

Arsi Liimatainen

Game loops and effective prototyping
Designing core gameplay for a hack and slash
game

Bachelor's thesis

Bachelor of Culture and Arts

Game Design

2022



South-Eastern Finland
University of Applied Sciences

Author	Degree title	Time
Arsi Liimatainen	Bachelor of Culture and Arts	March 2022
Thesis title		64 pages
Game loops and effective prototyping Designing core gameplay for a hack and slash game		
Commissioned by		
Supervisor		
Marko Siitonen		
Abstract		
<p>This thesis studied the methods of creating engaging core gameplay, with a focus on hack and slash games in particular. As the primary way of game development is through prototyping, methods of effective prototyping were also studied as a secondary subject. The theory was put to test through creation of a game prototype.</p> <p>Knowledge was gathered by studying previous research of the subject and analyzing core gameplay of two games of the hack and slash genre. Then, the learnings were used in creation of a hack and slash game idea prototype. First, a Tabletop Simulator prototype was created to efficiently design the rules of the game idea. After that version went through playtesting, the development moved on to a digital prototype, which was then tested with other players.</p> <p>To create engaging core gameplay, it was found important for various elements such as challenge, interesting decisions and good game feel to be present in the game. To make sure the game flows well, it is important to create a clear core game loop and design the game following that. All design decisions should also be based on a well-defined game experience that caters to the target audience.</p> <p>Applying this theory in the creation of the game prototype was successful. All the players the game was tested with clearly enjoyed the core gameplay created and there seemed to be potential for a full game to be produced based on the prototype. The usefulness of the methods used should be further studied through development of more unique gameplay ideas, as in this case previous games in the same genre had strong influence on the design. This could have made evaluating the methods more unreliable.</p>		
Keywords		
game loop, core gameplay, hack and slash, prototyping, game design		

CONTENTS

1	INTRODUCTION	5
2	WHAT CREATES GOOD GAMEPLAY	7
2.1	Core gameplay and game loops	7
2.2	Challenge and flow	10
2.3	Choices	11
2.4	Game feel	12
3	HOW SHOULD GAMES BE PROTOTYPED	13
3.1	Low fidelity prototyping	14
3.2	Digital prototyping	16
3.3	Playtesting	18
4	INITIAL CONCEPTS OF THE GAME PROJECT	20
4.1	Premise of the project	20
4.2	Target audience and intended game experience	22
5	SHORT ANALYSIS OF SIMILAR GAMES	23
5.1	Hyper Light Drifter	23
5.2	Hades	27
5.3	Analysis conclusion	30
6	LOW FIDELITY PROTOTYPING THE GAME IDEA	31
6.1	Tabletop simulator as the low fidelity prototyping method	31
6.2	Drafting and testing the base mechanics	31
6.3	Adding the grappling hook mechanic and ranged enemies	36
6.4	Playtesting the low fidelity prototype with other players	40
7	DIGITAL PROTOTYPING PROCESS	43
7.1	Creating the basic movement and combat mechanics	44
7.2	Creating the basic grappling hook mechanic	48

7.3	Adding ranged enemy and respective tweaks	50
7.4	Adding rotational grappling hook mechanic	51
7.5	Adding feedback and building the levels.....	53
7.6	Playtesting digital prototype with other players	55
8	CONCLUSIONS AND RELIABILITY.....	59
	REFERENCES	63

1 INTRODUCTION

As game development tools and resources have become much more accessible to the general population, the number of games being created has increased dramatically. In 2020, over ten thousand games were released on Steam alone (Clement 2021). Yet it is a known fact that many of the released games fail to recoup their development costs, and even more never end up releasing in the first place. Game development is risky, expensive and time-consuming endeavor with many moving parts, and success is behind hard work and some chance. It is more important than ever to make sure the game you are making is good and meets the high expectations of the gaming audience. No amount of money and time spent in marketing will save a bad game from failure.

While there is an incredible number of different aspects in what makes a good game and each can be studied in depth, this thesis will focus solely on the gameplay. As Tracy Fullerton (2014) writes in her book *Game Design Workshop: A Playcentric Approach to Creating Innovative Games*, what makes a great game is solid gameplay. Even if players say they really loved the visual aspects or the story for example, they would have likely quickly stopped playing the game if the gameplay wasn't entertaining enough. (Fullerton 2014, 3.)

The main research question this thesis tries to answer is how to create engaging gameplay, with focus on a hack and slash game genre. Naming of the game type was deemed important for the question, as the gameplay expectations for any game are largely set by the views the audience has on the genre. The research will be carried out on a small game project prototype, which leads to the secondary problem that will be studied: how exactly does one prototype effectively. If a game prototype's purpose is to save time and money by evaluating the game idea before committing into a full production, proper methods should be used to make sure the prototyping process fills that purpose instead of aimlessly creating different mechanics without clear direction.

First, this paper will go through some theory of what exactly is needed to create great core gameplay by referring to multiple different sources and looking into what has been done in games that fit in the same genre as the game project idea, to better understand expectations of the target audience. Then, the research is utilized in creation and testing of a prototype, as the only way to find out if a game idea works is through a cycle of prototyping and testing. The process will start from low fidelity, fast to make prototype and then move on to the Construct 3 game engine.

As the game idea is still only at a very early stage and the prototyping is properly started during the thesis process, it is entirely possible the main ideas of the game may have to go through various changes during the process, perhaps even changing entirely. However, some very basic mechanics testing was done beforehand in the game engine to ensure the technical side should be doable, which reduces this risk.

It is important to note that the focus of the thesis is on the early development and design of game ideas by focusing on the core gameplay, and early prototyping. This means that some subjects will be left to a more superficial amount of information, as they would only become more relevant when designing the entire game experiences after the core gameplay has been confirmed. This includes subjects such as the visuals and audio. While good visual and audio for example are extremely important in giving actions proper feedback, which is definitely a part of good core gameplay, focusing on that in such early phase of exploring the idea would be far too time consuming.

This paper also assumes the reader already has some previous knowledge of game design and terms related to designing gameplay, such as objectives, rules, resources and so on. To explain all the basic concepts would take long and simply repeat what other authors have laid out extensively in their own written works.

2 WHAT CREATES GOOD GAMEPLAY

The first answer most people would have when asked what good gameplay is would be “good gameplay means the game is fun to play”. A simple sounding goal, yet the methods to arrive there require good knowledge of both the wants of the game’s target audience and general game design principles. The word “fun” is very vague and can have different meanings for different people and different types of games. A better term to strive to achieve in a game project could be “engaging” gameplay. Engaging gameplay is the kind that keeps the player entertained and coming back for more, while likely making them think they are having fun, whatever their own definition of fun happens to be.

While some elements of good and engaging gameplay can be considered universal, it is important to understand that there exist incredibly many different types of game genres with completely different expectations of how they should play and what kind of experiences they should give the player. This chapter will focus on researching the general concepts, while the later ones focus on using the information for the game project associated with this thesis to become engaging and fun. The topics covered were also picked based on what was deemed the most useful for the creation of the game idea. To cover all the possible topics related to creating good games or game mechanics in general would require a much more in-depth study.

2.1 Core gameplay and game loops

Games like Tetris show that a game does not need to have dozens of different features or mechanics to be engaging. A few simple mechanics, when well executed, can create very entertaining game experiences. On the contrary, adding too many features without forethought often takes away from a game’s enjoyment. It is very tempting when creating games to add more and more features, yet often the games that stick to their vision of the player experience and cut away any features that are not necessary are those that truly engage their players.

The actions that the players most often perform in a game while trying to achieve its overall goal can be defined as the core gameplay (Fullerton 2014, 210). The core gameplay defines what the game really is about and gives it a genre. As such, it is natural to assume that these actions should be well designed to achieve great gameplay. When done well, the secondary features of the game will be far easier to design on top of the core gameplay.

Another often used definition in game design is game loops. Game loops gather information of player input, how the game system handles it and what is the following output into a loop structure. It is basically an abstract concept to help make good game design decisions. Game loops help understand what the players do and why while playing a game, and they can be defined with different level of granularity as needed for the research in question (Sicart 2015, 3). By utilizing the game loops when designing game mechanics, it is easier to understand them, their relations to each other and make sure that every action has proper feedback. They help the designer to think everything through in a way that will make the gameplay clearly structured.

By combining the concepts core game mechanics and game loops, we get to the idea of core game loop: the main actions that structure the entire game's design and that engage players repeatedly in a looping sequence (Lovato, 2020). These core game loops are most often discussed within mobile game development, where the games often have only one clear game mechanic that they base themselves around. However, they can be identified in all types of games, and by no means only apply for simple games. The core game loop is often very similar in games of the same genre.

When the core game loops are well defined and fun, the design of the game is likely very solid already, and the players can more easily understand the mechanics. It will also be much easier to expand the game and add depth when the core game loops have been designed well, by adding more mechanics and interactions with other, larger scale game loops. These game loops will be linked

into each other at different levels, making the games into the interactive, interesting pieces of entertainment they are.

As an example, a couple simplified game loops were visualized from Dark Souls (Figure 1). The core game loop of the game is about fighting enemies and has three distinct phases. First, the player must find or create their opportunity to act, usually to attack the enemy. However, the enemies try to attack the player as well and are often placed in groups, making them impossible to beat by simply keeping on attacking. This forces the player to go on the defensive every now and then and use more varied approaches. At that point the loop has gone back to the beginning, where the player tries to find the right moment when it is safe to act again.

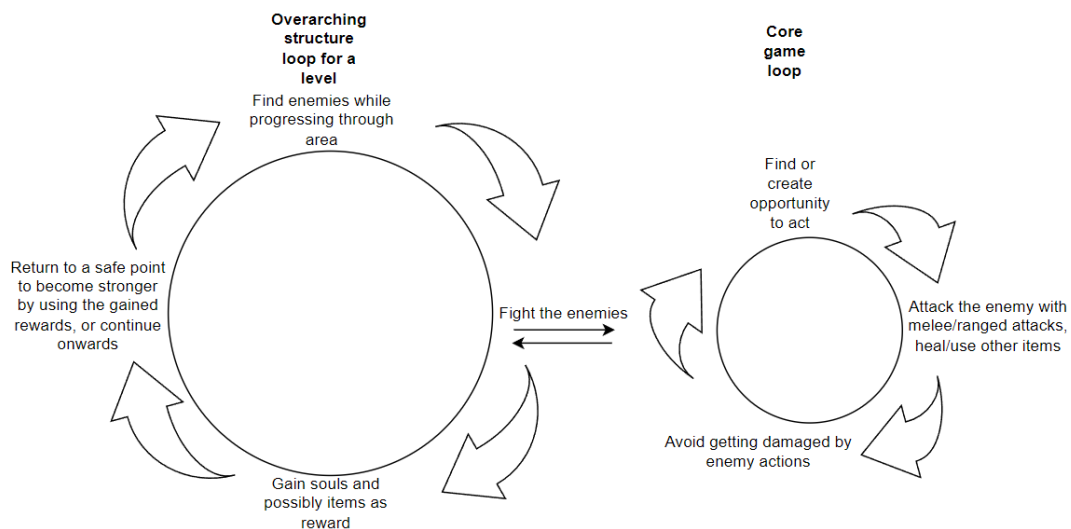


Figure 1. Two simple game loops portrayed for Dark Souls

On the left side there is a larger scale loop that the core game loop links to, on the scale of going through a single level. The player finds groups of enemies in a level, engages them in a fight, and gains rewards. At some point they will use these rewards to level up and become stronger, making further challenges easier for them.

These two loops are a simple example of how game loops can be used to portray game mechanics. A lot more detail and loops could be added to fulfill the picture of the whole game, but as we can see already from the simplified actions, the

game has a clear structure that is easily identifiable for the player: find enemies, defeat them by finding the right opportunities, get stronger. By visualizing the actions this way, it should be easy to see if there are points where the game gets stuck in the process instead of looping well back to the beginning. Essentially, game loops are a very useful visualization tool to use when designing and testing game ideas.

2.2 Challenge and flow

Games enjoyability heavily relies on various challenges they impose on the player. The level of this challenge will vary vastly between different games as they are tailored to their target audience. Challenge can take many forms, from manual dexterity and reflexes to puzzle solving and strategy. It is important to know what kind of challenge will fit the target audience of the game and design with that in mind.

