be very obvious if these things were missing. Something would just feel a little off. The screenshake however is essential to a game of this kind and most action games use it very effectively, but there is always something that can be improved about the way you do it.

3.4.4. Fifth Stage

The last level was all about polishing the effects that were already in place. This is why this level has the most changes done to it but most of them are rather small and none of them affect the gameplay in any way. I will start with the most obvious change, which is that the world now has a lot more context to it. Whereas before the character was running on floating boxes with textures applied to them, he is now standing on top of rooftops of some skyscrapers. When the character falls down between the rooftops he can now actually hit the side of these buildings. I also added some props to the rooftops in the form of electrical boxes, antennas and wires. These give the player some more context about the surroundings and changes the feeling of being present in the world. It adds some more reference points for the movement, which ultimately creates a stronger feeling of being in control of the character.

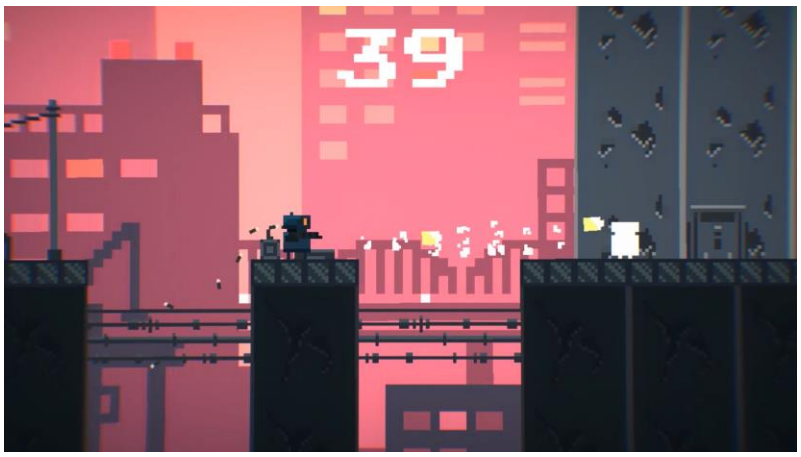

Figure 8 Screenshot of the 5th Stage

The second biggest change are all of the particle effects I added to the game. Let's start with the small dust particles kicked up by the feet of the character. Unity made the timing of this really easy, because for each animation, in this case the running animation, you can add trigger points which in turn trigger a function call. By linking this up to a function which spawns a puff of dust they would always be perfectly timed to the footsteps of the character. Having these in place really had another great influence on feeling the connection to the floor. So even though it essentially still is a box which is just sliding across the floor, the player now really feels every single step of the character, even though they are not actually happening at all. The weapon also received a small particle effect upgrade, in the form of small bullet shells flying out of the weapon. These are real objects with a rigidbody attached to them, which allows them to interact with the world. They spin uncontrollably through the air before hitting the floor. There is absolutely no use for them apart for the improved feeling of firing the weapon. It just adds this tiny bit of extra feedback which in turn makes it more satisfying to shoot. The gun now also has a small kickback when a shot is fired. It juts back and forth when shooting, which makes the weapon and the shots fired feel more powerful overall. It is a very subtle and quick effect, since it only takes about 50 ms to return to its original position. The bullets themselves now also leave a smoke trail behind them. This draws more attention to the bullets and therefore improves the feeling of power when shooting. On top of that it is really satisfying to watch them fly off in to the distance, since you can actually follow their path. For these particles I added a small random offset to the position and also applied a random rotation by a multiple of 90 degrees. This creates a more diverse look for each shot, so that it is always a little bit different with each bullet fired.

Hitting an enemy still felt kind of uninteresting and not very rewarding, since there is barely any feedback. Of course, they get knocked back a tiny bit, but this is not enough to provide instant feedback to the player about the actions happening on the screen. To counteract this, I made the enemies flash bright white for a split second. This is

something you can see very often in videogames, especially in 2D games like *Super Metroid* or *Hyper Light Drifter*. The whole sprite of the enemy is shown in full white for just 100 ms, which is a quick and easy way for instant feedback. Adding this to the game made shooting an enemy a lot more satisfying, since you can now tell right away that you are dealing some damage. Adding to that I made the enemies fly back and lie on the floor when they are dead, instead of simply disappearing into thin air. They received a very simple sprite where they are lying on the floor, which I also used for when they are still in the air. They would get knocked back by the last bullet shot and fly in an arc to the floor, where they would then lie until the level was restarted or finished. This adds a lot of physicality to the enemies since they don't simply disappear. Having them fly through the air and hit the floor emphasizes this fact. Letting them persist on the floor, without having them fade away or despawn, adds a permanence to the world which can create a nice satisfying feeling when looking back at the areas you have already been to. I added the same effect to the bullet shells of the weapon, which also never disappear and always stay lying on the floor.

