Player Profile Models

A Research in the Phenomena Player Types

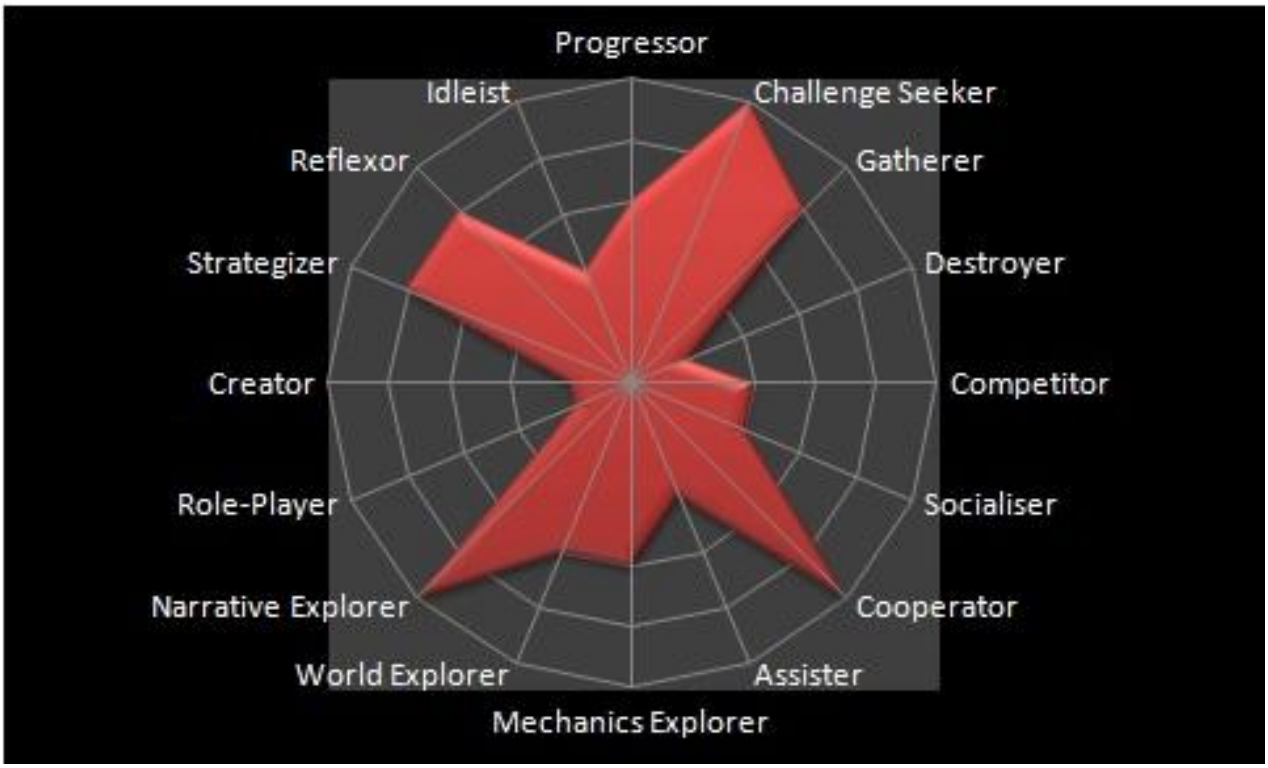

Table of Contents

Illustrations

Tables

Code Snippets

1 Motivation

I have been playing games for almost all my life. The different ways I've been affected to try new games have been many but I have bought more games that didn't fit me as a player than I've bought games that did.

Later on while being an active player in World of Warcraft I found that it could be hard to find like-minded people to play with. For example some people might give up faster than myself, or want to do other content parts, than the ones I wanted to do, which could frustrate.

In the later years I have studied as a games researcher. While creating test's I have found problems during testing because of the experience of the test participants. An example could be a test I did on competitive attitude where you could clearly see a difference in how players reacted depending on whether they normally liked playing in a competitive setting.

Finally I simply find the subject interesting and can see a lot of resemblance to the Myers-Briggs personality type test. (Briggs K. C., 1962) If it was possible to map what kind of a player people are, the same way companies use the Myers-Briggs to test which kind of person their new employees are, it would benefit both on the academic side and the corporate side of the gaming world.

In all cases I find that something common is missing at the moment. As a player it would benefit me if there was a better way of finding out whether a game is something for me. At the same time it would benefit me if it was possible to see whether the player wanting to join my group were the same type of player as the rest of my team to avoid conflicts of interest. As a researcher I think it would be beneficial to have a better way of delimiting test participants to people who actually matter within the given tests context. In general I think that all researchers and companies could benefit clearly from having a determined player profile standard which all players worldwide could be compared within.

2 Introduction

The concept of player types is an interesting one. Imagine being a player and have a model that would be able to tell you exactly whether or not you would like a game. Imagine being a game developer and be able to precisely know how large a group of a target audience your game would match. Imagine being a researcher able to make a demographic test depending on the precise type of player the test is designed towards. These are but some of the reasons the subject of player types is interesting and the reason this project has been created.

For many years researchers and game developers alike have worked with a concept called player types. One of the researchers creating this concept was Bartle back in 1996 and even to this day his model is still used in some regards. Bartle and other researchers all work from a standpoint that a player mainly has a specific type and works together or against other player types. (Bartle R. A., 1996)

Most people today would agree that this notion is wrong, just because a player likes socializing with his friends does not mean he does not like to achieve the goals of the game just as much. So even though Bartle's and other researchers theories are seen as somewhat wrong they are still used by researchers and gaming companies since they are the clearest model of a player base there is at this point in time.

The solution to this problem that this project will come up with is a new model which can replace the current concept of player types. The difference between these two models is that the new model will be able to take into account that a person may be regarded as a combination of a lot of effects rather than being seen as someone who has a main play style. While the focus of the project is to create a player profile model, the model will also be able to hold game profiles and thereby compare games and players with each other. Furthermore the Player Types was built towards the specific phenomena of MUD's (Multi User Dungeons) while this new model should work on any type of game.

The biggest problem about creating the player profile model is that people both play in certain ways and play for different reasons. This project will distinguish between the two and be focused on the concept of how people play games. The idea is that, to effectively map a player profile one would need a model showing why people play as well as one showing how people play, but since that has been a bigger focus for earlier researchers this report will focus on how people play.

Any person will then be able to identify themselves in the model, and be able to use it to see who they can play with, to get the most out of their gaming experience. For example two people with a high incentive to play challenging content might have more fun playing together than if one of those people gives up relatively fast. At the same time gaming companies can use it to develop games with clear focus on specific attributes within the model to try and target players with specific influences. Finally, researchers would be able to use this model on

test participants without having to change different concepts within the model in each new project as seen on the research scene today.

This project will focus on researching state of the art and use new input to create a model that is as accurate and flexible as possible. Then several small games will be implemented. These games look quite alike, but are quite distinct in the way they play out. In the end test participants will fill out the model directly and indirectly by filling out a survey. Then each test participant will play the developed games that affect different parameters within the created model to see if the game that matches the player's choices in the player profile model also is the game the player prefer.

3 Pre-analysis

Following the introduction a pre-analysis will be conducted. This is to delimit the problem into a manageable size. To do this the pre-analysis will mostly be a state of the art research chapter with some discussion about the different research. This approach is done due to the fact that there is quite a lot of research in the area and it would therefore be advantageous to use that. The pre-analysis will be split into two parts. The first part will focus on research related to player types, starting out with a more thorough look at Bartle's Player Types. The pre-analysis will look at different ways to look at the concept and how different researchers have tried to come up with a substantial solution to the problem of mapping players. Since this report focuses on how players play instead of why they play, that is an important thing to keep in mind when reading the different state of the art. Why we play is all about the reason behind the choice of playing a game. Is it simply to have fun, is it to meet new people, is it for the sake of escaping from everyday life or another reason altogether. How we play is about what a person does once they are inside the game, do they play strategizing battles where they use their brain to win, do they run around and try to get to the top of mountains, do they collect all the carrots within the game, or do they do yet another thing.

The second part of the pre-analysis will look into continuation desire. Continuation desire is the state of mind we are in while playing that keeps us playing. Player types are important because of this concept, and all researchers on player types will at some point have to talk about how the player type mixed with a specific game, influences the continuation desire. The reason continuation desire is so important is that it is this state of mind player types have the greatest effect on. A player might buy Minecraft even though s/he's not into building. He might come back at some time because of different effects like friends wanting him to try again, or a new patch being released. But if he is not into creating things and wants a narrative world his continuation desire will be limited.

The following sections will firstly look into a few relevant areas of research that could be applied to get a better understanding of what has and has not worked in earlier research. The relevant information from these studies will be extracted, discussed and finally formed into a Final Problem Statement, which this report will then attempt to answer.

3.1 Research Approach

Before delving into the pre-analysis and the analysis this subchapter will quickly describe the approach taken towards research. The project will use the general hypothesis project structure. This structure is based on the question coming first. Then research is gathered.

Because of the ease of finding different research theories within the subject of player types the analysis chapters will begin by doing extensive literature reviews where the most important and relevant will be covered. The literature will be gained from both scientific papers, books and online articles.

To be able to find more research the research gathering will also be gained by multi-linking references. This will be done by looking at references in used articles and find articles referring to the research currently used.

The information gathered from the literature review will be complimented by observations. These observations are mostly gained from an extensive gaming history. Finally focus groups will be used to have outside unbiased sources go into detail about the subject.

3.2 Player Types

Player types are a concept that has been researched since Bartle came up with the idea in 1996. A number of researchers have tried to categorize player types with varying efficiency. This section seeks to highlight three of the major researchers, and evaluate the methods used by them.

3.2.1 Richard A. Bartle

Back in 1996 Richard Bartle (Bartle R. A., 1996) sought out to categorize the players in the phenomena known as MUD's (Multi User Dungeons). MUD's were a kind of grandfather for the MMORPG (Massive Multiplayer Online Role Playing Game) genre which later got widely popular with the release of games like World of Warcraft, Everquest, Guild Wars and many other titles. Bartle's research ended up splitting players up into four different categories. Those four different categories are Achievers, Explorers, Socializers and Killers.