No matter the type of challenge presented, it is the moment where the challenge's difficulty level is perfectly in tune with the players skill level where games are most engaging, and the players feel like hours pass by in just moments. This state of mind is referred to as the flow state. For more in-depth look at the concept of utilizing flow in video games, one can read *Flow'n hyödyntäminen videopeleissä* by Ari Karppanen (Karppanen, 2016).

For some condensed guidelines, Sean Baron offers few select directions games could follow to make their players more easily reach the flow state, and thus be more engaging. These directions include giving players clearly defined goals with manageable rules, fitting goals to the player's abilities, clear and timely feedback on performance, and making sure any distracting, unneeded information isn't given to the player, especially on critical moments of the gameplay. (Baron, 2021.) By following these guidelines, the players won't be overwhelmed by the amount of information they have to process. Should the players manage to enter the flow state, their motivation for continuing their actions becomes the experience itself, rather than some extrinsic reward. (Fullerton 2014, 101.) And

that, if anything, is a sign of good and engaging gameplay the players will want to return to over and over.

Of course, all of this might be difficult to handle while still designing the core gameplay and many questions remain open on the direction of the game's overall direction. The focus should be on the rules deemed most important to the game's core gameplay, whereas for example trying to match the challenge to the player's abilities should have more focus later in a balancing stage of development.

2.3 Choices

For the challenge to exist, there must be choices for the player to make. After all, playing a game means constantly making big and small choices as you interact with it, some of them more conscious than others. For these choices to be interesting, they must somehow alter the course of the game. If there are no choices to be made, or the choices are so obvious they are not really choices after all, the activity of playing quickly loses its appeal and the whole point of games as an art or entertainment medium, interactivity, vanishes.

Making choices when there is an emotional element included, plenty of information to base the decision on or having balanced outcomes on either option create more interesting choices. On the contrary, choices without enough information to base them on, no consequences or obvious best option present are not interesting to make. (Fullerton 2014, 347-349.) It is easy to think of examples of these kind of choices in story games where the narrative moves forward based on your choices, but choices should also be closely looked at when designing the core gameplay. By making sure the player must make interesting decisions while interacting with the core game mechanics, the game will be more engaging as the player has an interesting activity at their hands whenever they play, rather than just on specific moments.

Fullerton (2014) additionally argues that it is important to take note of the importance of the decisions. The more important the decision is, the rarer it should be. If dramatic decisions with serious consequences appear constantly, the player will tire of them, and their impact becomes vastly lower. In general, keeping the player constantly making choices at a rapid pace could also tire them out. The choices should be balanced well with the overall flow of the game in mind. (Fullerton 2014, 348.)

2.4 Game feel

One principle that contributes largely to the enjoyment of games, especially those where action happens in real time, is the “game feel”. One definition of game feel is tactile sensation of controlling a digital object (Swink 2007). Games that nail down the game feel are usually praised by their players for their tight and responsive controls. Games of this type are usually those where the player controls a highly mobile character and much of the gameplay deals with using the movement to its maximum. Good examples of such games are platformer games such as the Mario games, *Hollow Knight* (2017) or *Celeste* (2018).

In their blog post at Game Developer (previously called Gamasutra), Steve Swink (2007) separates good game feel to six parts, and by focusing on these parts it should be easier to achieve. The first part is input: how much the player can express their input to the system. This is inherently tied to the kind of controller the game is played with: a mouse and keyboard is very different tool from a gamepad. Input devices have different capabilities of sensitivity for input, which has to be taken into account with the game mechanics. So called natural mappings can also be established: the kind of controls the player can understand immediately without more instructions. For example, if left analog stick of a game controller directly moves a character on screen in the exact same motion, that would be a natural mapping. (Swink 2007.)

The second part is the actual system’s response to the input. This is the other side where the sensitivity can be adjusted. A good example of this is holding the

jump button to jump higher in most platformer games. Even though the jump is usually just a single button with two states, it can translate to a responsive action in the game. By tweaking the input and the response, the amount of sensitivity can be controlled to a very high degree, to what result suits the game being designed. (Swink 2007.)

The third part of the game feel according to Swink is the context of the actions. Being able to perform interesting actions with a character means nothing if there is not a suitable environment to use these actions. Only after building fitting obstacles for the character the player controls to play around with, does the good game feel truly start to emerge. (Swink 2007.)

The last three parts are polish, metaphor and rules. In this case, Swink's use of polish refers to making the interactions truly feel alive by providing different feedback in the form of audio, visuals and interactions with other objects. This helps bring the actions of the player to life. Metaphor refers to expectations the player has: if the object the player controls is a car, they have strong assumptions how it should control already from before the game. Rules on the other hand were defined as the ultimate purpose of the mechanics: what are they meant to be used for, what is the goal of the game. (Swink, 2007.)

By following guidelines such as this while designing games that have a lot of importance on movement of the player character, the core gameplay should be elevated to a higher level. When executed well, the simple act of controlling the character will become engaging and fun, being its own reward for the player's input. If the movement mechanics are tuned for good game feel, it will be easy to expand upon them by use of smart level design that allows the player to truly show their mastery over the mechanics as they overcome various challenges.

3 HOW SHOULD GAMES BE PROTOTYPED

As it was discussed so far, games are an interactive medium of entertainment or art. This means the only way to truly evaluate games and understand what is

being created is by including this interaction in the creation process itself. In practice this means creating prototypes and repeated testing, both on your own and by inviting other people to play the game.

3.1 Low fidelity prototyping

How should one begin the prototype process, once an interesting idea has emerged and main concepts to test have been decided? Far too often inexperienced developers jump directly into making a digital prototype. The problem with this approach is that creating digital prototypes is incredibly time consuming, and it is easy to get caught up in the graphics and simply making things work the way you want. (Fullerton 2014.) After you noticed the game idea simply doesn't work, countless hours might have already been wasted in trying to make the idea work digitally where every part of the process is slow.

Instead, prototyping should begin from a lower fidelity approach and build up from there. So called physical or paper prototyping should be used. The point of these prototypes is to get to test the concepts as fast as possible, not wasting any time and more directly seeing how changing the game rules and mechanics affects the feel of the game. An additional benefit is that if the game is engaging to the testers already in this state with just the game mechanics implemented and no focus on good visual representation, the design is very likely well done. (Fullerton 2014, 197.)

These physical prototypes can be made from anything the designer has on hand. The only point is the ability to change the rules quickly and get to testing as soon as possible. Usual items for testing are paper, pens, different dice and tokens and so on. Experience with board game design can be helpful, as physical prototyping is usually the most used method during their development.

When approaching the very first prototypes for an idea, it is important to begin from the very core game mechanics. They should be identified and visualized in a diagram of some sort, for example the game loops discussed earlier. By

visualizing the mechanics roughly early on, it is easier to spot possible mistakes in the design or feature ideas that don't seem to integrate to the core game mechanics (Fullerton 2014, 211).

Fullerton (2014) describes the process as consisting of four parts. First you build the foundation, the very core gameplay with as few mechanics and rules as possible. In this phase more and more questions from the game will come up, and should be noted down, but not focused on yet if they are not integral to the core gameplay. In the foundation stage, base objects and the key procedures of the game are the most important and the core gameplay should be kept down to as few rules as possible. (Fullerton 2014, 211-212.)

Next is the structure stage, where you build the framework for the actual game from the core mechanics. It should be able to support the various features that the game may have in the future. It is important to identify what are the essential rules and what are the features they will have to support. It is also good to note difference of rules and features. Rules define how the game plays, whereas features are attributes making the game richer, and always require new rules to be added. The focus at this stage should be at defining the rules and leave additional features for later. (Fullerton 2014, 224-225.)

The third phase is adding more formal details; possibly some of the features that have come to mind so far but weren't deemed important enough for the core gameplay yet. Each addition should be carefully tested, and the less rules the game ends up having the cleaner the design is. All these additions should be tested one by one, and only those integral to the game should be left in. (Fullerton 2014, 226.)

The last stage is refinement, where the game is a playable system but can still have some rough edges. The questioning moves from if the game works in general to smaller details, and if the game is fun and engaging in general. This phase includes adding any features that weren't important enough to core gameplay so far but might be wanted. Like the last phase, they have to be tested

one by one to see their effects to the game structure properly, and only the most important ones should be included in the game. Testing is at the core of the process at this phase. (Fullerton 2014, 226-227.)

It might be difficult to follow such phases exactly in the process of prototyping, as sometimes their differences can seem vague. But the key point is to give the prototyping process structure and meaning. The designer should know exactly what they are testing and why. Fullerton also emphasizes that every prototype or iteration of a single prototype should have a well-defined question they try to answer (Fullerton 2014). After the answer to the question is received, it will be clearer what to do next.

3.2 Digital prototyping

Digital prototyping is the next step for video game ideas after physical prototyping has provided the required results. However, for certain types of games it might be valid to start from a digital prototype (Fullerton 2014, 236). For example, if the game's core gameplay relies entirely on fluid movement mechanics of the player character, it would be difficult to emulate those mechanics with paper prototyping. It is up to the designer to figure out what is the best prototyping method for each particular idea or mechanic.

As digital prototypes are far more complex with many moving parts and problems to solve, prototyping should be more focused on single mechanics or features at a time. There can be multiple of these prototypes for the different features, and they should be done as fast as possible just like in physical prototyping. This often means they will be discarded once the actual production begins, and more clean coding solutions must be used. (Fullerton 2014, 235-236.) In general, to not waste time it is important to make a "toy" version of the game mechanics first, to see if it works (Gabler 2009).

The good practices of physical prototyping discussed in the previous chapter apply all the same to digital prototyping, and if a physical prototype was made

before moving to digital, many of the designs should be easy to create as the design decisions have been made and just need implementation in the digital version. It might still be worth it to move back to physical prototyping every now and then if new features come up.

Apart game mechanics, in the digital implementation there is a need to somehow visualize what is happening. While there may be some prototypes that focus on the artistic side, in early gameplay focused prototypes these issues are often not spent any time on. It would after all be a waste of time to spend time creating aesthetic elements for the game if the idea will be scrapped. Fullerton however mentions that this rule can be broken every now and then, if a bit of visual or audio elements would be integral to the game mechanics (Fullerton 2014, 238). The key is to find how much time is it worth spending compared to the improvement the inclusion of some visuals or audio brings to the testing of mechanics. Utilizing ready-made assets found from the internet for example can be done very fast and have great results.

Depending on the team working on a game, it is also of course still important to keep in mind exactly how the visual side would work in the final product. It would make no sense to design and test game mechanics for a game which requires visuals the team is unable to produce in the final product. Often the coding of the game's mechanics depends on art too, for example launching attacks on a certain frame of animation. While the aesthetic elements should have minimal amount of time spent on them in the early prototyping process, it is still important to understand exactly how that side would work in the final product to make sure the game mechanics won't need to be compromised on later as a result. The same of naturally applies to any non-aesthetics related technological requirements as well.

Game feel was already discussed in its own chapter and is something that can only be tested in a digital prototype. In their post Swinks mentioned building a "garden" to help testing game feel. Before focusing on any external mechanics apart what the character can do, a sandbox of sorts would be constructed. It

would include the various objects the character can interact with such as platforms to jump on. The game mechanics could then be tested in this garden by focusing on the six rules mentioned in the game feel chapter, before introducing any external mechanics and objectives. (Swink 2007.)

3.3 Playtesting

Playtesting is an integral part of the prototyping process, first with the designer themselves, then playtesting with other players and eventually focusing on testing with the intended target audience. As game designers spend hundreds of hours working on their game, it is easy to lose the objective view and understanding of how a new player would see the game. Therefore, getting feedback is vital and should be done throughout the development process. Games often include high amount of emergence as different systems work together, and the only way to understand these systems is to see them working (Adams & Dormans 2012, 47).

Fullerton (2014) in fact argues that playtesting is the most important activity the designer participates in yet that it is often misunderstood. The goal of the testing should be to gain insight how the players experience the game and continue through the whole development. There are numerous different ways to conduct playtesting and what to choose will depend on the game and the size of the team or company working on the game. Yet often inexperienced designers only rely on testing without proper preparation or only test when the game is so late in development changes to the core gameplay systems are too late to make. (Fullerton 2014, 271-272.) You should always test a work in progress, not a finished design, and usually start testing as early in the process as you can (Pozzi & Zimmerman 2012, 2).