To emphasize the characters death and to make failing a little bit more enjoyable for the player I made the character fly back in a similar way to the enemies. This also creates a sense of physicality, in the same way it does for the enemies. The player can see that he is controlling an actual object, which doesn't simply disappear, and therefore feels more connected to it. To emphasize this, I made the character let go of his weapon and had it spin, as if he had received a very strong kick. This of course is not realistic at all but, as I already said, the goal is to emphasize the effects, which are already in place, and not realism. To really exaggerate the impact, I added a really quick white flash to the camera. The whole screen would flash white for 50 milliseconds in the exact instant the character died, which gives the player instant feedback about what happened.

The final changes made to the game are some small tweaks to the camera and overall presentation. To start off, I added some slight color correction and a chromatic aberration effect to the camera. These are very subtle, but I think it's worth mentioning, since it changes the feel of the game, even though the effect is very small. Something a little more noticeable are the black fade in and outs, which I added to the start and end of the level. Whenever the level is restarted the screen would first slowly fade to black and then start fading back to the reloaded game. This effect takes about two seconds and doesn't even affect the game itself, but it still makes restarting a little more natural. Before this fading was added the game would simply cut to the beginning of the level, which could feel a bit jarring, since there is no warning to let you know it will restart. This is simply a little bit easier on the eyes and prepares the player for the next try.

3.4.5. Changes for Second User Study

With all these changes in place I was now able to do two separate user studies. The first one involves all the five stages in the exact same order. For the second user study I had the player decide what he wants to upgrade, by using the collected coins in the level to purchase specific upgrades. The most important aspect is that all the changes are modular and work independently of one another. This means that the player could choose to only apply the polish effects or only upgrade the visuals.

I separated all the changes into groups, which were *Physics, Visuals, Audio, Camera* and *Polish*. The category *Physics* included all the changes to the actual parameters of the game like the fire-rate, dynamic jumping or the changes to the movement. *Visuals* included everything from the animated sprites to the props. *Audio* is pretty self-explanatory. For the category *Camera* I chose all the changes which were made exclusively to the camera. For example, the screenshake and the camera looking ahead of the player would fall under this category. *Polish* in this instance was all the miscellaneous changes like permanence, particle effects and bullet shells. For each category I had three tiers of upgrades, except for the *Audio* category, which only had one tier for sound-effects and one for the music.

The cost of the first tier of each category is only one coin and is increased by one with each purchase of this category. For example, to get the third tier of upgrades in the *Physics* category the player would have to spend three coins. Since I let the players play the same level five times, it meant that they were able to purchase upgrades four times. To make them actually choose between different upgrades I only placed five coins in the level. This way they would never be able to purchase every upgrade and they would have to decide, which category was most important to them.

4. Evaluation of Data

4.1. First User Study

As I have already mentioned previously the first user study was structured in a way that I would decide which upgrades the player was going to experience and in which order. The five stages of the previous section are the same five stages, which the player had to complete. Each stage would add the upgrades on top of the previous one. After they had played through the last stage of the game, I had them play the first one again, to see if they would give a different answer after they have seen the final result. The goal of this user study was to mainly show that these upgrades indeed change the way a player experiences the game. Even though I did not try to show, which changes to the game would affect certain psychological aspects the most, the results still showed some indication as to what works and what doesn't.

For this first test I had 10 students play the game, which ultimately is not enough to actually deduce a lot of information, but it is enough to show that we can actually test and measure game feel. Doing more tests would certainly take a lot more time. But I still wanted to get the most out of these tests, which is why I watched the players play the game and could therefore say a bit more about the experience the players had. This is something which is often done during the testing phases of game development, since this is one of the best ways to actually see what works and what doesn't. One observation I was able to make was that most players would start to engage a lot more with the game, the further they got into the stages. Whereas in the first stage their faces were almost emotionless, they started to smile and jump in their seats when something exciting would happen. Some would even scream at certain occasions, when they had died just before reaching the end of the level. It really did seem like they were more and more immersed in the game, even though they were still basically playing the same game. Although this cannot be used as scientific data it was still very insightful to me and simply showed me that I did something right.