Achievers are focused on completing goals within the game. These goals could be both intrinsic like "hmm let me see if I can collect 100 gold in one hour" or extrinsic like "ok this woman wants me to kill 10 rabbits and she will give me a reward".

The explorers will explore the game, and here Bartle means both the games breadth like all the areas on the map, and the games depth like how does this ability work and can it be exploited somehow.

Socializers are mostly using the game as a sort of communication tool. They like talking with other players and mostly use the game mechanics to have some common sort of interest to start conversations with.

Killers as the final group within the player types get their kicks from giving as much distress as possible to other players. A killer will try to give others a bad time by

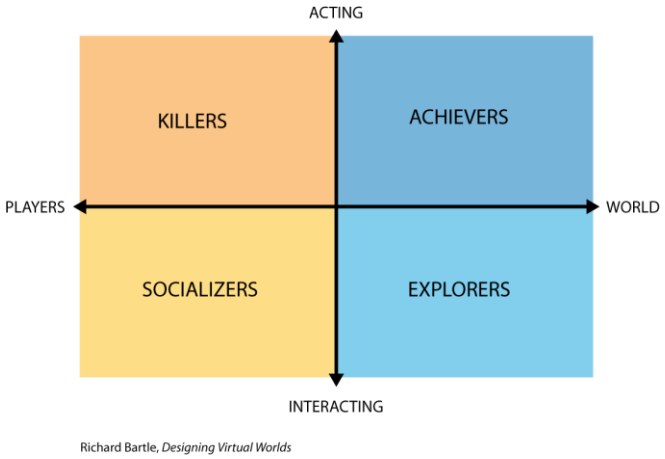

Illustration 1: Bartle's Player Type Model

executing them, destroying their stuff or in any other way make peoples life miserable within the game context.

These are the four player types Bartle categorizes players within. In Bartle's opinion a player will always prefer to be behaving in one way more than the three others which are indicated by Illustration 1. These four types are split by the two axes acting/interacting and players/world. However through the years a person may switch out which type of player they are according to Bartle. Furthermore Bartle explains how each type interacts with other types, and how a large amount of people within one type might increase or decrease the number of people within other types.

In Bartle's theories a couple of problems arise. Even though this research is created for MUD's and can be extended to MMORPG's using it for a single player game removes the idea about Killers and Socializers to great extend. This is more a problem from the perspective of people that use Bartle's research more than it is a problem with the research itself. The other problem however is that Bartle argues that at any given point in a person's life that person will be inclined to be mostly one type of player. This is seen as wrong by many researchers (Dixon D., 2011) (Kilkku V., 2013) (Yee N., 2002) since one person can be inclined to for example like to both socialize and achieve and will not like to do one thing without the other. Today's MMO scene is a great example with group content being widely popular and with a lot of people liking the challenge of overcoming hard enemies with their friends which they also socialize with during the time they play. Thereby it is worthwhile to note that while Bartle's model is elegantly designed and well modeled it excludes important aspects and also makes the types overlap.

Later in his book Designing Virtual World's (Bartle R. A., 2003), Bartle softened up to the idea of having multiple roles affecting a person. However he still believed each person to have a ranking between their groups where one would be higher than the others.

In Bartle's book he also adds another axis to his model, which opens up for a 3-dimensional model with 8 types. The new axis is called explicit/implicit and the model can be seen on Illustration 2. In this new model Bartle have created 8 new types which replace the old Bartle Types. Following is a quick one sentence summary of each type and further information can be found inside Bartle's book.

Figure 3.4. 3D Player Interest Graph.

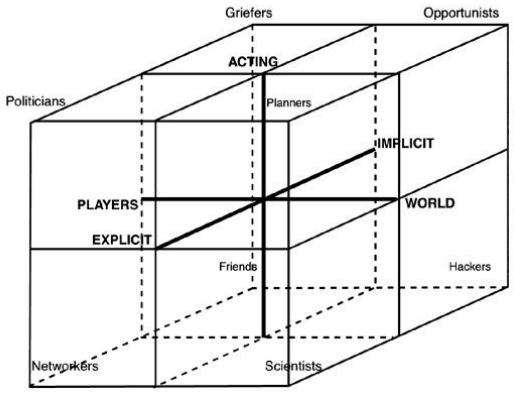

Illustration 2: Bartle's Advanced Player Type Model

Politicians: Players who act in an open fashion on other players, leading or interfering depending on how they are.

Networkers: Players who interact even with complete strangers on any and all subjects.

Friends: Players who mainly interact with the same well known group of people.

Griefers: People who will use force to get what they want.

Planners: Organized achievers who plans what to do and then does it.

Scientists: Explorers who use methodical experimentation techniques.

Hackers: Explorers that understand the world so well that they can explore based on intuition.

Opportunists: Achievers that go where they feel like.

These eight types take some of the problems of Bartle's model into consideration, but there are still some major problems with them. Even though Bartle claims the different types affect each other as lone types, they are even more overlapping than in the old model. A quick example is the Opportunist who in most regards would sound an awful lot like an explorer even though the explorer is split into the scientist and the hacker. Even though the model says politicians and friends are opposites of each other a person can easily be a leader when needed to while being a friend in the casual time within the game. So even with the upgraded model it is necessary to find alternatives to Bartle's Model.

3.2.2 Andrzej Marczewski

A more recent approach to gamer/user types is the approach taken by Marczewski (Marczewski A., 2013). Marczewski tries to take Bartle's model and use it as inspiration to create a model for the area of gamified systems.

Where Bartle has four player types Marczewski has created a model with six User Types which are each influenced by the thing the players are looking for within the game, like mastery or purpose as shown in Illustration 3.

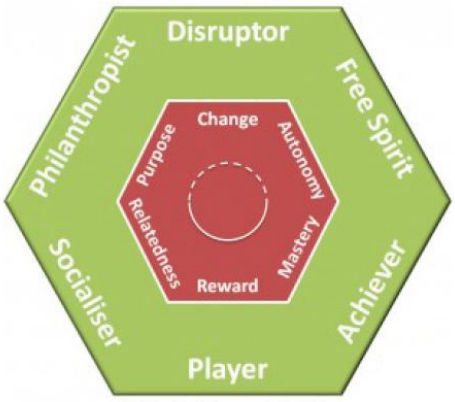

Illustration 3: Marczewski's Player Type Model

Socializers are very much alike Bartle's Socializers and therefore won't be discussed more in detail.

Free spirits can be seen as the same as Bartle's explorers however Marczewski amplifies that they both like to explore and create.

Achievers in this context is very much like Bartle's achievers however it is here important to emphasize on the aspect Marczewski talks about

called mastery. Since achievers here are in it to get knowledge and get better all the time which wasn't an aspect Bartle took into consideration.

Philanthropists are the first group that is distinct from Bartle's groups, these people try to help others learn and try to enrich the lives of people as much as possible. Think about people like the ones making information available on different wiki sides for games like guides and walkthroughs.

Players will do what is needed to get rewards within or outside of a system. They are in it for themselves and therefore won't help others unless it is needed to get better rewards.

Disruptors want to change the system either directly or through other users in a positive or negative way. Why follow a game rule if you don't like it?

Just as Bartle, Marczewski believes people to prefer to play one way over the other. However he loosens up a bit compared to his inspiration in terms of having the players and disruptors normally be affected by the other four types. For example a player might at the same time be an achiever if that helps him get rewarded.

3.2.3 Nick Yee

Perhaps the biggest opponent to Bartle and Marczewski is Nick Yee. He believes that every player like a bit of every type of game play. Nick Yee uses the model shown in Table 1.

Achievement	Social	Immersion
Advancement progress, power, accumulation, status	**Socializing** casual chat, helping others, making friends	**Discovery** exploration, lore, finding hidden things
Mechanics numbers, optimization, templating, analysis	**Relationship** personal, self-disclosure, find and give support	**Role-Playing** story line, character history, roles, fantasy
Competition challenging others, provocation, domination	**Teamwork** collaboration, groups, group achievements	**Customization** appearances, accessories, style, color schemes
		Escapism relax, escape from real life, avoiding real-life problems

Table 1: Yee's Player Types

As seen in the table Yee (Yee N., 2008) spreads out how we play into three groups with a total of ten subgroups. Bartle and Marczewski would say that these subgroups are not necessary since they are included in their fewer tables and that some of the points overlap (Bartle Bog). Nick Yee together with other researcher (Dixon D., 2011) (Kilkku V., 2013) however believes Bartle's and Marczewski's very black and white vision on player types to be wrong.

However the fact that Yee believes that we are all playing games in all these ways is a dangerous claim, since one player that does not play any games for any socializing aspects will destroy his model completely if that is an important claim for him. At the same time Yee focuses on multi player games. Where Bartle's first research was aimed at MUD's, Yee's is

focused at MMORPG's. Therefore some points might be missing if talking about single player games.

At the same time Yee seems to be a bit confused in some regards to one of his ten subgroups. While nine of the subgroups are easily seen as how we play MMOs the tenth and last subgroup escapism has more to do with why we play MMOs. Therefore it doesn't really fit the rest of the points all that well, which is important to remember. It is important to look at why people play MMO's and games in general but since this report focuses on how we play in general it is important that it is remembered to distinguish those two concepts from each other.

3.2.4 Jesse Shell and Christopher G. Klug

A last research worth mentioning is the one by Shell and Klug (Shell J., 2006). This paper is mentioned to discuss why it is important to distinguish between how people play games and why they play games. In it they list nine different player types on the emphasis of why we play, and their idea is that any person is at least two of these types and might switch emphasis depending on what game they are playing. This last idea is interesting and holds some truth to it, however if one were to use the model to anything a player would need to have a static model that could be used to compare with all games. The types are shown on Table 2.

Player Types	Description
The Competitor	Plays to be better than others
The Explorer	Plays to experience the boundaries of the game world
The Collector	Plays to acquire the most stuff throughout the game
The Achiever	Plays not only to be better now, but to be better on the long run
The Joker	Plays for the fun and social aspects
The Director	Plays to be in charge
The Storyteller	Plays to create and live inside a narrative world
The Performer	Plays for the show s/he can put on
The Craftsman	Plays to build, solve puzzles and engineer constructs

Table 2: Shell and Kurg's Player Types

Just as with some of the other research, the focus lies on why people play games. However when looking closely at the different parameters it is easy to see that some of them can easily be mentioned as a how we play. Shell and Klug also says that any combinations of types can be chosen and that we are either or on each type. This means the model is a bit hard to work with for game designers. While it is easy to design a game for the collector how do you specifically design a game for the joker? So the conclusion to this question is that if a model distinguishes between how people play games and why they play games it is also much easier to use it for game design.