The testing and prototyping processes are inevitably intertwined. This is called iterative design. After each iteration of the prototype as changes have been made, feedback is received by testing either alone or with others, and further changes are made based on observed results. As the game goes towards its finalization, the time between these iterations gets shorter and shorter but with

less drastic changes in-between. Similarly, the kind of people you test with should change. First the developer is testing their system alone until it warrants testing from others, moving to testing with people they know as they are easily available. Later the testing should be more involved with unaffiliated people who will have no personal ties to the designer that could hold back honest feedback, and who should also ideally belong to be the target audience. (Fullerton 2014, 273-274.)

Like with prototyping, it is useful for the tests to have a clear goal the designer tries to get an answer to (Pozzi & Zimmerman 2012, 2). This could for example be “does the player learn how to use the new abilities in this level effortlessly”. It might not take that long to get these answers. Most of the important observations will likely be received already after 6-12 unique testers and continuing testing with more people would simply provide the same information again and again (Zukowski 2020). At that point it is time to move on to the next iteration before testing again.

To get the wanted information out of playtesting, there are various qualitative and quantitative methods one can use to gain information from the players, and again the choice should depend on what the designer is looking to understand. At the core of especially early game concept testing, however, is simply observing the players play and taking notes of what they do, where they struggle and if the designs worked as intended with them. They should be encouraged to speak aloud their thoughts constantly to really get understanding of what they are thinking, while the designer should try to make themselves as invisible in the situation as possible, not correcting, advising, or explaining something to the player unless they get stuck for a long time (Zukowski, 2020). Only the bare minimum information required to play the game should be explained before the test, and the more complete the game being tested is, the less need there should be for such explanations, as the game should contain all necessary instructions and hints how to play.

To utilize the information from testing, recording the information somehow is vital. It is impossible to remember all the details of the test otherwise. The common

method is to simply take notes, which can additionally encourage the testers to give more in-depth feedback as they see you really caring about what they say (Pozzi & Zimmerman 2012, 4). If possible and with permission from the tester, recording the testing session on video can be an extremely helpful way to do this. As the footage can be reviewed carefully to gather accurate information of what exactly happened, it is both easier to analyze the player's behavior as well as notice exact circumstances under which the code or design might have had a dysfunction.

It is also good to note that a group of testers is good for generating ideas, but very bad in evaluating them since the testers can affect each other and sway the whole group's opinion to certain directions. This shouldn't dissuade from playtesting if a group testing is the only option available, however. (Fullerton 2014, 281-282.) For games like board games, it might be the only option after all. In such cases, clear communication to the testers is key. Making sure everyone feels their opinion is welcome goes a long way, as well as asking questions that focus on how the players feel rather than the quality of the game.

4 INITIAL CONCEPTS OF THE GAME PROJECT

This chapter will explain the initial direction of the game project that will be developed by utilizing the theory explained in previous chapters. A short look is also taken into a few similar games to see how their core gameplay has been designed.

4.1 Premise of the project

Game ideas could be categorized into two distinct approaches. The first is a more innovative approach where the developers try to create a game with entirely new kind of gameplay not really done before, and if successful, potentially creating entire new game genres. The other approach is making games that clearly fit into an established game genre that already exists, but often have some defining core feature, unique setting or style that sets it apart from other games of the genre. This game idea falls to the latter category.

The genres that the game idea would fall under are action games and hack and slash games. Action games are one of the most popular types of games played (WePC 2020). It is a loosely defined genre, but the main defining factor is that the gameplay is real time and fast paced, requiring the player to act fast and utilize their reflexes.

Hack and slash games are most of the time classified under action games and are somewhat difficult to define. The main defining factors for them could be the combat system being mostly based on melee range attacks, fighting enemies over and over being the core gameplay and said enemies often coming in massive numbers. The combat often looks stylish or over the top, as it should since it is the main gameplay activity and must stay interesting to the player over hours of play.

If these action hack and slash games usually have core gameplay of using melee-based combat systems to fight enemies, the unique factor to set the core gameplay apart from other games must come from how the combat system is executed and any possible unique and interesting game mechanics.

For this game idea, the unique mechanic is grappling hook mechanics. The player will be able to use the grappling hooks to pull themselves to various objects and even directly into enemies, as well as rotate around objects. These moves will grant the player a great amount of mobility to reposition themselves in the battlefield and dodge enemy attacks.

It is also planned that the player can use these grappling hook moves directly as attacks, slicing through hordes of enemies in a stylish spectacle. One can think of Attack on Titan's 3D maneuvering gear scenes as a good example of the kind of feeling wanted for the game's action. The game will be restrained into two-dimensional movement, as the view of the game will be isometric. This type of view is often used in hack and slash, one reason being it gives the player clear view of all the enemies around them, making large hordes of enemies feel fairer as you can react to them easily.

Currently, the game idea is not locked into any theme or setting, as the mechanics will be explored first and foremost. However, some kind of steampunk, low fantasy setting is the current expectation. A fantasy setting would give a lot of freedom in coming up with different enemy designs.

Should the core gameplay prototyping go well, and the game be further developed, the game would likely have some roleplaying game elements in it, such as upgrading the player abilities with a skill tree. However, this thesis will only focus on developing the very core gameplay combat system.

4.2 Target audience and intended game experience

While this game project is currently just a hobby project for this thesis, it is made with the assumption it could have potential to become a commercial project, should great potential be found in the core gameplay. This means that the business side should be kept in mind from the start, although with not so strict influence. The game industry is extremely competitive, and developers need to know what kind of audience they are making the game for. Knowing the target audience will help with making the correct design decisions and defining the player experience that the game wants to give to its players.

As an action game, the game is expected to mostly cater to the younger audience who like fast paced games and proving their competence in games by overcoming difficult challenges. These people are likely to call themselves hardcore gamers and are more likely to be male. They likely spend a lot of money into gaming and play many games. That combined with the fact that action games are one of the largest genres makes the competition tough as the players have very high expectations of the product, and the game has to really stand out if it were to become a commercial product.

These players will likely want to feel powerful and cool while playing, and as such they seek out hack and slash games where the player character can perform

incredible feats in combat against impossible odds. They are also less likely to care about the story unless it really grips them, so story elements should be kept lightweight and not spent too much effort on.

These target audience expectations combined with the initial game ideas create a good picture of the intended game experience that the core gameplay should attempt to convey. The player character should be powerful, but combat should still be challenging; in practice this could be that the player can kill the enemies fast, but also has low health themselves, making mistakes dangerous and skillful use of the game mechanics important. When the player does well, the game should clearly give strong feedback to reward them, mostly in form of the visual satisfaction of the combat, which also supports the player fantasy of playing a powerful character. Fights should feel quite fast paced, yet a skilled player should feel like they are in control of the situation.

5 SHORT ANALYSIS OF SIMILAR GAMES

Before starting development of the prototype, it was deemed beneficial to take a short look into some similar games and how they have made their core gameplay and combat system work. After all, all games are just a unique mix of already known concepts, and it would be beneficial to study solutions that have been created before.

5.1 Hyper Light Drifter

Hyper Light Drifter by Heart Machine (2016) was chosen as a game to look at since its combat has somewhat similar fast and risky combat as the concept of the project idea, and its overarching progression seems very similar in style and scope to what the game idea would likely be like as well. The game has also been well received by both players and game critics.

In Hyper Light Drifter (2016), the combat has been tuned down to just few simple mechanics that are executed well. The player's default offensive move is attacking with their sword, chaining to a combo of up to three hits. The swings

can hit multiple enemies at once and move the player slightly forwards to make it easier to land hits. Most enemies get staggered from each hit of the weapon, preventing them from acting, unless they have already started their attack animation. In that case, the enemy attack will happen and likely hit the player, if the enemy doesn't die before that. There is also a short pause after the third hit of the player's attack combo where they cannot attack, preventing the player from button smashing their way through enemies by infinitely staggering them.

Apart the melee attacks, the player unlocks various guns that they can use to deal damage from afar throughout the game. However, each gun only holds a small amount of ammo, and the only way to refresh this ammo is to attack enemies or destroy various objects that can be found in the levels. Shooting is both safer and more effective way to deal damage than melee attacks, but the recharging design forces the players to still mostly use melee attacks, making the decision when to use the ranged weapon more interesting as it can only be used scarcely. Another limiting factor with the guns is that the player must stand completely still to aim and shoot them. Against the fast enemies of the game, this further pushes the player to carefully choose the moments they decide to use their limited amount of ammo.

For their defensive options, the player has a healing ability that fully restores their health and can hold certain number of charges. More charges can be found in levels, and using the heal stops the character momentarily, forcing the player to find a good moment for using it in combat situations, like with the guns. If the player has started the fight with low health, it might also be beneficial to try to reach a health package from the level in the middle of the fight.

The player also has a dash that can go to 8 directions and has a short cooldown. It is possible to use 3 consecutive dashes after which there is a slightly longer cooldown to prevent the player from using the move without any thought to avoid damage. The dash is quite powerful defensive move, as it can go through enemies and projectiles without harm while repositioning the player. It will however not protect the player from melee enemies in attack animation or area

hazards on the ground. The dash is also used to cross holes in the ground, and many of the levels have many such holes to make the fights more difficult but simultaneously give the player more options how to outmaneuver the enemies.

All these mechanics demand highly skillful play from the player. This design direction is further solidified by the health values set to enemies and the player. Both the player character and enemy characters have very low health pools: the player has five hit points throughout the whole game, and most enemies have two or three. Majority of enemies deal only 1 hit per attack, but there are many of them attacking the player at once most of the time. The player must identify the right moments to go on the offensive on priority targets, take them out quickly and then dash back to safety or towards new enemies should the chance present itself. The enemies can be killed fast, but mistakes of the player are punished in equally swift manner. This makes the combat full of interesting decisions, weighing risk versus reward.

The challenge and number of decisions could quickly become overbearing, yet the game is extremely clear in its communication, made possible by the mechanics being so simple. Each enemy has clearly visualized health in exactly how many hits they take to defeat, allowing the player to make informed decisions. The enemies also have clearly identified attack animations, and simple designs in general.

One particularly interesting thing to note from the enemies is how they position themselves. Instead of all the enemies running directly towards the player, there is a delightful amount of variation in how they act. Some ranged enemies stand completely still to the end of the combat, whereas enemies that do chase the player often include periodic random movements to different directions rather than moving in the most direct line towards the player. As each enemy moves a bit differently and with a hint of randomness, the battles feel alive and more interesting.

The simple enemies become much more interesting with use of interesting level design. As the player plays the game, they go through various rooms where combat occurs. There are both larger open spaces, as well as tightly packed rooms that the player cannot leave once combat begins (Figure 2). Sometimes there are just few enemies presented at once, sometimes more will spawn in waves once the current enemies are destroyed. Variation in the enemy setups as well as the layout, size and possible hazards of each combat room efficiently create different experiences from the relatively small set of enemies without any random generation involved.



Figure 2. A small combat room that the player cannot leave before the fight is over

In the end, all of these factors lead the core gameplay to the specific experience the developers had in mind: fast paced combat where the player has to make choices how to use their limited tools constantly and mistakes are punished harshly, yet a skilled player can feel in control of the situation. The player can buy some upgrades to their character over the course of the game, but most of these are tied clearly into the core gameplay mechanisms. The game shows that a game doesn't need many different game mechanics and systems to create a great experience, but rather focus on honing a few simple mechanics to the maximum.

5.2 Hades

Hades (2020) is a very fast paced roguelike game with isometric view, also critically acclaimed and praised by players like Hyper Light Drifter. The game clearly fits to the action and hack and slash genres and has been made by Supergiant Games with official launch in 2020. While the game is roguelike which makes it vastly different from the game being worked on for this thesis, it might provide useful contrast to the previous analysis by seeing what decisions were made and why.

In Hades (2020), the player must fight their way through multiple different areas, going through various random rooms one by one and ending in boss fights that stay the same. Each room is filled with various enemies and often include different traps or other hazards, and the player has to defeat all the enemies to move forwards to the next room. Get through all areas or not, after the player finishes a run, the game repeats from the start.