The actual core of the user study was the survey, which each player had to fill out. For the first study I used a slightly shorter version of the one I had prepared for the second user study. This was simply due to time constraints. With the short version a survey would take about 20 minutes and since I wanted to find at least 10 testers I had to use this version. Otherwise it would have been nearly impossible to find people that were willing to do a 40-minute-long survey. The questionnaire consisted of four different questions, which were repeated for each stage, plus an extra page with some basic information about the user. These general questions about the user included things like "How often do you play videogames?", "What do you consider the most important part

in a videogame?" or their age and gender. In hindsight these questions were not very useful as most would give almost the exact same answers and I had not enough data to show some connections to their experience with the game. The four questions, which were directly linked to the game experience, proved to be quite useful. The questions were as follows:

1. How powerful did the weapon feel?
2. How good did it feel to control the character?
3. Describe the games feel in one word
4. How good did the game feel overall?

For each of these questions I gave them the option to assign a value between 0 (very bad) and 10 (very good). For the third question I gave them five suggestions, which they could choose, but I also gave them the opportunity to write down a word of their choice. The suggestions I gave them were *clumsy, responsive, stiff, smooth* and *tight*. What's interesting about this particular question is that across all 10 tests, the users only wrote down something of their own choice 14 out of 60 times. This is not even 25% of the total. This shows that it is indeed very hard for us to think of words, when we want to describe the sensation of playing videogames. Looking for better and especially more precise terminology has to be one of the main concerns, if we really want to tackle game feel. Something, which again shows how incapable we are at expressing sensations like playing a videogame, is that most people would really struggle to choose a word or even write one on their own. Nobody used words like satisfying, weighty or juicy, which are often used when describing a game's feel. Of course, this could have many reasons, but I have the suspicion that, if I had not given them these suggestions, most of them would have resorted to words like good and bad.

Even though communication can be a barrier I was still able to find some interesting patterns with this question. For the first stage the two words, which popped up the most were, *responsive* and *stiff*. This aligns with the findings of Steve Swink, as the attack time of the movement in the first stage was extremely short. The character would reach his maximum velocity as soon as the player pushes the stick in one direction. This leads to the game feeling stiff, but on the other hand it is very responsive, as the game reacts to the inputs in almost no time. Comparing this to the second stage, where nothing apart from the visuals changed, is very interesting. Even though the underlying physics of the game stayed exactly the same, a lot of players now changed their answer to the game feeling smooth instead of stiff. For the stages three and four I really couldn't find anything noteworthy about the data because everyone seems to have different thoughts about the game. The answers range from stiff to smooth and from clumsy to slippery, which made it seem almost random. For the fifth stage the results were more conclusive as

most of them described the games feel as either smooth or responsive. This can be an indicator for the game being in a state where most of the things fit together. The players expectations are aligned with what they are experiencing and it feels good to play the game.

The three other questions were a lot more precise, which is why most of the users had no trouble answering these. It is interesting to note, however, that most often they would look at their previous answers to give a more consistent answer. To show that the changes in the game affect the feel of the game I have averaged the answers of all the users and plotted them in a graph.