3.2.5 Summary

Since the research area of player types is large this chapter could have discussed dozens of other research projects. However with the ones mentioned the idea about player types and how the concept is used today is shown.

So to sum up, the biggest discussion is whether people have a player type or not. This report takes the stance that this approach is too black and white. People are different, but some people might be closer aligned in their likes and dislikes than other people. Therefore a middle-way has to be found. Shell and Kurg was on the right track when saying that people can be a bit of everything. However to fix the problem they mentioned with people switching emphasis between player types depending on the game played each type will need a bigger scale than to be either or. This model will therefore consist of many player preferences and the amount of those preferences will turn into a player type. People can like some things more than others but are not limited to one major thing. Furthermore it has to be possible for a model to show that a person is not the least interested in one player preference. One of the major problems with the research in this chapter is that all of it except Shell's and Kurg's are used for multiplayer environments while this report focuses on games in general. This means that some aspects won't come up for all people. A person only playing single player games won't have much of a socializing aspect in his gaming (he might have outside from talking with friends about the games, but nothing besides that). A final aspect the researchers have in common which this report wishes to distance itself from is the fact that all researchers here don't distinguish between why and how we play.

3.3 Continuation Desire

As mentioned earlier this paper focuses on how we play games. However how we play is strictly linked to why we play (if we don't get to play the way we want we might not play at all). The following chapter will investigate state of the art within the field of continuation desire and use that to delimit how we play are linked to why we play.

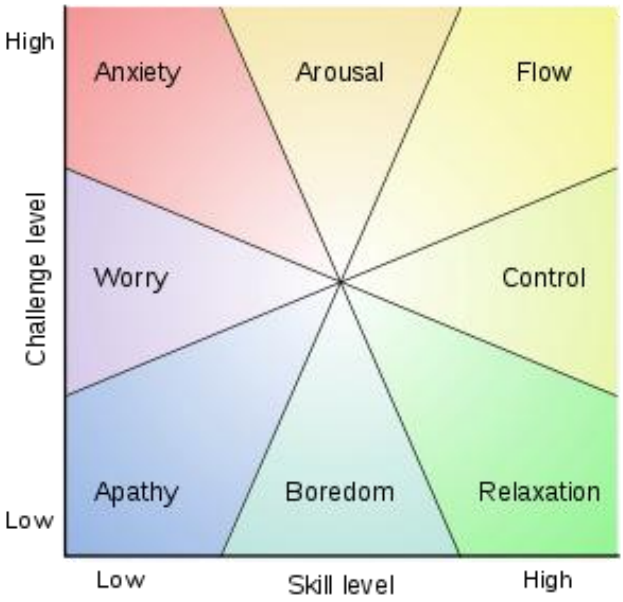

Illustration 4: Mental State in terms of challenge level and skill level

3.3.1 Flow

Continuation desire is grounded in the theory of Flow which was created by Csikszentmihalyi and Nakamura. (Csikszentmihalyi M., 2002)

Csikszentmihalyi have later continued his work with the concept and is today known as the father of flow. Therefore the rest of this chapter will mostly be referencing to him. Csikszentmihalyi believes flow to be the ultimate experience within the state of mind. Flow is together with experiencing out of the everyday life events the two things that can bring forth happiness in a humans mind. It is a state of the human mind where we focus greatly on something where we have great knowledge, completely forgetting everything else.

When in flow people say they forget things like hunger, tiredness and everyday problems. Csikszentmihalyi calls flow the way to happiness and refers to ecstasy as the emotion one experiences once inside flow.

As seen from Illustration 4 flow is a state of mind found when a person's skill level is high and the challenge presented to that person is high. As seen some of the other state of mind are great as well but the closer we get to flow the greater the feeling is.

This concept can be linked to the theme about how we as players play games. If a player is for example really great at playing games where he has to be very strategic and then is suddenly moved into a game where there isn't much strategy his skill level will plummet to a low level thereby giving him the feeling of either apathy, worry or anxiety when playing which would make him stop playing within a short amount of time. The same would happen if his skill level wasn't depleted all the way but the challenge were low and he would then experience boredom.

3.3.2 Continuation Desire

A researcher named Schønau Fog indicates that there are three periods within the lifetime of the relationship between a gamer and a game. (Fog H. S., 2011) The period where the gamer wants to try a game for the first time, the period known as continuation desire and the period where the gamer wants to come back for more or not. How we play is mostly linked to continuation desire and coming back to play again. The reason for this is that if a player loves playing in a cooperative setting with other people their continuation desire might not be all that high in a game that does not offer cooperation elements unless it has other very important aspects for the player. Choosing to buy and start playing a game is more defined by genre and regular marketing together with other influences like friends influence.

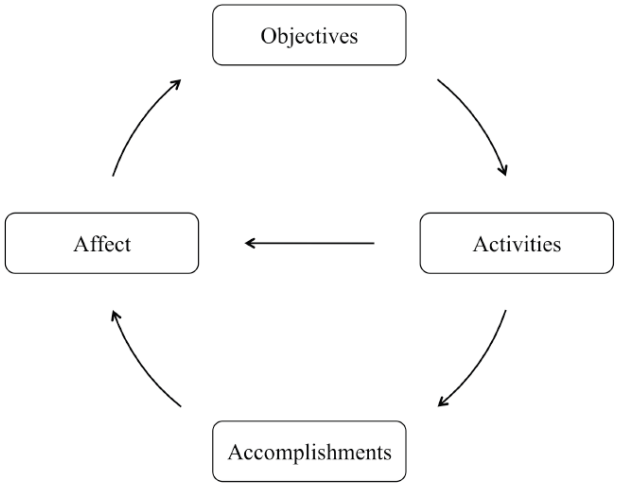

Illustration 5: The OA3 framework

Schønau Fog then focus on the continuation desire aspect where he splits parameters that keeps us to come back “why do you play” into four groups as seen on Illustration 5.

To summarize fast Objectives is about what a player wants to accomplish within a game, this can either be extrinsic (game created objectives) or intrinsic (player created objectives) objectives. Since the only way to affect objectives within game design is to add or remove extrinsic goals from the different activities these two parameters won’t be added to the model of how people play.

These Objectives are performed doing Activities. Activities is the part together with accomplishments that are closest related to how people play games and therefore also the group where this report can get inspiration for grouping within its model. Activities in Schønau-Fog’s paper refer to all these causes of continuation desire. This list is shown in Table 3 and will be used later in the paper together with the other player type models as a checklist for the created model.

Activities	**Description**
Solving	Solving Intellectual Challenges
Sensing	Experiencing Audiovisual Elements
Interfacing	Carrying out physical actions
Exploration	Exploring the game world and setting
Experimentation	Modifying the game
Creation	Build the game world
Destruction	Destroy game world objects
Experiencing The Story	Experiencing the storyline progression
Experiencing The Characters	Experiencing the characters progression, both player character and NPC’s
Socializing	Sharing experience with others

Table 3: Fog's causes of what makes people have continuation desire

When doing activities a player will get Accomplishments these are the rewards within the game or when reaching own objectives and can be grouped up into three subgroups. Achievements are when players try to achieve or get something. Progression is when players complete something to see what’s around the next corner. The last type of accomplishment is completion which is about completing what the game has to offer, finishing the story, or finishing intrinsic goals. Here Fog has a problem about completion being able to be the completion of achievements and thereby overlapping the achievement part quite a bit. While it is true that one can do something to complete an event without caring about the reward of that event, one can also argue if completing an achievement can’t psychologically be based on both accomplishments. Otherwise these three parameters are quite interesting and might be used in some regard in a model of how we play. A player that plays for the completion model will often love a challenge of a hard game while a progressor will often seek lesser challenges because they mostly want to progress further and the achiever just want to get things like better equipment and the like.

The last subgroup which is affected by activities and accomplishments are affects. Affects are how we are affected by a game, this can be in a positive way or a negative way, and

furthermore we can get absorbed by a game which isn't directly possible to add to either of the other subgroups since it can either be positive or negative.

With the information from this chapter the reader should have an idea about what continuation desire is and also some inspiration into which aspects can create this continuation desire.

3.4 Summary

This chapter was designed to get an overview of the strategies people have had when talking about player types and like minded theories. In 13.1 Table Summarizing Player Types the player types and game play types have been summarized. In this table Fog's activities and accomplishments are added as well since they will be used for creation of the model. Some of the most renowned researchers and their ideas have been summarized. From these theories a final problem statement has been created.

Most researchers think about a player type as a big union of concepts together, where they include both how and why and to some extent even who (females vs. males for example). Instead this report will therefore try another approach focusing on the question of how do players spend their time when playing games.

With the research in this chapter it is possible to make a background for creating different parameters for a new model which will be processed in the analysis. This model should try to avoid the problem that the most popular models are affected by which is that they either say that you are mainly one type of person or that everyone is a little bit of everything.

The second part of this chapter went more into detail about continuation desire and thereby what makes us continue playing. The reason for this is that how we play is tightly linked with these concepts.

4 Final Problem Statement

The pre-analysis determined the theories behind some of the leading researchers in the fields of player types and continuation desire. From these theories it is interesting to see whether it is possible to create a model that can show how a player plays computer games and use that knowledge as a player profile.

The goal is to see whether this newly created model can predict how much a player will like/dislike any game. Therefore the final problem statement is as follows.

Determining whether a player would like a game beforehand can be challenging. How can a model with relations to player types be created so that it is possible to fit gamers and games together more efficiently.

To be able to test this problem statement a test will have to be conducted. Therefore the following null-hypothesis is created on the base of the model being able to hold both players and games. Since the experiment conducted is a bit different than regularly the null-hypothesis will be explained afterwards.

Null-Hypothesis: *There is a difference between how a player will rank two created games, and how closely that player's profile model is correlated to said games' profile models.*

The experiment will have a player play three games created by the author. Players will then be asked to rate these three games from 1-10 (from bad to good). However since the games are not finished products, but rather small prototypes, people may rate them in a different manner (9.1 Method). Therefore the ranking between the games is used, so for example if a player rated one game 5 and another 8 the second game will be ranked higher than the first. The rankings of the three games are then compared with rankings calculated from the difference between the players' created player profile model and the pre-defined profile models of the three games. This process will be described in chapter 9 Test.