As a roguelike game, a lot of variance has been added into the game's combat system to keep recurring playthroughs fresh for hours on end. Each run begins with the choice of a weapon, which the player gradually unlocks more of. There are 6 different types of weapons, and each has 4 forms with their own effects, providing many different options to the player. During the runs themselves the player keeps collecting various upgrades called boons from a large pool of options, which mostly alter the basic actions the player has available. Getting a good set of synergizing boons will allow the player to become extremely powerful and defeat enemies in seconds, but of course such combination won't be achieved every run.

While each weapon is different, most simply modify the three base offensive actions the player has: a fast normal attack that can be chained with the later attacks of combo being more powerful or having different shape, a special attack which varies weapon by weapon, and sometimes the cast action that is a ranged attack shooting a single projectile. The biggest differences are between the types,

such as the bow being a ranged weapon, whereas the different forms of the same type have smaller differences.

The player's attacks can infinitely stagger most enemies, not allowing them to attack back. This helps the player control the situations, as enemies are often arriving in large hordes wave after wave. However, there is also a mechanic that prevents this approach, as sometimes enemies have armor effect on them, signified by yellow outline and second bar over their healthbar as shown in Figure 3. It provides a shield before the enemy can take damage to its health, and prevents them from being staggered until it is depleted. This mechanic forces the player to rethink their approach from time to time by being more careful around the armored enemies.



Figure 3. Enemies on the right have armor on them

As one major goal of the game is to constantly get stronger both between runs and within runs themselves, the game uses a granular health system where enemies have health bars shown without being able to determine the exact numbers of health left. The player's health on the other hand is shown as a number as well as a bar. Enemies generally don't deal that much damage per attack, but a careless player will end up taking a lot of damage from many attacks at once; by default, there is no effect that grants momentary invulnerability after

taking damage, but the player also doesn't get staggered themselves by taking damage, keeping the combat fast and fluid without much interruptions. It makes sense for the basic enemies to not have high damage, as the game is a roguelike and the health is a resource the player should be able to make last long as there is a limited amount of healing available during a run.

Like in *Hyper Light Drifter* (2016), in *Hades* (2020) the player has a dash option. The player's dash is short but very fast and ignores any damage that would hit them during its duration. The player can also gain more dashes from upgrades to chain multiple in a row before the cooldown begins. As the fights often end up full of area attacks and the enemies chase the player, dashing constantly is needed for survival. Dashing has a lot of offensive potential as well, since attacking an enemy from behind can deal increased damage and dashing is the way to position yourself for that. The dash can also be used to during other attack animations to reposition during them instead of entirely canceling them. This is yet another design decision that helps keep the combat extremely fluid and fast, as the player will very rarely interrupt their own actions they wanted to take by wanting to perform the dash to get out of danger.

The levels have various elements the player can take advantage of. Any traps that exist damage enemies just like the player, pushing enemies into walls hurts them and so on. The game does not however have the possibility of falling out of platforms for either players or enemies. This makes sense, as there is so much going on in the game falling off would happen very easily and interrupt the flow of the game. Traps on the floor fulfill much of the same purpose while not requiring designing what happens if the player falls out of the level.

Like in *Hyper Light Drifter* (2016), the enemies themselves are very simple with one or two different attacks they do over and over. Alone they are rarely a threat at all to the player, but the levels are always packed with multiple enemies spawning in waves after the previous wave is defeated. Most of the enemies have quite clear windup to their attacks, although it can become hard to keep

track of them when the screen gets completely filled with enemies, projectiles and the effects from players abilities.

Similarly to Hyper Light Drifter (2016), the player's approach to combat likely is to find opportune moments to attack, keep on the offensive as long as they can and then disengage when it gets dangerous. The difference is the player has a lot more agency and can go very aggressive, as they can control enemies easier and taking a hit every now and then is much less risky. Due to the roguelike nature of the game, the player can become extremely strong with the updates and not have to spend that much attention to completely destroy the enemies, at least until they run into armored enemies.

From these mechanics we can see Hades (2020) was designed with fluidity and high speed of combat in mind. The action in fights is non-stop and chaotic, yet the player character is strong enough to still be in control of the situation most of the time. Where Hyper Light Drifter (2016) required very deliberate and carefully calculated moves, Hades allows the player to get away with more button smashing, as long as they keep enough attention in their surroundings to dash away from enemy attacks when needed.

5.3 Analysis conclusion

While the two analyzed games both belong to the action and hack and slash genres and even share similar core game loops, they ended up having entirely different player experiences. Hades encourages the player into very hectic and daring maneuvers, unleashing all abilities they have at their disposal as fast and effectively as they can to quickly take out enemies. Hyper Light Drifter on the other hand is the exact opposite, forcing the player to be very careful with their every decision in combat as the enemies will kill the player just as quick as player kills them and the players resources are very limited.

For both games, these intended player experiences were clearly considered behind all the design decisions. This is likely precisely why both games have

been regarded as having great gameplay. Thus, it can be concluded that one very important element of creating engaging gameplay is having a well-defined player experience and basing all design decisions around it.

6 LOW FIDELITY PROTOTYPING THE GAME IDEA

As established earlier in this paper, the prototyping process should be started from a low fidelity approach and the process should be slowly add more mechanics and testing each of them until the game can be verified to work and move on to digital stage. However, the game idea being worked on is also very reliant on the game feel, which can only be tested digitally. As such, only the key combat mechanics that can intuitively be tested without a digital prototype will be done in this phase. The early prototype should mostly explore if the kind of decisions the player would need to do are interesting.

6.1 Tabletop simulator as the low fidelity prototyping method

By the writing time of this thesis, the world is still largely affected by the Covid-19 pandemic. Due to the circumstances, playtesting a physical prototype could prove difficult. Instead, the physical prototyping part will be made in Tabletop Simulator. Tabletop Simulator is a program that emulates physical components, mostly used for playing various board games digitally. The platform is excellent and fast to use for making quick prototypes and can be used to test with other players who own the software. As such, the early prototyping will be done in tabletop simulator instead of using actual physical components. The experience should be mostly the same as physical prototyping and making changes will still be fast as mostly only rules need to be changed.

6.2 Drafting and testing the base mechanics

As a guide for the whole process, a flowchart was created to present the core gameplay loop and flow of player actions (Figure 4). The focus was on identifying the flow of actions the player must take and what are all the situations where

player will use them. As can be seen on the chart, the main grappling hook mechanic fits well into overall flow of combat with having multiple uses, which seems like a good sign for the mechanic's potential.

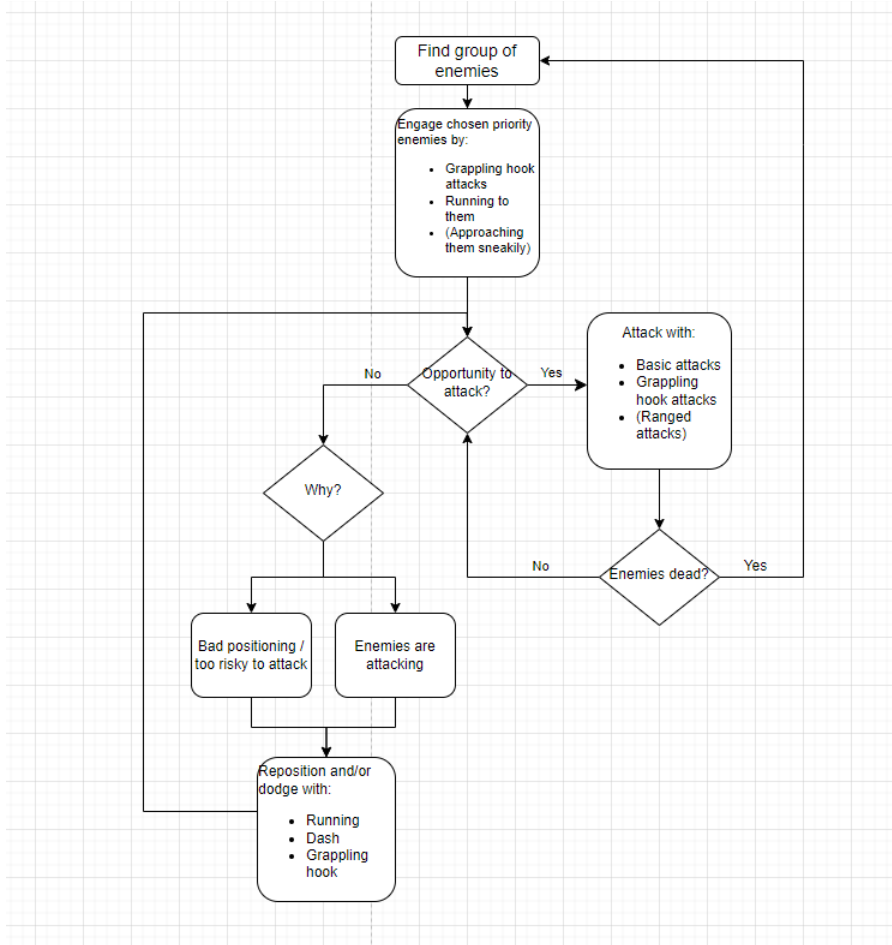


Figure 4. Flowchart for the core game loop of the game idea

As the game idea is reliant on interaction between the player and enemies, the first prototype already needs a decent number of rules to get any intended results out of it. There needs to be a player character who can move, enemies who have their own way to act and a way to kill those enemies. On less detailed level the core game loop should be similar to what was observed in the gameplay of the two analyzed games: the player tries to find an opportunity to attack, attacks the enemies when opportunity is found and after the time window of attacking is over, searches for the next opportunity while avoiding taking damage as much as possible at the same time.

As the game will have to include movement, a hexagonal board was first created out of a few pieces for characters to move in (Figure 5). Some movement blocking terrain was also immediately added, as it will be needed eventually either way for the grappling hook gameplay. A hexagonal grid was deemed the superior choice over squares, since distances a hexagon has to its neighbors are all the same. This creates more accurate portrayal of real time movement and range counting than squares.

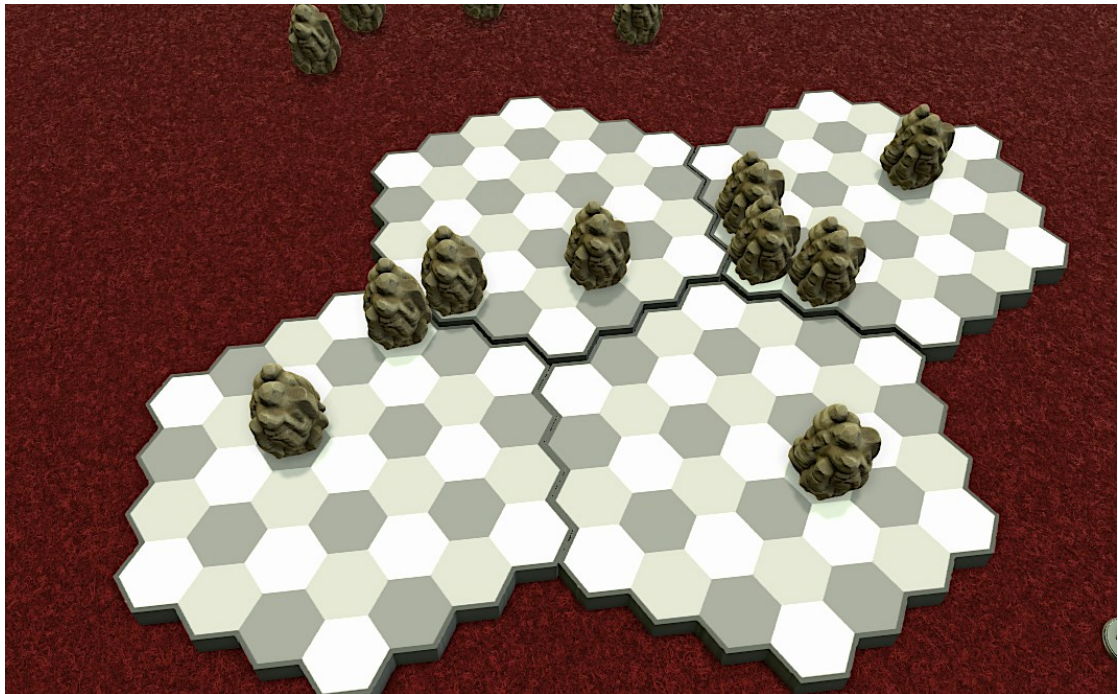


Figure 5. Hexagonal board with obstacles created in tabletop simulator

Some figurines were used to present the player character and enemies. Player characters health would be tracked with a counter, whereas enemies will have the health values written on information box that is visible when the figurine is right clicked in the program.