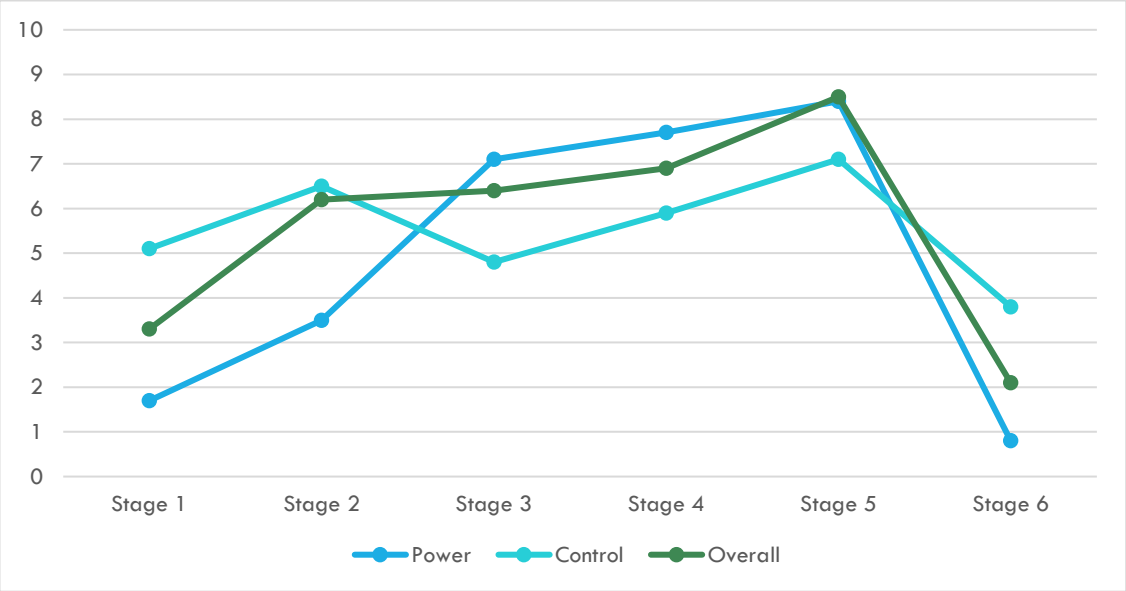

Table 1 Results of 1st User Study

As you can clearly see, there is a steady increase for all three values until a very sudden drop-off at the end. The biggest increase of the overall feel happens from the first to the second stage. The value jumps from 3.3 to 6.2, which is very fascinating. I did not expect to see such a big increase of the overall feel, since I only added some sprites and sound effects on top of the basic simulation. No values actually changed, but the animations and sounds tricked the players into thinking that the game feels better. It even influences how powerful the weapon feels even though it is still the exact same thing. The same thing happened for the control value. These results can be a bit misleading, as the visual change from the first to the second stage is extremely obvious and can be noticed immediately, which might have led to an exaggerated result. But it is still interesting to see that some very basic animations and sound effects can go such a long way.

The sudden increase of power from the second to the third stage was quite predictable, since this is where the rate of fire increased dramatically. The control value drops a lot which shows that the values I chose for the acceleration and deceleration are not very satisfying. This also aligns with their description of the games feel, where six players described the feel as *slippery* or *clumsy*. Most players also had some trouble adjusting to the dynamic jump height, which depended on how long they pressed the jump button. Most players would get the hang of this quite quickly though. For the stages four and five all three values gained around one point with each stage. The fifth stage is where the combined value is at its maximum. All three different values are higher than in any other stage, even the control value. This again shows that the changes to the camera and the overall polish and particle effects really helped in making the game feel better. In the fifth stage I added a lot of props, which give the player a lot more reference points to which he can compare his current speed. This might be one of the reasons, why they thought that this was the best feeling controls. The small dust particles, which get kicked up by the feet of the character might have also helped with that. The overall value of the fifth stage aligns quite nicely with their descriptions of the games feel. The added props and particle effects made the game a lot more complete and tied everything together. Shooting felt more satisfying with the added bullet shells, smoke trails, hit animations and the added permanence. There really is no point in denying that the changes clearly affected the way the players perceive the game. It even affected the things, which would usually be referred to as gameplay, even though the gameplay really only changed once. But even those adjustments didn't completely change the underlying simulation. They much rather modified some already existing parameters. From the first to the fifth stage the feeling of power quadrupled. The control value only gained two points. For the overall feel the players rated the last stage almost three times as high as the first one. As I said earlier, I had the players go through the first stage again after they had completed all five previous stages. When they had to assess the feel of the game for the second time, after they had seen what the end result of it all was, almost all participants gave a lower value than the first time around. On average all values decreased by about one point. This really doesn't show much, since the change is very abrupt, but I still wanted to make sure that the players didn't just get used to the games mechanics and therefore gave the game higher values.

4.2. Second User Study

The second user study was conducted as an addition to the first one. This is why I had only five participants for the following test. Again, these numbers are not nearly enough to actually arrive at a good result, but they are enough to show that it is possible to test these things. One of the main reasons why I had this second user study, was to look into the difference of the results, if the players can choose which aspects are going to improve. But to counteract the low number of participants I closely watched them play the game and tried to gather some more data this way.

I have already mentioned that I have used a different version of the questionnaire for this user study. The main difference is that I added four questions from the core module of the Game Experience Questionnaire (Poels, Kort and IJsselsteijn 2007). Since the wording of the questions is a little different to the way I decided to ask the questions, I tried to transfer them as best as I could, to minimize the difference. Also, the scale used is a little different, because in the original Questionnaire they used a scale from 0 to 5, whereas mine ranges from 0 to 10.

To use the scoring guidelines of the Game Experience Questionnaire you would first have to scale the values accordingly. The added questions of the questionnaire are as follows:

1. How skillful did you feel?
2. How much did you forget everything around you?
3. How bored did you feel?
4. How focused did you feel?