5 Analysis

To investigate the problem statement a number of topics have to be looked into. These subjects include which parameters the model should contain, which chart should be used to display the model, how to fill out the model, and last but not least how to test the model. A checklist follows and shows how the created model covers the parameters within the other researcher models and description of the reasoning behind parameters not covered is discussed. Finally from the findings of these different subjects it should be possible to create a list of requirements for describing an experiment and eventually requirements for a product needed for said experiment. To resolve these issues three different information sources have been used. First off past literature review has been used as inspiration. Secondly own ideas based on every day experience within gaming have been added to the mix of research used. Finally the qualitative testing method of focus groups has been held, with three groups, where especially the parameters of the model and how to add the values have been discussed.

5.1 Model Parameters

Finding the parameters that describe the X axis of the player experience chart is the first part of this chapter. To find the parameters to put inside the model research and personal experience ideas were added to make a list of parameters. After that three focus groups were questioned about these ideas with one mistake created on purpose to catch any groups which didn't have enough experience within gaming. This chapter will describe the process of how the parameters where found and in the end list and describe each parameter put into the final model.

5.1.1 Finding the Categories - Researchers and Own Additions

The first step in finding the parameters were to take each researcher used earlier adding a couple and finding out which of their parameters to use.

The chapter will start out with Richard A. Bartle (Bartle R. A., 2003), (Bartle R. A., 1996) and his model shown on Illustration 1: Bartle's Player Type Model. Bartle categorizes players into very large groups of four, or in his newer model eight groups, and in a more precise model many of these should be split into several categories. However the socializer of Bartle's model is fairly distinct in the way they play games and therefore this could be used as a separate category in a more precise model. Bartle's explorer type is general for two kinds of people. Bartle enters both people who explore the breath of the world and the depth into the same group. With this more concrete model these two can be split up into a world explorer and a mechanics explorer. Bartle's killer is in newer terms known as the griefer. Therefore the model will add that as a separate category. Bartle's killer could have included other types of player behavior but since other researchers include these in their model we will get back to that. The last group in Bartle's model is called the achiever. This group is the hardest to directly transfer since it is very much both a how we play and why we play group. However looking only at how the question is if there is different ways in which people try to achieve goals. Fog in his paper (Fog H. S., 2011) went into details about exactly that so that will be taken once this chapter comes to that part. Bartle's upgraded model does not add anything important that the other models does not cover and is therefore not covered here.

Marczewski's model shown on Illustration 3: Marczewski's Player Type Model brought a couple of new aspects into his model that can be used to expand the created model. Especially Marczewski's Philanthropist and Free-Spirit will add categories to the model. The free-spirit of Marczewski is an explorer like Bartle's but at the same time they are creators. So whether to build stuff in the landscape or to create the items they are using they like to create things. This has been used to add two parameters into the model, the Builder and the Crafter.

Marczewski's Philanthropist is the person who will help other people within the game without maybe getting anything in return. This person could be called an Assister as it will be when added to the model.

The next researcher on the list is Nick Yee. From Table 1: Yee's Player Types a couple of new play styles can be added. First off are the Competition people. These are the ones that Bartle miss a bit with his killer. While the Killer (or Griefer in this model) is mostly killing others just below their skill level to frustrate said players the competition element in Nick Yee's model is more of a fair fight. Therefore the competitor will be added.

The second important group on Nick Yee's model that hasn't been included is the Teamwork aspect. This can be seen as a cooperator in terms of a player type and is therefore what this parameter will be called.

The third aspect from Nick Yee's model important to discuss is the group Discovery. Within discovery is what we earlier named as World explorer, but at the same time Nick Yee talks about people interested in Lore. Thereby adding the Lore-Junkie to the model. Lore-Junkie should be understood as a person who needs a lot of lore to read to be satisfied with the game world, it is not the best term but was the one passed on to the focus group participants which could talk about other names for the group.

The fourth and final important group from Nick Yee's study is the Role-Playing aspect. This group is not determined in neither Bartle's nor Marczewski's studies and therefore have to be added as well as the Role-Player.

Two other research papers linked with Nick Yee's in a paper by Barbaros Bostan was investigated as well (Bostan B., 2009). In Bostan's research he lists these three models and compares them to a model he creates himself. However as seen from Illustration 6: Bostan's Motivational Study Comparison and the two research papers (Sweetser P., 2005) (Malone T. W., 1987) just as Nick Yee these researchers are interested in the why we play instead of the how. The points these researchers have that Nick Yee does not, will not add anything to the how people play games model. Bostan's model itself is based on player needs with 26 different aspects. Bostan has aspects that other researchers, doesn't but mostly because of the focus of needs. For example Bostan adds a Sexual need within his model which this project can't really use for distinguishing how we play.

The final research combed for potential categories is the paper by Fog (Fog H. S., 2011). To start off Fog has by his way of ordering things shown how Bartle's achievers can be very different. From Fog's discussion two distinct kinds of achievers were found. Fog talks about the completionist as the one that seeks a challenge that he or she will have to overcome. The progressor however just looks to progress fast through the content as to see the next part of the story. These are two very distinct approaches to gaming and therefore the model will have the Progressor and the Completionist added.

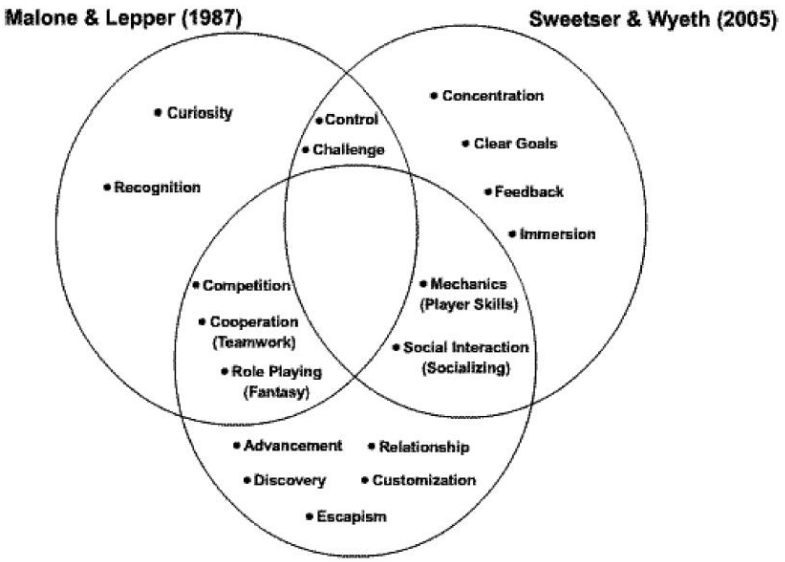

Illustration 6: Bostan's Motivational Study Comparison

Many of Fog's points are either strictly what causes people to continue playing and therefore might be hard to link to how people play. However one more can be linked to how we play that hasn't been added in the model yet. Fog's solving parameter can talk about players that

deliberately play content where they have to use their brains to strategize, solve puzzles and so on. These will be called the Strategist.

After adding all these parameters a couple more felt necessary based from personal experience. The Idleist and the Reflexor.

The Idleist is a type of person who plays without really paying attention to what's happening. A great example might be a person fishing in World of Warcraft while waiting to go play with friends. At the same time this person does other things in real life but even though he or she does not focus on the game they still sit and fish.

The Reflexor is a person that seeks games that challenge their reflexes, overview, memory and other cognitive abilities. While the strategizer and reflexor might seem as the opposite of each other and therefore should be able to be put into one category they are not. A game like Counter Strike facilitated both strategizing how to proceed and reflex based combat.

With these 16 parameters summarized in Table 4: Parameter Summarizing Table, it was time to talk with other people about the parameters and see their likes and dislikes. The reason to do this was to get other people's input so the model would not be created by one person alone.

Parameters			
Completionist	Progressor	Assister	Griefer
Competitor	World Explorer	Mechanics Explorer	Cooperator
Lore-Junkie	Crafter	Builder	Socializer
Idleist	Strategic	Reflex	Role-Player

Table 4: Parameter Summarizing Table

5.1.2 Finding the Categories - Focus Groups

The method used for foreign input is the collective conversation method known as focus groups. The reason to use this method is that it allows for in-depth discussions, and thereby the subject can be discussed in greater detail. Participants for the focus groups have been gathered using a combination of sequential sampling and

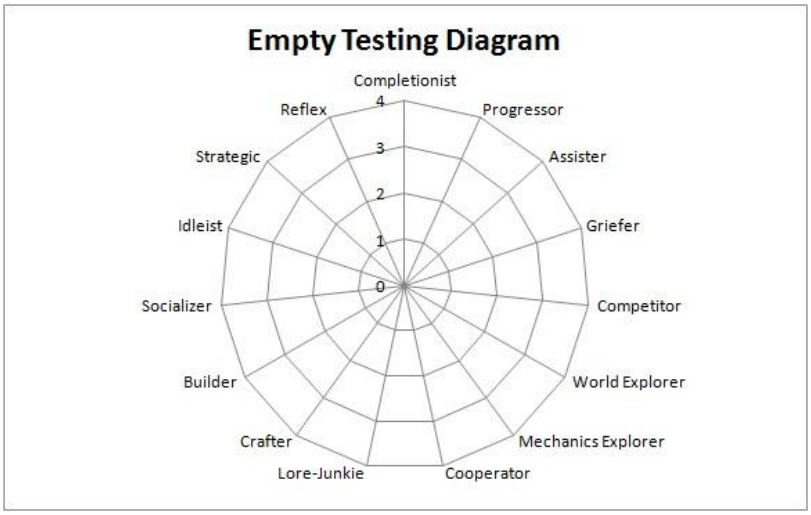

Illustration 7: Prototype Model without Role Playing parameter

convenience sampling with an added condition that people should play games regularly. Convenience sampling was used as a method of finding test participants because there were no other requirements than the fact that participants had to be frequent gamers. The sequential method was used to add discussion to each focus group. The participants were not chosen based on the previous results, but the final part of the discussion were changed depending on the results. Three focus groups with each 4-5 people were questioned for an average of 1½ hour. The first two questions were introductory questions to get the discussion started. First the groups were asked what their answer were to be if asked how they play games. Afterwards a common game all participants in a certain focus group had played was found and used for the next question. That question was regarding how the people played this game. After this process the groups were shown and explained Illustration 1: Bartle's Player Type Model, Illustration 3: Marczewski's Player Type Model, and Table 1: Yee's Player Types. After this discussion the groups were shown Illustration 7, showing 15 of the 16 categories found in the previous section. The reason to leave one parameter out were as earlier mentioned to make sure whether the groups caught the missing parameter and thereby showing relative knowledge within the field. The groups were asked to discuss the benefits and disadvantages of the model just as with the other three models. Furthermore they were asked to edit the model in any way they saw fit to make it more precise. Once the groups finished discussing the model they were shown the previous group's change decisions and asked to comment on those.