To keep the health numbers simple and easy to understand, they were kept very low at this stage, as it might even be a worthwhile approach to consider for the whole game as it worked well in *Hyper Light Drifter* (2016) as was analyzed. The player maximum health was set at 7, whereas the first enemies health at 5. First combat encounter was decided to be tested with 3 enemies at first.

For the enemies and player to act, turn structure was needed. While simultaneous actions could more closely match real time nature of the final intended product, it would become very difficult to track multiple characters actions on a step-by-step basis. Thus, a simpler turn structure was used as an abstraction of the real time combat. The player character would take a turn, and then all the enemies would activate, and then the player goes again and so on. The enemies activate in order, with those closest to the player acting first. Enemies at the same distance would activate simultaneously, which could matter for a dodge mechanic.

Since the enemies would be very simple, rules for what they do during their turn were designed first. If the enemy isn't next to the player, they move up to 3 tiles closer. If the enemy was adjacent to the player to begin with or ends next to the player with a move, they would try to attack the player for 1 damage. If the player runs out of health, the fight ends in a loss. Simulating varying paths would be needless detail at this point, so when 2 hexes would both take the enemy character closer to the player, one is arbitrarily chosen. As the enemies could easily end up stacking to same hex while choosing the closest path to the player and that would be a problem, a rule was decided for each hexagon to only fit 1 character and passing through other characters to be impossible. This of course also applies to the player.

Then, the player's methods for moving and attacking were created. To simulate fluidly being able take different actions in combination, the player was given 4 actions to use per turn. Three basic actions of moving, dashing, and attacking were added first (Table 1). It felt necessary to already allow the dash to be used to dodge enemy attacks, as otherwise there would be no defensive maneuvering at all for the player.

Table 1. Initial player actions for the tabletop simulator prototype

Action name	Effect	Action point cost
Move	Move 1 tile	1
Dash	Move 2 tiles. Can be used once in a round. Can be used when enemy attacks would land to dodge the damage instead.	1
Attack	Choose 2 adjacent hexagons within 1 range and deal 1 damage to enemies in them. If this is the last action this turn, those enemies are stunned.	1

To make the attack action simulate the kind of gameplay where the player can control multiple enemies by staggering them, two more rules were added to it. Firstly, the player's attack action would hit any two hexagons within 1 range that are adjacent to each other. Additionally, if the last action the player takes on their turn is the attack, the enemies hit by that attack are stunned and won't act on their next turn but cannot be stunned again until they have taken one turn where they are able to act. This simulates the kind of mechanic that was observed in Hyper Light Drifter, where the player's attacks build up a combo and at the end of it there is a pause before attacking can resume.

While this was already quite a few rules for the first test, it felt necessary to add at least this many mechanics as a base to get any interaction out of the prototype. As the mechanics so far were simple and very similar to established gameplay of the genre, there was not much that could go wrong yet.

Fighting two enemies, one starting at range 4 and one at range 3, were tested first. The combat ended very quickly in an easy victory with one health point lost,

as the enemies could easily be maneuvered in a spot where they can be both attacked at once. Further tests with the same ruleset were made with 3, 4 and 5 enemies with varying compositions. Taking your time and carefully planning the moves made it possible to survive with full health quite easily before the jump to 5 enemies. So far, the prototype felt like a somewhat accurate abstraction of how the combat would work in the final game, with the largest flaw being the lack of motion at times due to the turn-based mechanics.

6.3 Adding the grappling hook mechanic and ranged enemies

The next phase was adding the grappling hook mechanic to the game to see how it works out. This was done one step at a time like mechanics before to see how the interactions with the other systems work.

The first addition was simply allowing using the grappling hook to move around more effectively. The player was given a new action where they could use 1-2 actions to move 3 hexagons for each action used. The target hexagon to travel to must be next to a target enemy or obstacle with no other things in the way. All places where the hexagon grid ends were also considered as obstacles, so the player could drag themselves to the outer walls of the play area.

As discussed earlier, each prototype and testing round should try to pose a question. The question for this iteration could be worded as “what effect does the increased mobility have on the gameplay”. The effect observed was that it was easier to do proactive defensive moves by getting far away from the enemies, rather than waiting for the enemy turn and then using the dash to dodge attacks. However, the range of the grappling hook felt like it needed to be longer, as in practice the one action version only traveled two spaces making it same as the dash, while it should be a longer distance move in comparison. After increasing the movement value by one for each action point used the ability started to feel correct.

To give more use cases for the increased mobility, next a ranged enemy was added. Having a mix of melee and ranged enemies available should already greatly increase the variety of different situations the player has to work with. It was also decided that the attacks of the ranged enemies would travel on board instead of instantly trying to hit the player, as it can be assumed in the digital version this would be the preferable gameplay. If the projectiles have some travel time, it will create different safe and dangerous zones for the player to maneuver around and not feel unfair to defend against.

For their behavior, ranged enemies were made to shoot towards the player if they are within 6 range and there is a clear line to them without obstacles. Other enemies did not count as obstacles. The ranged enemies would additionally move away after attacking if the player is within 2 range or closer to them. Their attacks place the projectile on the board as a token, and it will travel 2 steps immediately and 1 step after each action the player takes, damaging them if they get hit. The steps will roughly be estimated as 1 hexagon distance, as sometimes they will not travel in a straight line through hexagons and make the measurement harder. The ranged enemy was also given one less health and movement to give some balance cost to its ability to attack from safety.

During first test with 3 ranged enemies, it was clear the projectiles already on board from previous enemy turns should also move 2 forwards when the enemies take their turn again, as the projectiles shouldn't stop at any point. The ranged enemies were observed to immediately make the player approach the combat in a different way, weighing options of taking out ranged enemies first.

Next, a mix of ranged and melee enemies was tested (Figure 6). One thing that immediately stood out with that test was that there were many little things the rules didn't cover yet, such as when the enemy projectiles overlap 2 hexagons, would they hit the player if they are in either of them. For now, the answer was marked as yes, and all such cases were written down to be utilized later in making clear rules for other playtesters.

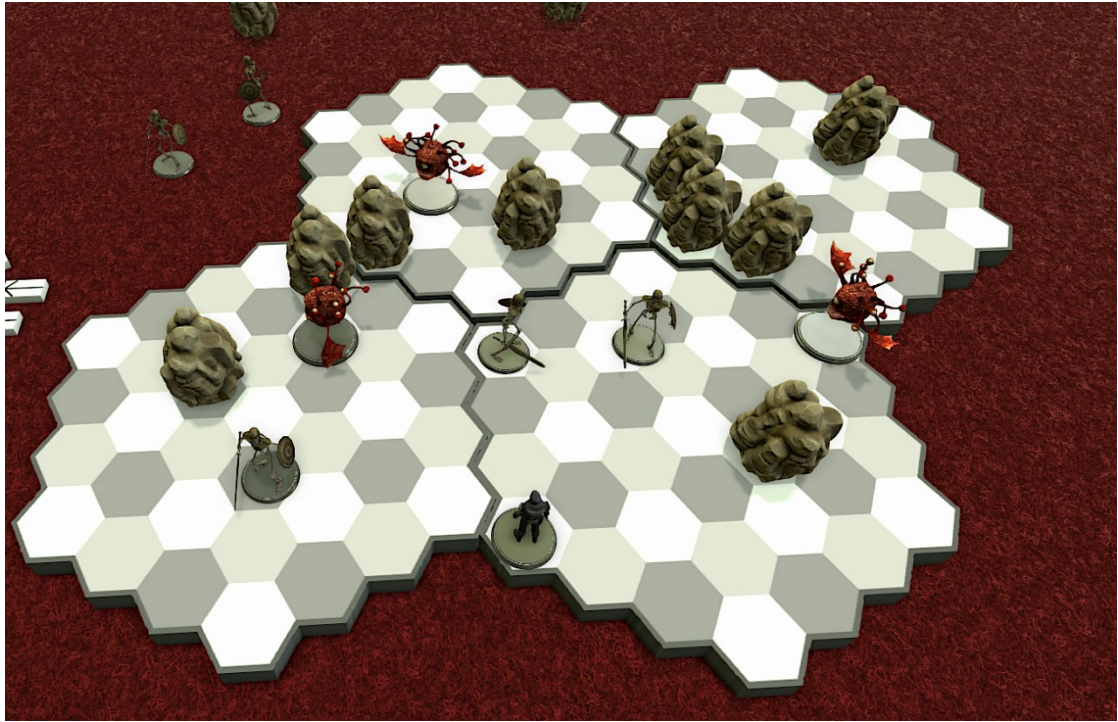


Figure 6. Playtest initial setup with 3 melee enemies and 3 ranged enemies

This test also took longer and started to push the limits of manually doing so many different steps to simulate real time mechanics. In general, the ranged enemies seemed to be an interesting addition and work as expected with these initial values, with their projectiles creating temporary obstacles to maneuver around on the board.

One planned mechanic for the grappling hook was that the player could latch onto the targeted object and rotate around it, instead of pulling themselves to it immediately. This type of motion however would be very difficult to portray accurately in the board game style prototype and was left to the digital version. Thus, the only mechanic missing from the grappling hook was the ability to use it for attacks.

During the previous tests, it already seemed quite clear how this mechanic could work. At this point there wasn't a clear differentiation in mind for a non-attacking grappling hook action and one used just for movement, as that is a level of detail that will become clear further into the development. As such, it seemed easy to simply test a design where any grappling hook action is an attacking one. While

there would be some slightly unclear cases when the players movement isn't aligned to the hexagons, the move dealing 1 damage to each enemy within 1 hex range during the travel seemed like a good implementation to test. Characters situated behind the player character on start of the move won't be hit, as it would become very easy to deal damage while simultaneously escaping, which could easily become a dominant and boring strategy to utilize.

During the testing it became clear the attack shouldn't have the same stun effect as the basic attack, as it would be easy to prevent numerous enemies in a single turn from acting while also dealing damage to them simultaneously. Overall, the addition made the gameplay much faster and easier which was expected. By effectively being able to avoid damage and deal it on the same move, the grappling hook became a very powerful tool. It became possible to fairly easily avoid all damage in the current setup of the test.

This was acceptable at this phase, as it is meant in the game idea for the player to be powerful and good play should avoid nearly all damage. Limiting the grappling hook attack to twice per turn was added, to avoid the player too easily running away from all enemies; in the digital version, some kind of overheating system could be tested instead, punishing players if they overuse the move. It was also tested if the player could hit 3 hexagons in front of them instead of 2, but that change was deemed to give the player too much power against enemy groups consisting mostly of melee range enemies.

Throughout all tests so far it was clear that the initial positions and number of different types of enemies mattered a lot. Fighting only melee enemies easily stacked them together especially if they started from the same direction, making disposing of them easy. More split up initial placement and including both types of enemies created situations with more interesting decisions.

6.4 Playtesting the low fidelity prototype with other players

As mentioned earlier in this paper, prototypes and playtests should have a clear purpose and questions they try to answer (Fullerton 2014). In this game idea's case, the physical prototype cannot very accurately portray the fast paced, real-time combat that is the goal of the final product. As such, the purpose of doing playtesting with the physical prototype was mostly to see if any clear flaws can be found in the basic structure of combat where you try to control the groups to your favor and then attack on opportune moments. By seeing how players will interact with this prototype, correct design decisions should be easier to make for the digital prototype, hopefully leading to shorter development time. So, the main question the testing tries to answer could be worded as "does the gameplay follow the intended game experience".