Since the general questions of the first user study didn’t yield any interesting results, I decided to remove some of them. I kept the ones, which seemed most interesting in the first user study, in case I could find some connections, but unfortunately the results were basically the same as in the first test. The four added questions, however, seemed to give some very interesting results. They were all connected to being in a flow state, which is a very interesting aspect to investigate.

To understand the data, it is important to know which tier of which category influenced which aspects of the game. For this I have provided a detailed list of all available upgrades in the appendix.

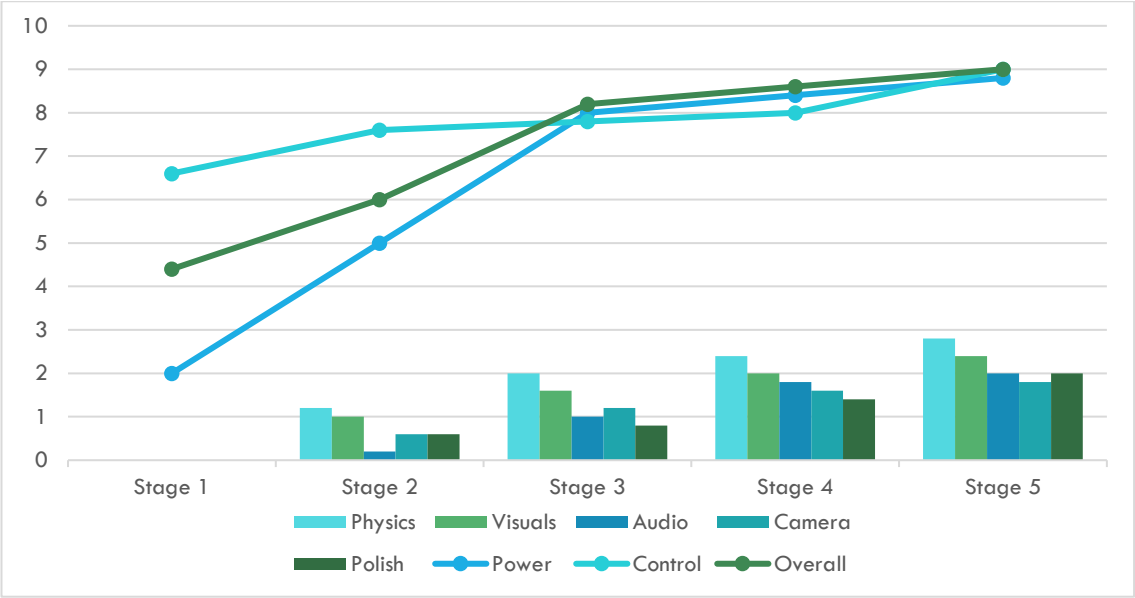

Table 2 Results of 2nd User Study

Let's first take a look at what the scores of the previous three questions look like. The bars on the bottom represent the current tier of the category, which range from 0 (no upgrades) to 3 (fully upgraded) and have been averaged as well. The values of *Power*, *Control* and *Overall* Feel in the first stage are almost the same as in the first user study. This is not very surprising, but it is still good to see a consistency across the tests. It shows that even though this whole topic is almost entirely based on soft metrics, we are still able to get consistent results. Overall you can see that the general trend is towards a better feeling game, but one striking difference to the first study is that the values reach a very high value more quickly. After the third stage all the values are already roughly at 8 points. After that there is almost no improvement anymore. Unfortunately, I couldn't find any correlation between the values and the selected upgrades. One reason for this could be that since the participants were allowed to choose the upgrades, they were satisfied with the game more quickly. It is very interesting, but saying anything other than that would be guessing.

The results for the Input category did not align with the findings of the first test. There was no drop off after they have purchased upgrades for the Physics category, instead the value increased at a steady rate. Even though the value doesn't change as much as the others it still shows that letting the player decide what he wants to change has an influence on the results. The reason for this might be that they already know how the characters movement is going to change and they have to actively make that decision.

It is also interesting to note that despite them not fully upgrading the game, the participants of the second user study arrived at a higher maximum for all three categories. This shows that if we give the player a choice, he will be more pleased with the result, than if we were to simply decide for him. This might be useful for game

designers if they found a way to use this in a videogame. They could for example give the player the option to change certain parameters, which some games have already done in the past. Most sports games, like *FIFA* or *Madden,* allow you to change the camera angle and the speed at which it follows the action on the pitch. Other than that, this part of the second user study did not reveal any new findings.