As shown in Illustration 7: Prototype Model without Role Playing parameter, the observant eye might see the Role-Player missing from the chart. This was due to being able to see whether the groups would have understood the term about how we play enough to catch the missing parameter. All three groups told that they thought the model needed a parameter called Role-Player or something similar.

To summarize the three groups were in general quite similar in which mistakes they thought the chart had. In total the groups' similarities was used to merge two categories, add one, and change the names of four categories.

The category merged were the two categories Builder and Crafter being merged into one. The groups thought the two things were too much alike and that they should be merged into a single category called Creator. In general the arguments made with people using both in a game like Minecraft and the argument that the two categories overlapped made sense so the groups were merged.

At the same time the groups did not like the name completionist for the description of someone seeking challenges. Therefore the completionist was changed to Challenge Seeker which is a more describing term. At the same time a new category was added to include what people thought about when told of the completionist term. The participants all wanted someone that collects all the possible achievements, Pokémon's, quest completions and so on. The groups all struggled with the naming of this group, terms like collector and completionist were discussed but in the end the term Gatherer was decided by two groups. This term describes rather well what the category symbolizes and therefore it is the naming used for the model.

The last part was the renaming of two of the other groups. People did not like the term griefer since it is a very internal term within the MMO genre. Furthermore Griefers only hurt by killing other players and therefore the participants wanted to rename it to Destroyer since hurting others can be in other ways than killing them (example destroying their creations in Minecraft.) This was a mistake caught by the focus groups that were obvious when found and was therefore changed accordingly.

Lastly all the groups disliked the term Lore-Junkie and all three came up with the term Narrative Explorer to replace it. As mentioned earlier Lore-Junkie was a kind of placeholder name and was therefore changed. The notes from the focus groups can be found in appendix chapter 13.2 Focus Group Notes.

Thereby the chart and its categories are finished and the next sub-chapter will show the full model and describe each category.

5.1.3 Description of parameters

This chapter will start out by showing Illustration 8 and continue with explaining a bit about each category. Several other researchers have grouped their categories into different groups. One of the focus groups considered this option as well. However even though the parameters of the model could be grouped, the idea behind it is to be more extensive and precise and therefore the grouping won't add anything but an extra visual layer.

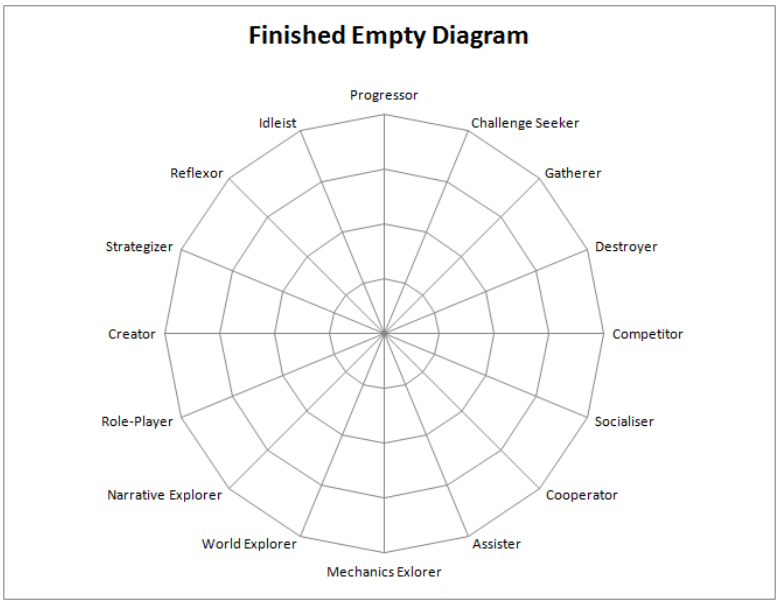

Illustration 8: Finished Player Profile Model

As the focus groups went on it was discussed what the minimum and maximum value should be. In general the test participants found the best value

to be 0-4 as to make room for variety but at the same time not make too many values and thereby making the difference between each too shallow.

- **Progressor**
  The progressor seeks to continue through the game in a fashionable speed. If they get stuck they quickly lose interest and might be inclined to stop playing. With a high value in progressor you might be inclined to seek out games where you can progress through the game without too much of a challenge. Example of games that allow for a high progressor value would be games where a player can add easy to the game difficulty and complete most things in first try.
- **Challenge Seeker**
  The challenge seeker seeks the thrill of the challenge. He knows nothing better than finding an opponent that kills him 100 times because if he then defeats it after 101 tries he will get a rush of accomplishment. If you have a high challenge seeker score put games on the highest difficulty and look for games within your genre that renown themselves on being the hardest.

- **Gatherer**
  The gatherer collects all the things. Whether it is achievement points, Pokémon or vegetables. The gatherer wants to get them all. Games allowing for any sort of collection can affect the gatherer, however if the game specifically is about collecting it will affect stronger.

- **Destroyer**
  The destroyer puts hurt upon others. This can be by killing them; destroying things they have created, stealing from them or in other ways making them angry. With a high destroyer score the player first and foremost wants to find games where other players are present. Secondly the game has to have as much possibilities of hurting others as possible. Preferably it should be possible to find targets less skilled than one self.

- **Competitor**
  The competitor wants to play against others in as fair a setting as possible. The competitor seeks games where this is allowed. The competitor as the destroyer looks for games with other players and where one can affect others. However while the destroyer wants to hurt others by doing unfair things towards them, the competitor looks for a fair fight.

- **Socializer**
  The socializer uses games as a social platform. Whether it is to talk with people inside the game or playing the game just to be able to follow the discussion with ones friends the player does it for the social aspect more than any game aspects. High value socializers should look for multiplayer games where communication tools are great. Other than that socializers can look for games that people in their everyday life are playing.

- **Cooperator**
  The cooperator wants to work together with others to complete a common goal. Whether it is to defeat some game created challenge or some challenge these players create themselves is not necessarily important as long as it's a challenge they have to work together to complete. The cooperator should look for multiplayer games where there are challenges that cannot be completed alone.

- **Assister**
  The assister uses a lot of time on helping others. Whether it is by telling them where to go. Or whether it is by actively helping them kill some monster the assister just uses time on this aspect. Assisters should look for multiplayer games where the communication are great and where people are looking for advice from time to time (so there is actually a need for the assister).

- **Mechanics Explorer**
  The mechanics explorer looks for all the small extra features in the mechanics of the game. This can be as simple a thing as to find out how much a headshot hurts, compared to a shot in the leg. Or something as extensive as finding the optimal build in an MMO. The mechanics explorer should look for games with as extensive mechanics systems as possible. For example a game like flappy bird can be explored rather quickly.

- **World Explorer**
  The world explorer sees a tall mountain far away and straight away wants to go there. Even though the game might have sent the person to some other corner of the world, the world explorer wants to see all the caves, mountains, fields and so on. A world explorer should look for games with as open world as possible. A very linear game, or a game where the open world is not in any way interesting, will limit the possibilities of exploring.

- **Narrative Explorer**
  The narrative explorer is the guy or girl who knows every background story of every monster in the game. He will read every quest text, talk with all the NPC's, and read every single book of lore in the game. Some might even go outside the game to look up background stories of characters and monsters within the game. The narrative explorer should look for games with as big a narrative width as possible. For example the story in a super Mario game can be decent but the width and story amount of each character is rather limited.

- **Role-Player**
  The role-player is the character instead of playing the character. The role-player might take an action that for the game plays sake is obviously worse but for the sake of the character seems more realistic. For example if a role-player has chosen that their character is scared of spiders instead of killing the spiders they might stun themselves

in terror. Role-players should look for games where the story of the main character is not defined already so they themselves can define it.

- **Creator**
  The creator adds things all around them. Whether it is buildings or other things affecting the landscape around them, or costumes and items that can be equipped. The creator should look for games where there are great crafting systems and where it is possible to change the environment as a player.

- **Strategizer**
  The strategizer plays with their mind. They try to figure out the optimal way on completing a challenge and might even go think about how it can be completed without even playing. Obviously the strategizer should find games where they have to come up with plans to complete the challenges rather than just trial and error.

- **Reflexor**
  The reflexor challenges their cognitive abilities as much as possible. This can be their reflexes, their memory, or any other ability. They improvise a lot from how the game acts on them. In short if a strategizer acts, it's possible to say that a reflexor reacts. A reflector should look for games where their reflective cognitive abilities are put to the test. For example a turn based game might not be the best game for a reflexor while a shooter might be a great choice.

- **Idleist**
  The idleist has the game running but for the most part the computer works harder than the idleist. The idleist does other things while playing and their focus might not be all on the game. Idleists should look for games that does not require full attention like a click-reward game (Farmville) found on facebook. An example of an idleist behavior would be the auction house players in World of Warcraft. These people are online for most of their waking hours. While being online what they do is to look for wares on the auction house they can buy cheap and sell more expensive to earn gold. They might be online and do nothing else in game for many days but since the cognitive abilities used for this is rather low they are able to do all sorts of things outside of the game.

These sixteen parameters are held up against the parameters from appendix chapter 13.1 Table Summarizing Player Types in Table 5. A couple of parameters from the list are not covered by the model; the reasoning behind this is explained below. Furthermore some of the comparisons have to be explained as well.