Tabletop simulator gives some perks to testing. Since testing is done in a digital environment, recording the sessions is very easy. In this particular case, it will also be possible for the developer to "play" the enemies, as if they were being controlled by artificial intelligence like they will be in the final game. This will help with the player experience being closer to the digital end goal, as the player can't cheat by moving the enemies to more beneficial positions and don't have to burden themselves with the task of doing the enemy moves every turn. They can instead just focus on their objective of dispatching the enemies effectively while trying to keep themselves from taking too much damage.

To give players better information for the test, few rule notes were created in the test environment itself (Figure 7). Enemies were changed into tokens that directly show the health and have buttons to reduce it, to make it easier to assess the overall situation. The blocking walls were also added around the arena now, to visualize you can in fact use the grappling hook to the edges of the play area instead of an abstract rule. Three separate maps were made with different enemy positions and shapes of the map itself, starting with assumedly easier setups and getting harder. The different variations should give more information how the layout of the level affects the gameplay as well.

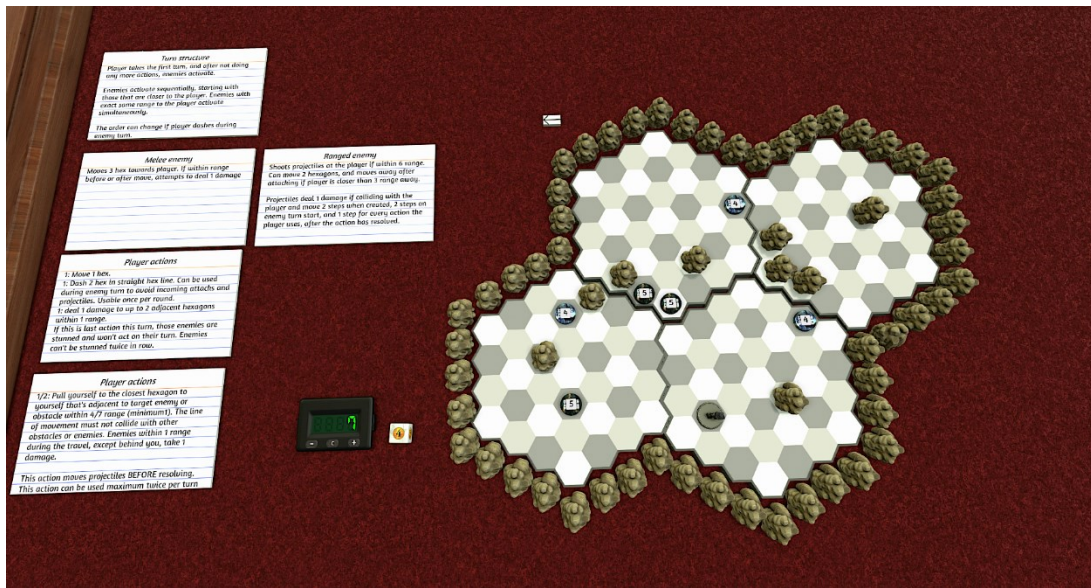


Figure 7. The final look of the testing environment, with the second level opened

The testing was done with 4 separate players, 3 of which had experience in game design which should help gain useful feedback. They were told what they need to know to be able to play the game and the purpose of the prototype and then observed playing, while also being encouraged to speak out their thoughts during the session. The sessions were recorded so that information won't be missed. At this early point of development, it was assumed most vital information would come just from observing how the players act. As such requested feedback was kept simple. After finishing the 3 levels, the players were asked these questions, focusing on their feeling of the game:

- Did your character feel powerful?
- Was there something that felt particularly good or satisfying?
- Was there something that felt unfair or particularly annoying?

Afterwards, they were also asked to say anything else they might still have in their mind.

Testing of the physical prototype was definitely a success: the core game loop worked, and the resulting gameplay was even perceived as fun, which was somewhat unexpected at this point. The players ended up focusing on maneuvering the enemies into opportune positions, and then went heavy on the

offensive, enjoying the moves they could pull off with the grappling hook and the amount of control provided by the options they had. Various interesting tactics were performed by using the grappling hook for movement in and out of combat, and it was easy to imagine those moments being very cool in the final game. This was all wished-for behavior, and the players naturally gravitated towards acting that way.

Various playstyles emerged. One of the players focused heavily on timing their dash perfectly to gain the edge by avoiding multiple attacks at once while another one in fact forgot they could use the dash to dodge enemy attacks on their turn entirely. Instead, they made up for that by making more proactive movements on their own turn and using the stun very efficiently. Some players were more careful with their health, while others gladly took some damage if it meant they could have a very effective turn themselves. This seemed to show that the core gameplay has enough depth to allow interesting choices to be made, although the physical, turn based prototype might have made these decisions weightier than they would be in the fast-paced digital game. There were clear interesting benefits to both being proactively aggressive by trying to stun multiple enemies and being reactive with the dash during enemy turn instead.

Two of the tests ended with the player having just one health left in the last tested level, yet they both still managed to pull off the victory. These were exciting moments, and the game still allowed for these players to be quite aggressive even though they couldn't afford to take any more damage. Carrying that feeling of skillful play being able to overcome such dangerous situations to the digital prototype is a good goal to aim for.

The three different setups of enemies and terrain placement also had noticeable effects on the gameplay. When the enemies were more spread out, the players had considerably easier time to remove some threats from the board before they are in danger themselves by the other enemies converging on them. But on the other hand, rushing to a group of enemies immediately was sometimes a viable option, to quickly dispatch two enemies within first two turns. The number of

ranged enemies also clearly changed the flow of combat. The more ranged enemies were present, the more the projectiles appeared on the board, limiting the amount of space the player could maneuver in without getting damaged. It was possible to either try to take the ranged enemies out first to avoid this or run away from them as they moved slower than the melee enemies.

For the effect of the terrain, in the map with more blocked space everywhere it was easier to get the enemies to clog the passageways and group together to create opportunities of attacking many enemies at once. However, it was also harder to line up the grappling hook moves with more terrain around. Since the enemies blocked the players movement, it would be possible to get stuck inside a zone, so the players had to be more careful with where they moved to each turn.

Most of the frustrations of the players could be appointed to the limitations of the physical prototype. This included the way the enemies behaved being a bit weird at times, the way the arrows were finicky to simulate and how checking clear line of sight for the grappling hook was difficult. This was a great result, as no actual clear flaws were found in the expected gameplay that would carry over to the digital prototype.

After testing the prototype with the four separate players, it felt like it had provided enough information to warrant moving to the digital stage. Only changes made between different players during the process were rules clarifications, as the gameplay ended up working as is and fit the intended player experience. There are various systems that could have been added as they likely will be present in the digital prototype but simulating things like line of sight or the rotating grappling hook move would have required far too much complexity to be added to be worth testing in the physical format.

7 DIGITAL PROTOTYPING PROCESS

For the next part of the process, the prototype was created in digital format in Construct 3 game engine. For the sake of keeping the thesis focused and on the

topic, this chapter will not delve deep into how exactly things were accomplished technically, but rather focuses on the important design decisions required. The end goal was to create at the same mechanics as in the physical prototype and add the rotational grappling hook move.

7.1 Creating the basic movement and combat mechanics

To begin the process, the player character and its basic movement had to be created. Even though the game would have only top-down movement, there are still many attributes to tweak. Should the character be quick to accelerate or take time to reach their top speed? Do they have deceleration period or stop immediately? Since the intended game experience expects the character to be fast and powerful, they were given quite fast base speed and almost no acceleration and deceleration required at all. This makes the character easier to control precisely.

Something to immediately decide in this part was what controller to think about while creating the character controls. Gamepads are often preferred by players of these types of games due to their greater comfort than playing on a keyboard. In this case, keyboard controls were however chosen as the initial controls to think about. This is both because it should feel like a natural fit for aiming the grappling hooks with the mouse when implemented, and that to test the prototype with as many players as possible it is good to not require a gamepad yet as not everyone owns one. In the future, should the game project become serious, both forms of control would be a good idea to support.

The next step was to create the dash mechanic, which is a staple defensive maneuver in many games. The concept is simple: the character should move a set distance faster than usual. To make using the dash an interesting choice, it should have a short period of time it cannot be used afterwards, to prevent the player from using it over and over, replacing basic movement entirely.

The values for the movement are also important for the game feel. How far should the dash travel and at what speed? Does the dash have acceleration or

deceleration? Various values and methods were tried. The initial dash landed on a short cooldown of 0.6 seconds and was a brief but fast burst of speed that decelerates at the end. During the dash the player's normal movement input is ignored, and it returns just as the dash is about to finish. As soon as the dash felt good enough, rest of the tuning was left for later; it would be ineffective to work too much on the numbers before an actual combat situation is possible and provides context for the dashing mechanic's purpose.

The basic melee attack was the next thing to do. The character would simply attack to chosen direction, visualized with a simple slash visual effect that was created. The slash hitbox was made to be about 120 angle cone dealing 1 damage, being able to hit multiple enemies as that is essential for the wanted gameplay of controlling hordes of enemies by well-timed attacks.

To test the attack, the first basic enemy had to be created. Free to use sprites were found from the internet to properly show what the enemies do without wasting any time creating custom animations for the prototype. The implementation was very basic for this phase. The enemy was made to move towards the player if they can see them, and if within range of attack, play an attack animation which deals damage on certain frame. Having a placeholder character with animation was essential at this phase to make the attack possible to react to. Some target dummy enemies who do not fight the player were also created to make it possible to test the mechanics without actual enemy interference as well.

Making the enemy movement feel good was quite tricky and took some iterations. Having all enemies move directly towards player looks and feels very unnatural, and it felt necessary to limit how many enemies are actively trying to get next to the player. To solve this problem, the 9 closest enemies were made to choose different spots around the player to move towards. Enemies beyond this number would instead do small, random movements set distance from the player, waiting for the player to either come closer to them or other enemies to die before

starting to chase the player properly. The enemies were also made to push each other slightly, so they would not overlap each other at a single point as easily.

The enemies were given five health initially and if it runs out, they are destroyed. A very basic line of sight behavior was created with full 360-degree coverage, making the enemies aggressive towards the player only when they first went close enough to them with no obstacles between. Line of sight behaviors could be elaborated later to allow the player some stealth options should the game become a fully developed experience.

Now with an enemy created, next details added to the slash were making the character unable to move during the attack but during it moving them slightly forward as well as turning it into a combo with set amount of hits, like was observed in Hyper Light Drifter. The maximum combo was set to 4 for now. These decisions force the player to think more when they attack as they cannot move around the enemies freely while attacking and cannot keep attacking without a pause after every fourth hit.

The last mechanic to the slash was the staggering. Enemies would get slightly pushed back by the attack, and become staggered, which stops them from acting until they recover. With this mechanic the player should be able to control groups of enemies with ease with correct positioning and timing, but the recovery would be set to a short enough time where the player must be careful at the end of their attack chain. For fast to create but efficient feedback to the player, the enemies were given a stretchy effect when hit.

The enemies having animations made the unanimated player feel very clunky in comparison, so a free to use 2D character rendered from one direction was found and added for the player (Figure 8). A character rendered in 8 directions would have been ideal as the game is pictured from top down, but no suitable free characters were found for that. The combination of the 2D sprite with one direction with the previously made slash visual however ended up feeling

surprisingly good. The same character also had animations for idle, run and slide which were used.



Figure 8. Player character sprite for the prototype, created by username rvros on itch.io

As very basics of combat were now ready, some testing was conducted. Combat against as many as 12 enemies at once was attempted and it was still easy for the player to defeat them. Perhaps a bit too easy already, in fact. One reason for this was that currently all the enemies were attempting to get close to the player and chose the closest direction, meaning they would very easily become one big group to attack at once. Some experimentation was conducted for ways to make the opponents split up a bit more as a result. This however ended up being a complex effort that would take too long to be worth solving for this phase. In final version of the game idea, different enemies would likely have different ways of movement either way.

To punish the player for bad play, a mechanic was created where if the player gets hit by an enemy and they were not recently hit, they are interrupted momentarily, stopping any actions already playing and not allowing movement. With this addition, the basics of combat mechanics were done and worked as expected. The player could rather easily control the groups of enemies with good positioning but couldn't be completely careless. It was however evident at this point that for games of this genre, enemy behavior needs a lot of design and thoughtfully made systems to create good gameplay. While the enemy behavior wasn't at such level yet, it was deemed enough for this prototype.