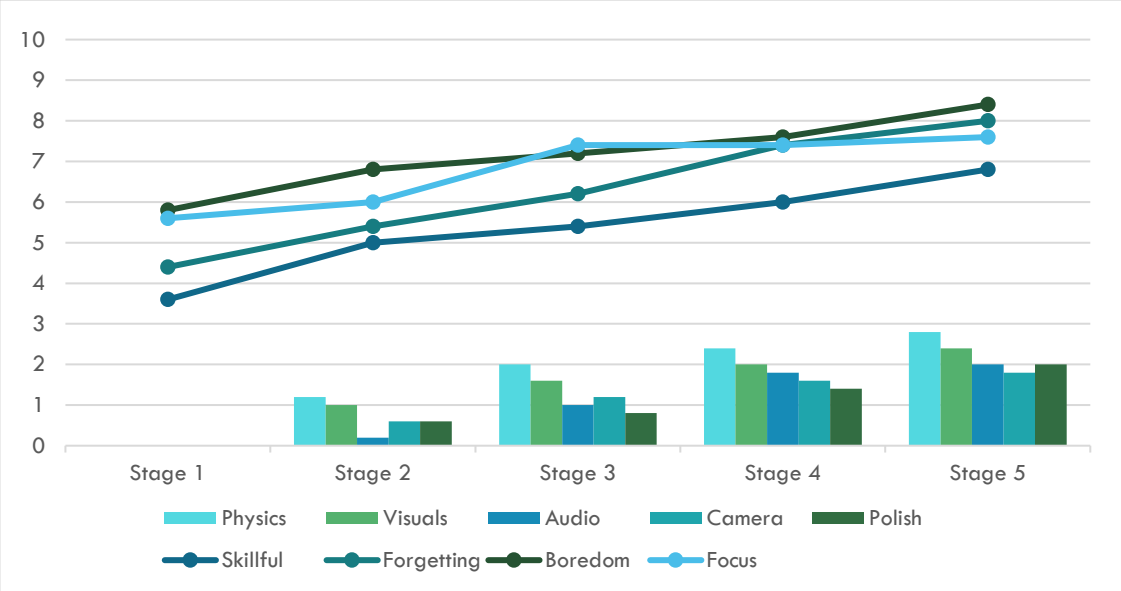

Table 3 Additional Questions of 2nd User Study

For the four additional questions the results again show a steady increase. It is important to note that a low value for the *Boredom* parameter means that the participant was very bored. The biggest increase was for the *Forgetting* parameter, where the participant had to rate how much he forgot everything around him. This shows that game feel is directly linked to the immersion of the game. This is also reflected by the value for *Focus* which also increased. Being focused on an activity and forgetting everything around you are two of the main indicators for being in a flow state. So apparently the feel of the game can influence flow. As I said earlier when I tried to classify the whole topic of game feel, there is some sort of connection between flow and the feel of a game. These results clearly show that my initial thoughts were correct and that there really is a connection.

Something that has been very consistent among all participants is the priority of certain upgrade categories. This can be seen quite easily by looking at the graph. It is obvious that most players would prefer a *Physics* upgrade over the rest. The category for *Visuals* is the second on the list for most players but *Camera* and *Polish* were almost no priority at all. Not a single participant has fully upgraded the *Camera* category. The fact that the priority for upgrades are in a descending order is not because of the ordering in the game, since most players spent quite some time carefully choosing which upgrade to get. It is more likely that this is due to the very obvious impact of the *Physics* and *Visuals* category, whereas the changes for *Camera* and *Polish* might not be as striking.

5. Conclusion

As I have already mentioned multiple times now, this thesis is not going to solve any problems or show incredible new results. The whole point of this was to create some awareness about this topic. Doing the user studies was simply a way to confirm my initial thoughts and to show that it actually is possible to test the impact of certain effects. However, these tests were not nearly big enough to reach a precise conclusion. Having only 15 participants in total is not enough to prove all the connections of game feel to other topics in game design. The time frame and scale of the project is also not sufficient for proper research, but it might give someone an idea for future tests.

The structure of this test was focused entirely on showing that the changes affect the experience of the player, but nothing about it was trying to show exactly which aspects correlate to certain psychological subjects. To do this you would have to build a game where there are multiple versions of the sprites or different movement parameters for example. By tweaking those parameters and experimenting with them you could show how they correlate to the experience of the player. In my project you could have one level where the character is the pixel-art robot and then slowly have it morph to a more realistic version of the same robot. Changing this could influence the expectation of the player about the physical interactions between objects. This could be one way of how you could test the impact of Metaphor, one of the key metrics described by Swink. Or you could give the player direct access to the acceleration, gravity and friction values. This could give you some insight into what ADSR envelopes feel the most satisfying for a certain type of game. One way to tackle the problem of terminology for game feel could be to have the game choose random numbers for all of the parameters and then have the players describe the feel of the game. It could yield some interesting results, but it is also possible that the players resort to very basic and vague words, which would completely defeat the purpose of the study.