Rueløkke	Bartle 1	Bartle 2	Shell	Yee	Marczewski	Fog
Progressor	Achiever			Advancement	Achiever	Progression
Challenge Seeker	Achiever		Achiever		Achiever	Completion*
Gatherer	Achiever		Collector	Advancement	Achiever	Achieving, Completion*
Destroyer	Killer	Politician, Griefer				Destruction
Competitor			Competitor, Achiever	Competition		
Socializer	Socializer	Networker, Friend	Joker*, Director*	Socializing, Escapism*		Socializing
Cooperator			Director*	Teamwork		
Assister		Politician		Relationship, Socializing	Philanthropist	
Mechanics Explorer	Explorer	Hacker	Explorer	Mechanics	Free Spirit, Disruptor*	Experimentation
World Explorer	Explorer	Scientist, Opportunist	Explorer	Discovery	Free Spirit	Exploration
Narrative Explorer			Explorer	Discovery		Exp. The Story, Exp. The Chars
Role-Player			Storyteller, Performer	Role-Playing, Escapism*		Exp. The Chars
Creator			Storyteller, Craftsman	Customization	Free Spirit	Creation
Strategizer		Planner				Solving*, Destruction*
Reflexor		Opportunist				
Idleist						Sensing*, Destruction*
NOT COVERED					Player*	Interfacing*

Table 5: Parameter Checklist

Marczewski's Disruptor and Player: The disruptor and player are very broad concepts that as Marczewski mentions can be combined with any other parameter. To play within the game world will therefore not fit into a how to play model since every player will do that at some point. Disruptor is more interesting but again this can just be seen as some sort of mechanics exploration.

Yee's Escapist: Escapism is a full out why people play games and the closest related how we play content would be the role-player and the socializer since these two parameters are some of the strongest at getting people away from everyday life. However most parameters could meet this criterion.

Shell's Joker: This parameter can be half fulfilled by the socializer however the part about just having fun, is not something that can be game designed and is very much a why people play games question. Therefore it is not covered in the model.

Shell's Director: The director is closely related to the socializer and the argument would be that you can be a socializer without the director, but being a director without the socializing would be difficult. Therefore this parameter is not added.

Fog's Sensing: This has a lot to do with why people play games. However if a game has its soundtrack in a separate part and a player enters that just to hear the soundtrack it can be put under the idleist category.

Fog's Interfacing: Since all parameters will be ranked from low to high, it is hard to use the interfacing aspect of Fog's model. How one does chose what is best between a controller and a keyboard and a mouse. The only real way to distinguish controls within a scale is the interactivity and thereby the exercise. However this makes the parameter a question about playing because of the need or wish for exercise, which is a why category.

Fog's Destruction: Fog's destruction as mentioned earlier is about the destruction of game object. Whether it is another player, one's own creations, or a part of the game world. The destroyer parameter can be aligned with the first of these three cases. The other kind of destruction can both be seen as a strategizing destruction or an idleist, fun, form of destruction. The fact that any forms of destruction outside those scopes are so limited, together with the fact that not a single person within the focus groups even mentioned this aspect, that it has been decided to keep this parameter out of the model.

Fog's Achieving and Completion: Fog's achievement and completion parameters are very closely aligned and are mostly a difference on why someone completes a task. Therefore these are both seen as a sort of gathering with the completion having a sense of challenge seeking as well.

With these descriptions and the table in appendix **Error! Reference source not found.** the model should show that it contains the important parameters from the other researches. The list only lists the researches mentioned in the pre-analysis since the other researches did not add any parameters. Therefore the next subchapter will discuss which chart the model should be visualized by.

5.2 Model Chart

One of the questions about the created model is which chart should be used to illustrate players' tendencies within gaming. The model will have 16 different parameters on the X axis, while at the same time it should be possible to measure each point within the chart. Following are three types of charts that each could be used to visualize the data of a given player. The advantages and disadvantages of these charts within the bound of this project will be discussed and one of them will then be set up to be the preferred chart.

5.2.1 Bar Chart

The standard Bar Chart which can be seen on an example on Illustration 9 is widely used. Its advantages are many but here the main focus is three great aspects. The model can show each data category in a frequency distribution. It can estimate key values easily (which is important for finding the important or unimportant aspects in a players gaming behavior.) And last but not least it is great at comparing data categories for two different objects (i.e. a game and a player or two players).

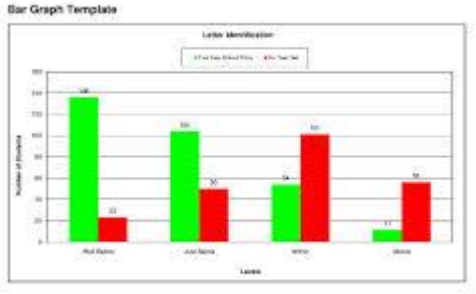

Illustration 9: Bar Chart Example

The big disadvantage with the bar diagram however is that with enough different categories the list of bars becomes exclusively long and overwhelming for the reader. This makes this model a bad choice since the model has to be easily readable by a player so he or she can look at each point. Therefore it would be smart to find a more compact chart.

5.2.2 Circle Chart

The second chart that could be a possibility is the Circle chart as exemplified on Illustration 10. Compared to the bar chart the circle chart is much more compact and no matter how many categories is added the model won't be any larger in the space it needs. As the bar model it is possible to estimate key values rather easily with the circle chart as well.

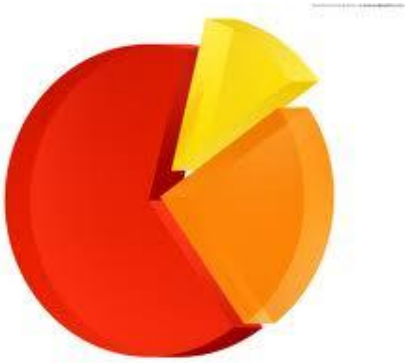

Illustration 10: Circle Chart Example

However the circle graph work in percentages instead of occurrences or real number which creates a big problem. It is not very precise and at the same time two circle charts cannot be compared to each other. Thereby it would be impossible to compare a player with a game as an example. Therefore a third option has to be found.

5.2.3 Radar Chart

The third option has to take the compact nature of the circle diagram and mix it with the comparability, frequency distribution and key value estimation aspects that the bar chart had as advantages.

One model that fits this description is the radar chart, or spider web chart shown on Illustration 11. Furthermore the radar chart brings another advantage to the table. While other tables are good at comparing magnitudes of different aspects within a model the radar chart can show

the symmetry of values as a whole. This is great for illustrating to a player that the symmetry of their model is close to the symmetry of a certain game. Therefore the radar chart is chosen as the chart used to visualize the data gathering.

With the model and its categories in place the next chapter will look at the different ways it could be possible for a person to get their personalized values.

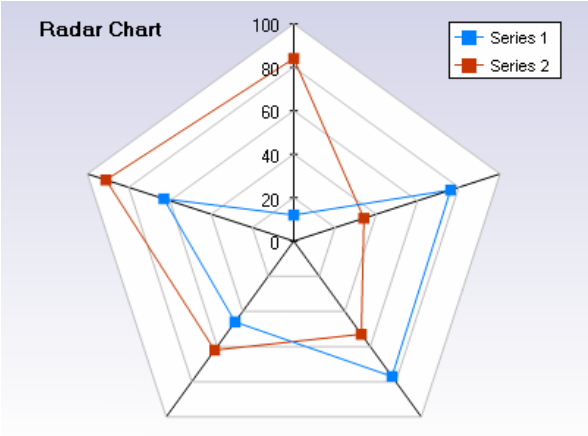

Illustration 11: Radar Chart Example

5.3 Adding values

Now that the model has been created it is time to find out how to fill out the values for a person. During the research period five different approaches have been invented. This invention has been based on inspiration from other researchers mixed with ideas from the focus group. Between the approaches there are both extrinsic and intrinsic ways of filling out the player profiles. This chapter will discuss each and come up with the best approach.

The first approach possible would be the direct intrinsic approach. The approach is to explain each category to a person and let them say how much they see themselves playing that way from zero to four. The benefit of this approach is that it is the fastest to implement of all five and the fastest to fill out for each individual person. The big disadvantage of this approach is if people are not able to give correct values to themselves. A person might believe challenge seeker to be a positive thing and therefore give it a higher value than it actually should have and thereby corrupting his model.

The second approach would be an extrinsic questionnaire. The questionnaire would be about subjects not quite related to the subject and where each answer of each question is as preferable as the other options. The advantage of this approach would be that if the questionnaire is designed well the player will get a very accurate prediction of his or her player type. By well designed questionnaire one can get inspiration from the Myers Briggs model (Briggs K. C., 1962) which is used by both small and large corporations when employing new people and is regarded as the common model for showing personality types. The model works just as Bartle's two models, however the Myers Briggs has four axes where Bartle's has two and three. The big problem with this approach is the time and resources it would take to design and create the questionnaire. The Briggs Myers model has four axes and nearly 100 questions. To be fulfilling in that regard this questionnaire would need almost 400 questions. This would take a long time to develop and would be such an extensive questionnaire that it would not be possible to use for common people since most people would refuse to answer it.

The third approach would be the extrinsic “What do you like to play” approach. This approach would ask a person to list off their three favorite games at the moment. Each game would then have a model of how they can influence players. These models would have to be gathered from the appropriate companies or each game would have to be played objectively and for a long time by the author so it would be possible to fill out the game models. A calculation would then create a model for the player from the game models. The advantage of this approach would be that the player only has influence on his or her values from whatever games they choose. At the same time the testing of each test participant would be the fastest of all the approaches. The disadvantages would be the implementation of this system however. To be able to make this approach the numbers of a large amount of games would have to be added to a system that would then always be able to come up with the numbers for a game a player had chosen. Since one person cannot fill out these numbers it would only be possible by getting in contact with some of the major game distributors, such as Steam, PSN, XBox-Live or more locally Viaplay. Since this has not been possible this is not an approach that has been possible in this project but could be interesting for future perspectives.

The fourth approach would be the “How do you like to play your favorite games” approach. This approach would like the third approach ask the players their three favorite games. However instead of having some common value of how the game can be played this approach would ask the player how they play each game, having them fill out a model for each game. After these models are filled out the player’s type would then be filled out. This approach might be more accurate than the previous since they remove the ways a person does not play a game. However as in the direct approach the problem is that the player fill out numbers themselves and thereby might corrupt the data.