7.2 Creating the basic grappling hook mechanic

With the basic systems working, it was time to add the main mechanic of the game: the grappling hook. The very basic implementation was very straightforward and easy. The player would use the right mouse button to shoot a rope that travels towards the mouse position over short time, and if it hits a solid object before the maximum range is passed, the player is pulled into it. Following the intended game experience, the move would have to be quite powerful. As such the range of the ability was made quite significant and the player would traverse the distance quickly as well.

For more control, the player can interrupt the move with a dash or another grappling hook action during the pull to the target. A minimum distance required for the rope was also added, as very short distance travels wouldn't feel as good and might cause balance issues when the move will be able to deal damage.

At first, a version was tried where the player would stop after initiating the action until the rope hits an object. However, it was quickly found out that the sudden pause didn't fit the flow of the intended game experience, and the player was allowed to move instead while waiting for the rope to hit. Attacking or dashing during this time was however not allowed to avoid both technical problems as well as unintended interactions.

When using the grappling hook to obstacles was working, it was straightforward to add the offensive functionality. The player would simply deal 1 damage to every enemy in front of them while traveling visualized by a blue attack sprite in front of them (Figure 9) and could now directly target enemies instead of only obstacles, causing them to stagger momentarily. The grappling hook attack was made to function similarly to normal attack, having a stagger and pushback. The pushback was made slightly stronger than the normal attacks to make the maneuver feel more powerful, but some tweaking was done to ensure it would not push enemies so far they couldn't be smoothly staggered with normal attacks afterwards.



Figure 9. Clear blue visuals shows where the player deals damage during the dash

Especially while testing targeting the grapple hook to the enemies, it was easy to notice that it was at times quite difficult to hit the wanted target. Some mechanics had to be implemented that would give the player some room for error instead of having to perfectly hit the wanted targets in middle of combat, while still allowing accuracy and skill expression.

A little help system was quickly created, where if the rope would end up missing an object near the mouse location by a set small amount, it would be directed to the center of that object instead. This worked surprisingly well immediately, although a different approach would have to be thought of for gamepads if they are introduced later as a control method as they can't rely on position of a mouse pointer.

From testing so far, it was also clear the grapple hook required some limitation, just like there was in the tabletop simulator version. Otherwise, the player wouldn't have interesting choices to make, as the best choice of action in any given situation would likely be to constantly use the grapple hook attack, move away with another grapple hook action and repeat. In the tabletop version, the limit was reducing the use to twice per turn. One way to turn this mechanic into

the real time combat would be to have charges required for grappling hook use, that then replenish one periodically.

However, a slightly more thematic approach was tried first. The grappling hook was made to add certain amount of overheat to a new meter that constantly goes down over time (Figure 10). Then, if the player would go past a set limit, the grappling hook would be disabled for a while entirely. This way, the player will have opportunities to both use the move more sparingly avoiding the overheating or to use multiple grappling hook moves in quick succession but then must survive without the ability for a while. The meter was also made visualize how many grappling hook actions the player can perform in row at the moment with separate visual arcs above it.



Figure 10. Meter showing the grappling hook overheating status

After some tests, allowing up to 3 grappling hook moves in a row felt like a good value good for now. The meter was also made to glow red when overheated. This way players could clearly see they cannot use the grappling hook move currently, without having to carefully observe the UI element.

7.3 Adding ranged enemy and respective tweaks

One last element missing from the digital game that wasn't there from the physical prototype was the ranged enemy, and that was implemented next. The behavior was made exactly as the physical prototype. The ranged characters

shoot projectiles that travel towards the player if they are within attack range, and after attacking they will attempt to move away from the player if they are close enough.

The projectile's traveling speed felt like the clearest balancing value to tweak, and it was experimented on quite a bit already at this point until it felt fair but still challenging to evade them with groups of ranged enemies attacking the player. It became evident the player's dash should have short invulnerability at this point, so that they could aggressively dodge attacks while moving towards the enemies. Another addition was allowing the player to dash to interrupt their own attack animations, making the dash feel more responsive for the fast-paced combat.

Besides the normal dash, it also felt quite difficult to use the grappling hook attack without being damaged by all the projectiles shot towards the player. Once again looking at the intended game experience stating that the player should feel powerful and be able to take on groups of enemies with relative ease, it was tested what would happen if the player was invulnerable to projectile attacks during the grappling hook move. The result did allow much more aggressive playstyle with the grappling hook action, which was wanted, and the change was kept.

Some other tweaks were still made to reduce frustration against the ranged enemies. It was made possible to interrupt them further into their attack animation than the melee enemies, and for making chasing them easier when they run away the player was made to move bit further with their attack. The speed the grappling hook rope travels with was also made faster to make it easier to reposition out of tough situations quickly, and similarly to the dash, it was made possible to use it to cancel other animations.

7.4 Adding rotational grappling hook mechanic

The rotational grappling hook move which was missing from the physical prototype would add a very different movement mechanism to the game. In that

sense, it would make sense to first playtest the digital prototype with the same mechanics as the physical game with other players, and only then add the rotational grappling hook mechanic to see how exactly it changes the experience. However, the rotational move was decided to be included in the prototype already for the first playtest, as doing 2 separate sessions of testing for the digital prototype would be too much to cover in this thesis.

Basic design for the mechanic was quite simple again. The player could shoot the grappling hook with a modifier key down, and then will start to rotate around the grappling hook target in set direction, with the rope constantly getting shorter. Player's movement direction is detected when deciding will the rotation go clockwise or counterclockwise, to keep the controls as natural and easy as possible, forming the kind of natural mapping mentioned earlier in the game feel chapter. The move ends when the player performs any action or when the rope gets too short. Since enemies are constantly moving and preventing them from moving for a long period of time makes no thematic sense, this move was only allowed to target obstacles.

Speed of the rotating grappling hook move was adjusted to depend on how long the rope is until it felt good, and the move did start to feel fun very fast. Another detail added was that the player will fly in a straight line for set time after letting go of the rope, instead of all the momentum immediately stopping which would feel very unnatural. The duration of this flight was also made dependent on the length of the rope. The player would be stuck in this flight no matter what, having to wait for their landing.

Many other questions appeared with this mechanic, such as what happens if the player hits obstacles or enemies with the rope. Hitting other obstacles was made to end the action similarly to player ending it themselves, whereas for now enemies were simply not interacted with at all. If hitting enemies would break the rope, the move could quickly become quite difficult to utilize at all. The move was also made to constantly increase the overheat meter as player holds it down, to

require some more strategic thought for how long the player should hold on to the motion.

Like the straight grappling hook move, the rotating one was made to deal damage by spawning an attack next to the player. It was on the other hand not given the immunity to ranged attacks, as it is a move the player can keep going for a longer time and prolonged immunity that way might be too strong. This could be proven wrong in the testing of course, but for now it felt like the right move to not allow such long window of invulnerability.

The mechanic felt very satisfying to use in the test map with a lot of space between different obstacles. The player could use it to create more space very efficiently between them and the enemies following them, as well as smoothly follow the action with a straight grappling hook move. On a different test level where obstacles were more tightly packed to each other, the usage was more difficult as the rope often cut from colliding into another object. Some more tuning could perhaps be done there, but the mechanic was in good shape for the testing now.

7.5 Adding feedback and building the levels

Since real time games rely heavily in giving the player proper feedback for their actions to feel good, some time was used to add much more feedback to the player's actions. A steamy particle effect was added when using the grappling hook actions, and a blood visual effect was added when either the player or enemies were hit. The former was simply made from a particle drawn in a minute, whereas the blood effects were found from a free asset pack. With a very small amount of time spent, a big improvement was made particularly with the blood effect.

Second, even larger inclusion for feedback was adding audio. While making or even searching for perfectly suitable sounds would take long, just finding sounds that seem somewhat fitting was much faster. With perhaps two or three hours of searching and setting sound effects to the correct events, nearly every

imaginable situation had sounds added. Hearing a sound when enemies are hit or the player takes damage adds incredible amount of information clarity, and with a relatively small effort all of that was achieved. Most of the sounds were found from ZapSplat, which was known from previous projects to be a good website for free sound effects.

Now, the only thing left before playtesting could begin was to build some levels where the playtesters would play the game in. So far nearly all testing had happened in a haphazardly put together level with randomly spawned enemies and some obstacles sprinkled here and there. For the playtesting, maps should be made with manually placed enemy groups and more thoughtful obstacles, more clearly mimicking what the full game experience would be like. This would still of course not be as carefully designed as level design in a final game would be.

First a tutorial level without actual enemies was made (Figure 11). Here the player could test the mechanics of the game, with some obstacles to test the grappling hook on and target dummies to strike at. Relevant controls and game mechanic information were added as text on the ground as well. No actual enemies were placed in this level.

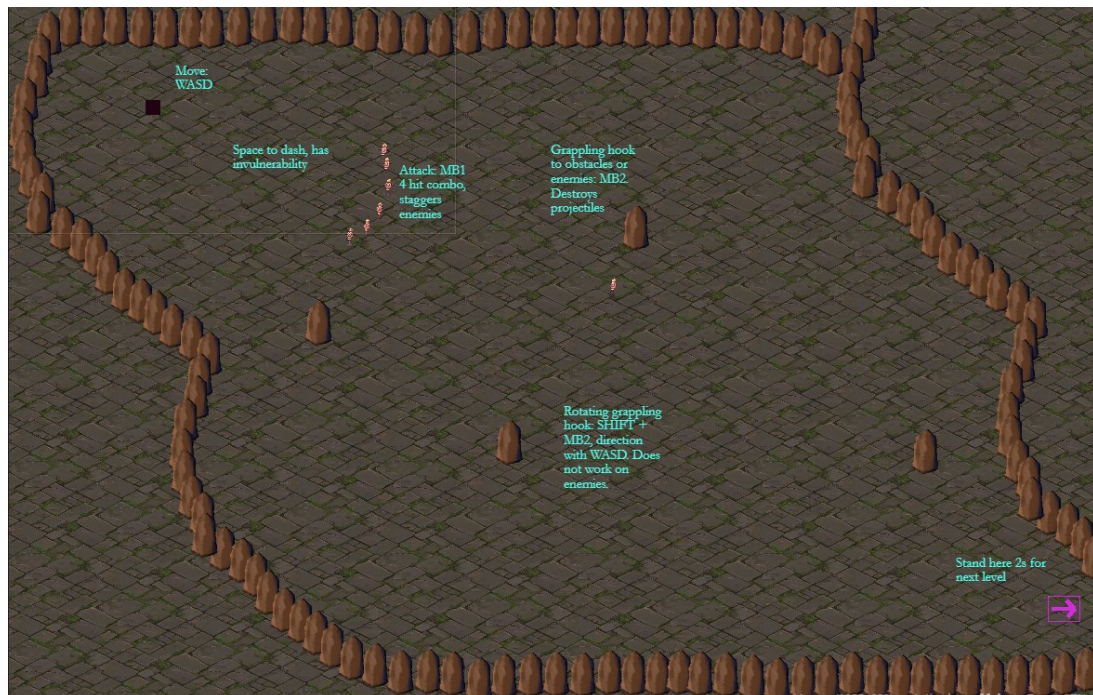


Figure 11. The tutorial map with game instructions clearly written on the ground

Next was creating levels where the player would fight the enemies. Three levels were created where both melee and ranged enemies were placed in small groups or completely separately. A higher health, slower moving melee enemy variation was also added. The levels had different amounts of obstacles set to see how playing differs in more open or dense spaces, with the last one having quite narrow pathways at times. These levels were still created fast without complex level design, as quite simple setups should be enough to see if the core gameplay works or not, and the enemies are not complex enough yet to create intricate level design as well.

There was one last set of tweaks made to the systems in general before testing. Enemies were made to get pushed slightly further if the player attacks them while right next to them to not have the player slide past the enemies while attacking as easily. Pushback values in general were slightly adjusted, player dash made slightly shorter, and enemies were made to alert others near them when they see the player so that they cannot be pulled to the player one by one with careful movement. Small tweaks like this still had a large impact to the game feel, which is something the playtests would try to evaluate.