The biggest problem with my project is that I was only able to focus on the Polish aspect of the game. Only a few upgrades were centered around the Response and Context metrics. Testing the other metrics would have required a completely different setup to what I have built. To test all six metrics at once is probably impossible, since there are simply too many factors that can influence each one. You have to focus on one particular metric to actually yield some interesting results. The ones I would be most interested in taking a closer look at would be the Context and Metaphor metric. How interesting would it be to see a game with similar mechanics to the *Super Mario* games but with high-fidelity photorealistic models. Only a few titles managed to really leverage these aspects to a point where people started to notice them. *Shadow of the Colossus* is one of them, but not a lot of people seem to notice the perfect spacing of platforms in *Super Meat*

Boy, to make jumping from one platform to the next more satisfying. I really hope to see more experiments with this whole subject in general and maybe even some more books and papers. It would be really amazing to see more game developers try to push the limits of game feel and maybe try something, that hasn't been done before. Unfortunately, this is all I can do at this point, but my findings were able to show that game feel is something that can be tested. It has an immediate connection to the game experience of the player and is even able to influence flow. Even though I couldn't give any answers to why there is a connection and what the characteristics of this connection are, I was also able to show that there is still a lot of research that can be done. Answering these questions could really help the whole industry in understanding why games are so much fun. The nature of the connection between the two topics is something that really should be investigated more precisely.

This thesis was my attempt at showcasing the possibilities within the subject of game feel. There is so much that we still don't know, and I hope that in the future we will be able to find more answers about the nature of game feel and how it interacts with all the different aspects of game design. Steve Swink has played a very big role in trying to figure out how everything works, but there are quite a few gaps in the knowledge presented by his book. An important thing to focus on at this moment is to find a more accurate term for this subject, since game feel is not very precise and can be very confusing. Kinesthetics might be something to look into as well and maybe test the connection between game feel and kinesthetics. Understanding the true nature of the sensation of handling a virtual object, might be one of the most difficult aspects about all of this. Gaining more knowledge about this could potentially help designers make better decisions about their games and how to fine-tune certain parameters. The key metrics presented by Swink in his book might be one of the biggest achievements in this field, but there are certainly more things that we can test and improve. He has given us the foundation to explore this new territory of game design. Comparing different games and looking at the way they create their unique feel can be a great tool for game designers. With the metrics Input, Response, Context, Polish, Metaphor and Rules we can take a closer look at how the many systems of these games work together to create this very special sensation, which is unique to videogames. It is such a young medium and we most certainly haven't yet reached the boundaries of what's possible. There have been many games, which have leveraged the power of game feel, but I hope to see more games that try to give the player a more visceral sensation of controlling a character. Being in the moment and getting absorbed by the game is one part of games, which cannot be replicated by any other medium. Everything else is a rather passive activity, but in games we have the chance to actually be the hero of the story.

I hope that my analysis of this topic was able to showcase some of the most interesting aspects about game feel and why I think that it is such an important part of game design. Many of the participants in my user studies were fascinated by the subject and were really interested in the general topic of game feel. I talked a lot to the participants after the user studies and most were not aware at all that this existed. Most of them admitted that they had never even heard the term, which really is a shame, considering that most of them were games engineering students. Even researching the topic was sometimes really difficult because there really isn't much that we know about it. Having to use a YouTube video for a definition of this huge topic felt very wrong. But the fact that it was one of the most formal approaches to this topic left me with no other choice. This again shows how big the gaps of our knowledge about this topic really are and I hope that we can find more accurate answers to some of these questions in the future. The possible connection to flow creates a whole new range of possibilities, which could influence a lot of different aspects of game design. I really do think that the feel of a game is the glue that holds everything together. It is not the most important aspect of videogames, but it would most certainly stick out like a sore thumb, if it was missing. Maybe there isn't even a clear answer to all these questions about how game feel works and what it does, but the impact it has on our gaming experience is big enough to warrant some discussion.

6. Works Cited

Chen, Jenova. "Flow Theory." *JenovaChen.com.* 2006. https://www.jenovachen.com/flowingames/flowtheory.htm (accessed December 28, 2018).