The fifth and final approach is the metrics approach. By creating a game that could allow for every category equally and measuring how the player played this game it would be possible to calculate a model. This might be the most precise approach of all but at the same time the hardest to conceptualize. Creating a game that can influence all sixteen categories equally is an enormous task, and would require a large amount of time and resources. Thereby this approach is unrealistic for the scope of this project.

So from the five approaches two can be fulfilled. The direct approach and the “How do you like to play your favorite games” approach. To find out which approach is the most accurate a test was conducted. This test can be seen in chapter 9 Test.

5.4 Design of the experiment

With every aspect of the model in place the only thing left to find out is how to test whether the model works or not. For this two approaches have been explored. This chapter will explain each approach and the benefits and disadvantages of each will be discussed, and in the end a test requirement list will be created.

The first approach would be to compare each player's model with a list of games and have the player try one game that the model would recommend and one that the model would say is not something for the player. The benefit of this approach is that no technology has to be implemented and it is therefore a rather fast approach from here to the test phase. The disadvantages of this approach are that for one there has to be a strong library of games to be able to find a game that someone really dislikes and one that the same person likes for each person. At the same time to be able to fill out the model the writer of this report would have to play each game for an extensive time to get an accurate fill on the model. The other way would be to get developers to fill out the model for their game but that would be hard for so many games as well.

The other approach would be to develop three games within one game genre. Each setting would affect the player with a different set of categories from the model. That way once the player has filled out their model they can rate the three games. In that way it is possible to see if the player like the game with the closest model more than the other games. This approach has the benefit on having more precise game models since they are developed with the model in thought. The disadvantage of this approach is the time consumption of developing the game.

From these two approaches the latter is the best choice and therefore three smaller games will have to be developed. Furthermore the experiment will include the preliminary test and therefore the test participants will have to fill out the player profile model in two different ways.

From this the following requirements for the experiment are made.

- Players will have to fill out a player model.
- Model should be compared with three distinctly different game models
- Players will have to test all three games for a time period of at least five minutes each or until a certain goal have been met
- Results should indicate whether the model with at least 80% precision can predict how players will rank the games in comparison to each other.
- Test should be quantitative with at least thirty participants to be statistically significant
- It should be possible to remotely send out the test as to make the testing process easier.
- Test should be within subject tests as the test should be designed to find distinctions in subject rather than in groups.

With the test requirements in place the product requirements has to be created so the product can be designed and implemented.

5.5 Product Requirements

With the test requirements it is possible to create a list of must haves for the product. This subchapter will describe the different product requirements which will be used when designing the system. As mentioned in the test requirements the game should have three different modes. These modes should be significantly different from each other so calculations can be made. The games should all have the same controls to minimize the controls effect on the experience. The graphic style and sound features also has to be the same so as not to have any influence on the ratings of the user at all. Furthermore the game's controls should be so easy that they can be understood without the need of a test conductor.

So the hard requirements are:

- One general game genre
- Three distinct games from this genre
- Same easy to learn controls
- Same graphic style
- Same sound features
- Questionnaire

Furthermore some requirements are not necessarily needed but would be a great addition. The first soft requirement is that the system should be as flexible as possible as to allow for future upgrades. Furthermore it is important to have a strong structure on the project from the start as to have a good management over the many scripts and prefabs. Five of the parameters inside the model is multiplayer related, so to be able to test any of those parameters some sort of multiplayer setup would have to be created. However testing a multiplayer setup with several people would be unrealistic for the scope of this project and therefore any multiplayer features implemented should be asynchronous.

So the soft requirements are

- Flexible system
- Strongly Organized System
- Asynchronous multiplayer features

With these product requirements it is possible to design and implement the game used for testing the hypothesis.

6 Method

With the product requirements defined, the product is ready to be designed and implemented. However before the chapters describing these processes this chapter will describe the scientific methods used in the process of designing the system, implementing and testing. The product will be designed in an iterative design fashion as to catch bugs and flaws within the system during the process. The test will be designed as a within subject test. This is due to the fact that the test tries to compare subjects with each other. Furthermore the test will be designed as a quantitative test. The reason for this is that the test need a lot of test data to be valid for rejecting the null-hypothesis. The test participants are gathered with one general requirement in mind. All test participants have to play games regularly, but other than that the need of any ethnographies are not there so a convenience sampling have been chosen as the preferred method. The reason no ethnographies is needed is that the created model should be able to work for all game players and thereby the testing group can’t be broad enough. The project will include a preliminary test that should improve the likelihood of external validity within the final test. This test will research whether intrinsic or extrinsic data gathering is needed for the final test. To explain, the intrinsic data gathering comes is when a test participant gets to intrinsically fill out their player profile model themselves. The extrinsic fulfillment is getting the test participant to perform an action that then indirectly fills out the participant’s player profile model.

7 Design

With the research completed and the product requirements lined out the next step is to design the product based on the experiment. In this report this process is described in a very structured way, while the real process was much more iterative. This chapter will be segmented into two different parts. The first part will be the general game design aspect. This part will focus on which game genre to use for the experiment, the strategy for assigning the parameters into different games and the design of the three games with their respective parameters. This subchapter should make sure that the hard product requirements are all met. The second part of this chapter will be focused on designing the system, which includes which classes and prefabs to include and how they communicate with each other.

7.1 General Design Choices

As mentioned, this chapter will describe the general design choices for the system and with that the hard product requirements should be met. It will mainly focus on the important design choices made. These are the game genre, parameter assignment strategy, design of the three games and audiovisual strategy. This will lead up to the subchapter about the design of the actual system.

7.1.1 Game Genre

This chapter is created to handle the requirement about one game genre. Choosing a game genre is an important design aspect. Not all game genres' can implement all parameter within the model. As an example if turn based strategy games were chosen it would be hard to implement the reflexor parameter from the player profile model because of the aspect with endless time to take an action. As described in an earlier chapter the reflexor is the person who uses reflective cognitive abilities when playing games. Therefore the important part of this aspect is to choose a genre that is relatively well known and that can fulfill all the parameters without inventing new game mechanics.

Since some of the parameters within the player profile model like the cooperator or the destroyer, requires a multiplayer setting, one genre that came to mind was the MMO genre. A MMO game like World of Warcraft can fill out every single parameter within the player profile model as shown on Table 6.

It is self-explanatory that a single person cannot create an MMO the size of World of Warcraft in a four month time limit. Therefore the MMO genre has to be delimited into something more manageable.

The MMO genre is a more advanced multiplayer version of the Hack and Slash Role-playing Genre. It would be possible to develop on a hack and slash game and make it into an MMO thereby creating functionality for fulfilling any parameter within the model. Creating a Hack and

Slash game would therefore allow creating systems able to fulfill any of the parameters within the player profile model. Furthermore the hack and slash genre is a relatively simple genre to create and it is possible to find tutorials and different free assets that can help create the games.

Thereby the chosen genre for this project is the Hack'n Slash genre. All three games will share controls and style to lower any unwanted variations in test data. The general common aspects of each game for this genre were developed by following a tutorial created for this genre. (Laliberte P., 2010) With this chapter the product requirement of one game genre should be fulfilled.

Parameter	**Way to affect**
Progressor	Quests, Levels, Dungeon Layout
Challenge Seeker	Raiding, Hardcore Difficulty
Gatherer	Item Sets, Pet System
Destroyer	Open World PvP
Competitor	Battlegrounds, Arenas
Socializer	Hub Areas, Chat System
Cooperator	Instanced Group Content
Assister	Strategy Explanation, Ingame Wiki
Mechanics Explorer	Talent System, Rotation System
World Explorer	Open World, Achievements, Hidden Map
Narrative Explorer	Quest Text, Lore Books, NPC Emotes
Role-Player	Role-Playing Servers
Creator	Crafting, Housing System (In 2014 expansion)
Strategizer	NPC Mechanics, Class Synergy for PVP
Reflexor	NPC, Player and Environment Mechanics
Idleist	Crafting System, Chat System

Table 6: World of Warcrafts possible effects on parameters within created model

7.1.2 Audiovisual Strategy

As mentioned in the product requirements the artistic graphical style and the sound has to be the same for the three games to limit unwanted variations in test results.

The visuals mostly had to do with the graphics style and the models used. At the moment realistic grimdark style of graphics have been very popular with games like Guild Wars 2 and Skyrim being the inspiration. Furthermore instead of creating all the models from scratch it was decided that with the focus of the project on other aspects of development it would be possible to find free models within this style. These models were found on (Turbosquid, 2014) and (Unity Technologies, 2014) and can be seen on the DVD.

To limit the difference in test results from unwanted sources it was decided to remove any sound from the game. It would have been possible to create small sound effects or some background music but since the games are three different types of action RPG one tune or one piece of music might fit one game better than another. With different lyrics test participants might like one lyric better than another. Finally the sound effects might get more annoying in a game where they are played a lot (one with a lot of combat) than one where they are only played once in a while. Therefore it was decided to keep any audio out of the game.

7.1.3 Parameter assigning strategy

This chapter will be about the requirement to have three games from the product requirements. The finished player profile model has 16 different parameters. Of these parameters some are easier to implement than others, for example the cooperation requires a deal of network implementation, at least if the cooperation is live. Furthermore some parameters would make the testing setup bigger and more complicated, for example a great amount of test participants have to be put inside a setting for quite a while to test for a socializing aspect.

For this reason it is not possible to use all parameters inside the model. This is reasonable enough since not many games within the game market today actually have an effect on all 16 parameters.

The second objective to look at is how to assign the parameters within the three games. Since there has to be a clear indication on whether or not a player likes a game depending on the parameters used to affect the game play the parameters used should be different for each game. By delimiting from using the same parameters at all it is possible to get as clear an indication on difference as possible. Therefore the games will each get two differentiating parameters they will affect. Each game might outside of the design affect more than the two parameters but each game should be significantly different.

7.1.4 Design of games

With the strategy of game design completed this subchapter will talk about the design of the three games and which parameters each game has been developed to affect.

As mentioned earlier the controls will be the same in each game to lower unwanted test result variations. The controls have been developed in order to try and complete the ease of learning requirement within the product requirements. Starting out with movement, any player who have played a first person shooter, an MMO or a 3rd person adventure game will have tried the WASD controls. Therefore the movement system is a standard WASD known from most first-person and third-person PC games with space as the designed jumping button just as in most games within those genres. At the same time shift is as in most game used as the sprinting button.