7.6 Playtesting digital prototype with other players

Now that everything was ready, it was time to conduct the playtesting. First set of players amounted to four, with two players from the previous test and two completely unfamiliar with the game so far. Similarly to the physical test, minimum amount of information needed to play the game was explained and then the play was observed, with some questions in the end. Players were again asked to tell their thoughts during play, although it was harder to do with the game being so fast paced. The question the playtesting would try to answer is if the prototype had turned out engaging and fun to play

Throughout testing with these four players only small clarity changes were made, such as adding an additional tutorial text boxes and a minimum grappling hook range showing circle around the player (Figure 12). For example, one of the

players didn't realize they can attack to every direction, as they were looking at the sprite animation which only had two directions. As such more information about the attack was added to the tutorial level.



Figure 12. A circle around the player shows the minimum range required for the grappling hook move to not fail

Overall, the core gameplay seemed to achieve its goal. After getting used to the game, the players were mostly aggressive and daring, dashing from one group of enemies to another when the right opportunity presented itself and maneuvering the enemies to right positions. Most of the times when the players played the game in unexpected ways, like attacking only once and then dashing away repeatedly instead of attacking more confidently which stuns the enemies, it happened due to there being a lot to learn at once which shouldn't be a problem in a full game experience. It was clear to see such learning progress at times, where after the players figured out exactly how long they could stagger the enemies they fought more aggressively.

On the same note, the players often relied on the basics of the combat without the grappling hook first while getting used to it. This way they were still able to

very successfully play around the enemies, which was good. This means even without the grappling hook the combat system was robust enough and seemed enjoyable. However, for next round of testing, it would make sense to have one level without the grappling hook first and then give it to players after that, so they have less to learn at once, better mimicking full game experience.

When the players did use the grappling hook, they did feel it was powerful and fun tool to use when it worked as they expected. There were however some control issues with it, where it didn't always feel perfectly intuitive, especially for the rotating grappling hook. A good suggestion that came from the players was that you could use shift even after the rope has already been fired to initiate the grappling hook, to make the combat more responsive. Another issue was the terrain bit too easily interrupting the rotational grapple which can be confusing but would need a lot more design work to fix.

It was also evident that the enemies were very simple and bit too easy to outmaneuver, especially by hiding behind obstacles to make them into a big group. For example, the melee enemies all group up in a way where it feels like fighting a single enemy, even though they are multiple. More enemies with different variety and interesting behaviors would be needed for the full game to be interesting, but that was to be expected at this phase. The player's dash was also a bit too easy to use at least against these enemies, somewhat invalidating the need to use the grappling hook at times and encouraging too defensive playstyle compared to the intended experience.

The first question the players were asked after the testing was if they felt the character was fun and responsive to control, as that was big part of trying to make the core gameplay good. The answer to this one was very resounding yes from everyone. The mechanics allowed the character to be easily controlled for the most part, with only some occasions of the grappling hook usage feeling hard to control exactly as wanted. One player also found it annoying how the character can sometimes attack to unintended directions if the cursor was on top of the player character, even though enemies were only in one direction.

The second question asked was the combat overall satisfying. Keeping in mind the current state of the game being an early prototype, the players also agreed that this was the case. One player even wanted to continue playing longer to see if they could beat the levels without getting hit, clearly signifying there was fun to be had playing the game already. One player had some concerns that related to enemies again, as the melee enemies constantly moving towards the player didn't really leave much breathing space for the player during combat.

Two last questions were asking what was particularly fun and particularly annoying. Despite its slight difficulty of use, grappling hook was clearly the most fun part for everyone, which was good as it was the expected result. The issues in its use were mentioned for the particularly annoying problem, as well as the way the enemies behaved due to their simplicity for the prototype. Just like with the physical prototype phase, it seems the problems found were those that were expected due to the prototype stage of the game and no alarming issues were found in the game loop.

After feedback from these four players, changes were made to address some of the easy to fix concerns. The rotational grappling hook was made more reliable to use by allowing it to be queued with shift at any part before the rope hits and was given the same projectile immunity as normal grappling hook had. The levels were split into playing the first actual level without the grappling hook, and then introducing it for the next two levels to ease the learning curve at least a bit. For this reason, some enemies were removed from the first level to make it easier as well. Dashing around to avoid damage was also made slightly harder by a longer cooldown, hopefully encouraging the aggressive grappling hook heavy play style that should provide the most fun experience to the players. Some bugfixes were also done, mostly on the enemy AI.

Three more players were asked to playtest the prototype after these changes. Separating the mechanics into two learning sessions helped, although it was a still a bit much to learn the game at times especially for the rotational grappling

hook move. On the other hand, all of these players found and used the strategy of attacking once and dashing away even with the slightly longer cooldown dash. This meant there would still have to be some more work to discourage this rather boring gameplay behavior by making it less effective, but it was already noticeable players mostly did it when they wanted to be especially careful at low health and were bolder and more aggressive when they had more health left.

Like players who played the first digital prototype version, the three last players found the control of the character fun and responsive and combat in general also felt satisfying for them. Most issues the players faced were clearly due to the prototype stage, such as lacking clear enough animations and behaviors for enemies or not having well enough done level design for the rotational grappling hook to be as useful as it could be. The players found maneuvering the enemies to opportunistic positions and pulling off great moments with the grappling hook very enjoyable, which was indeed meant to be the focus of the game.

For all players, despite clear flaws of the game due to the prototype stage, the gameplay was found fun and engaging. The players were eager to try again even if they died during testing due to the levels being somewhat unfair at times and without any prompting many mentioned they were already having fun. As such, the digital prototype had reached its goal of validating if the core gameplay of the game idea is good: the answer was a resounding yes.

8 CONCLUSIONS AND RELIABILITY

This thesis set out to find out what is needed to create good core gameplay, with focus on hack and slash games as that is what the game project worked with falls under as its genre. It was found out that the player should stay engaged by offering them various interesting choices in the moment-to-moment gameplay and these choices should include challenge to not make them obvious. Such attributes are easier to add to the design by formulating the gameplay into a clear game loop, which helps make sure everything is connected and the flow of the game keeps running smoothly. The gameplay of the game project worked on for

this paper was designed with such a loop, which was utilized in designing the gameplay of both physical and digital prototype. The practice proved useful and kept the development very focused and clear and could have contributed to very few problems coming up throughout the process.

Analyzing the core gameplay of two games of the hack and slash genre showed that all design decisions should be based around a well-defined game experience, crafted with the target audience in mind. Some of these design decisions can be very small such as how many attacks the player can combo together seamlessly. Yet, these seemingly small design choices can have massive impact on the feeling of the game and if some of them do not fit the game experience, something would likely feel off to the player. Adding anything to the game that does not clearly fit the intended game experience should also be avoided for the same reason.

For fast paced games of the hack and slash genre, making sure the game character feels good and responsive to control was also deemed important. This way the players will enjoy the mere act of controlling their character, leading to a high level of enjoyment. This clearly worked successfully for the digital prototype, as the players did enjoy moving around with the grappling hook mechanic and how the character acted without long delays and being able to interrupt previous actions. In a full game experience, this would be even more valuable to make downtime between combat encounters more interesting.

The secondary question explored was how to prototype the designed core gameplay as effectively as possible. It was found prototyping should always start from the fastest and most suitable to work with method, such as physical prototyping where it is very easy to test drastic rule changes. For things like the game feel, this might however also mean starting with the digital prototype makes sense. Prototypes should be done fast, have clear questions they try to answer and be tested with other players as soon as possible, even if the designer feels the game is not ready for that yet.

When playtesting with other players, most of the information will be gained through simply observing the players interact with the game, but various other methods can also be used in addition. It is important for the players to play authentically, meaning the designer should communicate with them as little as possible to get a picture of how the game would play for players playing it alone. While for an early prototype it likely makes no sense to create an in-depth tutorial to teach the players as that would be very time consuming, taking care to not overwhelm the playtesters with all the mechanics at once is a good practice and more accurate to how the final game should play in the end. This was clearly noticed in the game prototype, as players struggled to pick up both the basic attacking and grappling hook mechanics at once.

All this information proved very useful for the game prototype creation process. The prototyping went smoothly, with first creating the basic gameplay in a Tabletop Simulator prototype in place of an actual physical prototype and then moving on to digital prototyping. Rules could be changed and tested fast in the Tabletop Simulator, and through creating the intended gameplay as closely as possible it was possible to see various play patterns when testing the game already at this stage. It would have been easy to create large gameplay changes at this point, but there was no need to do that in this case as the designed gameplay worked immediately with smaller tweaks.

The Tabletop Simulator prototype allowed figuring out the design of the core gameplay with ease, and the gameplay translated well over to the digital prototype, despite high level of abstraction the physical prototype had through a turn structure compared to the real time action of the physical prototype. When the design of the game was already clear from the physical prototype, the digital prototype creation was much faster as all the time could be used on thinking how to create the mechanics and their smaller details, rather than what the mechanics should be in the first place.

With the digital prototype it was found out that even if making a lot of assets for a prototype is a huge waste of time, spending some time finding and setting up

suitable placeholder graphics and sounds from the internet was vital to properly test a game that relies on audio and visual cues. This is specifically true for the hack and slash genre games like the one created, where players require visual and audio feedback to be able to react properly to what is going on in the game. While the graphics don't need to look pretty at this stage, having clarity of what is happening is important.

While the result of applying the discussed theory to the game project of this paper was successful, it would be beneficial for the same to be done for game projects that attempt to create more novel gameplay not clearly covered by existing game genres. As the game idea of the project worked on in this paper clearly fit under existing game genres, there was a lot of help from observing how other games of the genre had been designed. Should that not be possible, the usefulness of the theory could be evaluated more accurately.

REFERENCES

Adams, E. & Dormans, J. 2012. Game mechanics: Advanced game design, Berkeley, CA: New Riders.

Baron, S. 2012. Cognitive flow: The Psychology of Great Game Design. WWW article. Available at: <https://www.gamedeveloper.com/design/cognitive-flow-the-psychology-of-great-game-design> [Accessed 1.12.2021]

Clement, J. 2021. Number of games released on Steam worldwide from 2004 to 2021. Statista. Available at: <https://www.statista.com/statistics/552623/number-games-released-steam/> [Accessed November 27, 2021]

Fullerton, T. 2014. Game Design Workshop: A playcentric approach to creating innovative games 3rd ed. Boca Raton, FL: CRC Press.

Gabler, K. 2009. YouTube video. Available at: <https://www.youtube.com/watch?v=aW6vgW8wc6c>. [Accessed 11.12.2021]

Heart Machine. 2016. Hyper Light Drifter. Video game. Culver city, California: Heart Machine.

Karppanen, A. 2016. Flow'n hyödyntäminen videopeleissä. Karelia-ammattikorkeakoulu (Pohjois-Karjalan ammattikorkeakoulu). Bachelor's thesis. PDF document. Available at: <https://urn.fi/URN:NBN:fi:amk-201605249576> [Accessed 28.11.2021]

Lovato, N. 2020. How to perfect your game's core loop. GameAnalytics. Blog post. Available at: <https://gameanalytics.com/blog/how-to-perfect-your-games-core-loop/> [Accessed 28.11.2021]

Pozzi, N. & Zimmerman, C. 2012. Don't follow these rules! A Primer for Playtesting. Essay. Available at:

<https://static1.squarespace.com/static/579b8aa26b8f5b8f49605c96/t/5962a494bebafbc89ca001b6/1499636884792/A+Primer+for+Playtesting.pdf> [Accessed 11.12.2021]

Supergiant Games. 2020. Hades. Video game. San Fransisco, California: Supergiant Games.

Swink, S. 2007. Game feel: The secret ingredient. Blog post. Available at: <https://www.gamedeveloper.com/design/game-feel-the-secret-ingredient> [Accessed 11.12.2021]

Video game industry statistics, trends and data in 2021. 2021. WePC. WWW-document. Available at: <https://www.wepc.com/news/video-game-statistics/> [Accessed 1.12.2021]

Zukowski, C. 2020. How to get feedback on your game. Cheaply! Blog post. Available at: <https://howtomarketagame.com/2020/03/02/how-to-get-feedback-on-your-game-cheaply/> [Accessed 11.12.2021]