Daniels, Derek. "Why some games feel better than others – Part 1 of a gazillion." *LowFierce.Blogspot.com.* 2006. http://lowfierce.blogspot.com/2006/05/why-some-games-feel-better-than-others.html (accessed January 28, 2019).

Eiserloh, Squirrel. "Math for Game Programmers: Juicing Your Cameras With Math." *YouTube.* 2017 https://www.youtube.com/watch?v=tu-Qe66AvtY&t=412s (accessed December 27, 2018).

Franklin, Chris. "Errant Signal - Kinaesthetics" *YouTube.* 2012. https://www.youtube.com/watch?v=MrFK4NlGCo0 (accessed January 29, 2019).

Hunicke, Robin, Marc LeBlanc, and Robert Zubek. *MDA: A Formal Approach to Game Design and Game Research.* 2004.

Komiku. "Battle of Pogs" *FreeMusicArchive.* 2018. http://freemusicarchive.org/music/Komiku/Captain_Glouglous_Incredible_Week_Soundtrack/pog (accessed November 18, 2018).

McCloud, Scott. *Understanding Comics.* William Morrow Paperbacks, 1994.

Nijman, Jan Willem. "The Art of Screenshake" *YouTube.* 2013. https://www.youtube.com/watch?v=AJdEqssNZ-U&t=1357s (accessed January 28, 2019).

Nitsche, Michael. *Video Game Spaces: Image, Play, and Structure in 3D Game Worlds.* The MIT Press, 2008.

Poels, Karolien, Yvonne de Kort, and Wijnand IJsselsteijn. *Game Experience Questionnaire: development of a self-report measure to assess the psychological impact of digital games.* Eindhoven: Technische Universiteit Eindhoven, 2007.

Rogers, Tim. "In Praise Of Sticky Friction." *Kotaku.* 2010. https://kotaku.com/in-praise-of-sticky-friction-5558166 (accessed December 26, 2018).

Saint Augustine. *Confessions.* Penguin Classics, 2003.

Swink, Steve. *Game Feel: A Game Designer's Guide to Virtual Sensation.* Routledge, 2008.

7. List of Figures

8. List of Tables

9. Appendix

9.1. Results of User Study

9.1.1. First User Study

	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Stage 6
Power	1,7	3,5	7,1	7,7	8,4	0,8
Control	5,1	6,5	4,8	5,9	7,1	3,8
Overall	3,3	6,2	6,4	6,9	8,5	2,1

9.1.2. Second User Study

	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
Power	2,0	5,0	8,0	8,4	8,8
Control	6,6	7,6	7,8	8,0	9,0
Overall	4,4	6,0	8,2	8,6	9,0
Skillful	3,6	5,0	5,4	6,0	6,8
Forgetting	4,4	5,4	6,2	7,4	8,0
Boredom	5,8	6,8	7,2	7,6	8,4
Focus	5,6	6,0	7,4	7,4	7,6
Physics	0,0	1,2	2,0	2,4	2,8
Visuals	0,0	1,0	1,6	2,0	2,4
Audio	0,0	0,2	1,0	1,8	2,0
Camera	0,0	0,6	1,2	1,6	1,8
Polish	0,0	0,6	0,8	1,4	2,0

9.2. Upgrade List for Second User Study

- Physics:
  - Tier 1:
    - Improved Walking/Acceleration/Deceleration
    - Dynamic Jumping
  - Tier 2:
    - Higher Rate of Fire
    - Faster Bullets
    - Less Accuracy
  - Tier 3:
    - Enemy & Player Knockback
    - Faster Enemy Movement
- Visuals:
  - Tier 1:
    - Animated Sprites for Character, Enemies & Bullets
  - Tier 2:
    - Background Image
    - Dynamic Animation Speed
  - Tier 3:
    - Props
    - Post Processing Effects (Chromatic Aberration, Bloom, etc.)
- Audio:
  - Tier 1:
    - Sound Effects
  - Tier 2:
    - Music
- Camera:
  - Tier 1:
    - Smooth Camera Movement
  - Tier 2:
    - Look Ahead
  - Tier 3:
    - Screenshake
- Polish:
  - Tier 1:
    - Bullet Impact Particles
    - Weapon Kick Back
  - Tier 2:
    - Enemy Hit Animation
    - Dust Particles at Feet
  - Tier 3:
    - Bullet Smoke Trails
    - Permanence
    - Player Death Animation
    - Screen Flash on Death
    - Screen Fading