The only other controls within the game were the combat system. This system was limited to a simple target and hit system to complete the ease of learning system. Therefore the combat system has been implemented as a tab-targeting system known from MMO-games like World of Warcraft and Star Wars the Old Republic. The tab targeting system switches targets by clicking on tab and the current target can be struck if it is within range.

Following this is a design description of each game with the respective game's filled out profile model. The games were designed to hit certain parts of the profile models and therefore the profile models have been filled out by the author.

First Game: The Open World

The first game is designed with a focus on World Exploration and Challenge Seeking. To fulfill the World Exploration element a large open world has been designed. This world has been filled with different landmarks to lure the player towards them. One landmark being a giant tree, one a big cathedral and the last one being a great forest. Each landmark have been filled with different challenges in the form of enemies requiring different tactics thereby fulfilling the challenge seeker parameter. The enemies was mostly designed by changing common values like damage and rotation speed and thereby as an example having an opponent that would quickly dispose of the player if they did not get in behind it. A screenshot from the first game can be seen on Illustration 12.

Illustration 12: Screenshot from The Open World

On Illustration 13 the model of the first game can be seen. Note that the model goes from -1 to 4 even though parameters only go from 0 to 4. The reason behind this is a simple one of visualization. Since the game affect very few parameters the chart can be quite hard to read without the extra value. As mentioned the first game is designed for the challenge seeker and the world explorer. The world explorer comes from the relatively big open world. Of course this world is not big compared to big titles, but when only having five minutes to explore the game the test participants won't see everything the world has to offer and it is therefore possible to give it a value of 3. Some of the enemies within the game requires some sort of tactics or the player will lose. Since these challenges might make some people stop playing it affects the challenge seeker parameter. Furthermore this tactic on some mobs makes the strategizer parameter 1. Again it is nothing compared to a real challenging game but considering the 5 minute limit the challenge seeker is set to 2. With the open world and the open run spaces there are opportunities for a bit of idling and therefore the Idleist parameter is set to 1.

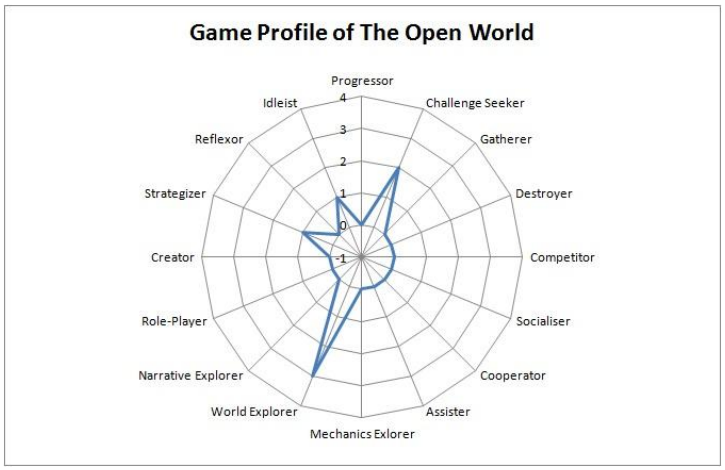

Illustration 13: Game Profile Model for The Open World

Second Game: The Closed Story

The second game is designed with a focus on Narrative Exploration and Progression. For the Narrative Exploration parameter to be fulfilled a story have been constructed. The story is shown in a small textbox in the bottom of the screen so it was visible but without being distracting for people careless about the story. Furthermore the game is linearly progressed through to fulfill the progressor parameter in as simple a way as possible. To affect the progressor parameter even further all enemies were made quite easy to remove any obstacles. This game was designed in a small indoor setting using closed corridors with only a single possible route at all times. A screenshot from the second game is shown on Illustration 14.

Illustration 14: Screenshot from The Closed Story

As seen from Illustration 15, the game fulfills the progressor to a 4 because of the very linear structure and the nature of one step at a time storyline. The story is for a small five minute game a strong little story and thereby this trait gets a 3. Because of the very closed environment and the very short and easy fights the game also affects the idleist parameter with 1.

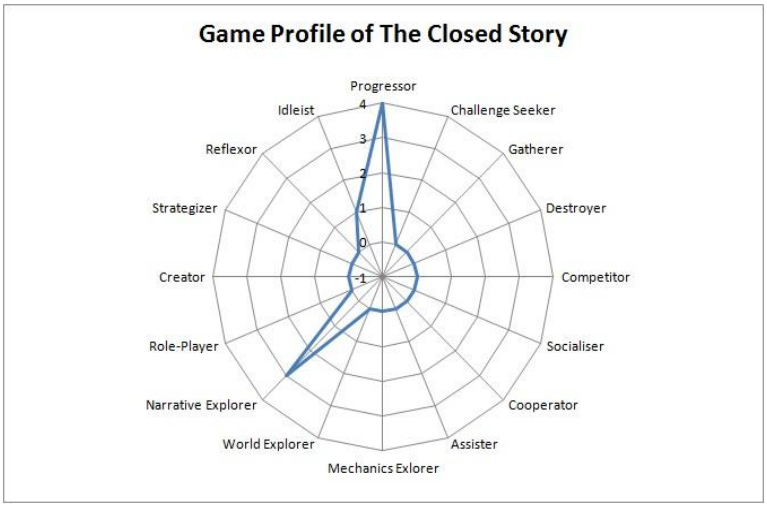

Illustration 15: Game Profile Model of The Closed Story

Third Game: The Arena

The third game has been designed with a focus on Competition and Strategizing. The game has been set up in a closed arena with hundreds of spiders being instantiated. Once a spider is instantiated, it will immediately walk towards the player to kill him or her. A timer will then count down from 300 and the player will get a point for each spider s/he kills. Meanwhile the score and a high-score are shown. The competition element is developed from the high-score which gives the player an incentive to beat something another player has done. This makes the game fulfill the product requirement about an asynchronous multiplayer game element partly. (The high-score is fixed but the player does not know that). The strategizing comes from the fact that the player has to find out which way they can most efficiently kill spiders without getting killed in the process. An illustration showing a screenshot from the third game can be seen on Illustration 17.

Illustration 17: Screenshot from The Arena

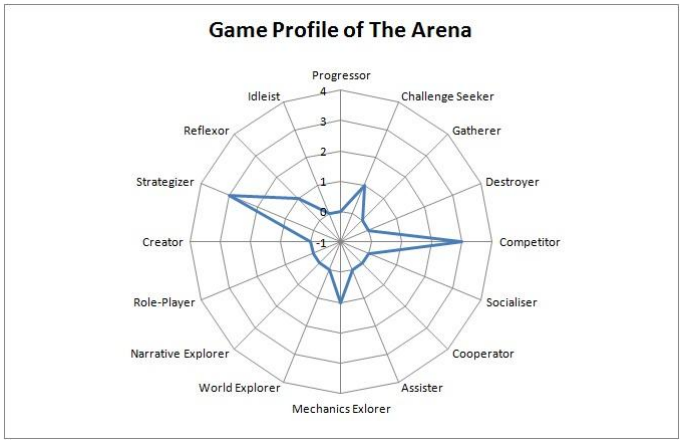

Illustration 16: Game Profile Model of The Arena

On Illustration 16 it is possible to see that the games competitor element has been ranked as 3. This is due to the fact that it is a competition element and that the scoring system will make test participants try to beat the high score. At the same time finding the right tactic for defeating spiders requires quite a lot of strategizing and thereby the game offers a 3 within that parameter as well. While those two parameters were designed upon the game also offers 1 in the reflexor since a player that can hit the spiders just before they hit him get more points without getting hurt. The challenge seeker has 1 point because of the fact that the spiders might overwhelm the player and kill him. And the 1 point within the mechanics explorer is gotten from the fact that the attack range is actually quite long in this game and thereby the observant eye will find out that it is possible to lure the spiders to a wall where they will be stuck and then the player will be able to kill them from a safe distance.

7.1.5 Summary

This chapter has discussed the general design decisions for the system, starting out with a choice of genre where the choice ended up on a Hack and Slash style game. Then an audiovisual strategy was created and with that the choice of a realistic graphics setting and no audio was created. Afterwards a strategy for making the three games distinct within the Player/Game Profile Model was created. This strategy was about making each game so the main focus were two aspects within the model that the other two games did not have as their focus. In the end the three game design parameters were “World Explorer and Challenge Seeker”, “Progressor and Narrative Explorer” and “Competitor and Strategizer”.

The Open World	World Explorer	Big open world
	Challenge Seeker	Harder Enemies
The Closed Story	Progressor	Linear Navigation Path, Easy Content
	Narrative Explorer	Linear Narrative Story
The Arena	Competitor	Scoring System
	Strategizer	Spider Killing Tactics

Table 7: List of Game Elements Fulfilling Parameter Aspects

Within Table 7, a summary of which game elements each game uses to affect the parameters they were created to effect. Note that the games are rated rather high in their parameters in regards to how they would fare against big triple-A titles; this is due to the limited size of the games and the limited time the game participants have within the game. The Narrative Exploration of the second game is of course not as detailed as the one in a triple-A title like Skyrim.

All this leads up to the Technical System Design where some of the most important prefabs and classes will be explained.

7.2 Technical System Design

This subchapter will describe the design of the bigger aspects of the system so it is possible for anyone to recreate the testing environment relatively easy. Furthermore the chapter describes the process created to fulfill the product requirements about a flexible and strongly organized system. This is due to the nature of the project where each new game created can be measured within the Game Profile Model and be compared to a person's Player Profile Model. The ideal goal was to reach a state where the implementation time of each new game were relatively low compared to setting each game up from scratch. Furthermore it makes it possible to expand upon a game and thereby make one game that could influence many parameters within the model. As mentioned earlier the tutorial gotten from (XX) was followed for several of the base scripts. Note that all design decisions are based on Unity as the game development toolkit with Mono Develop as the chosen IDE.

7.2.1 Instantiation

As the system itself is quite large, UML diagrams have been used to make it more manageable. Illustration 18 is an overall class diagram of the game setup. This diagram is the one that illustrates how the player, camera and mobs (short for monster) are instantiated. Note that not all properties and/or methods might be included in the diagram.

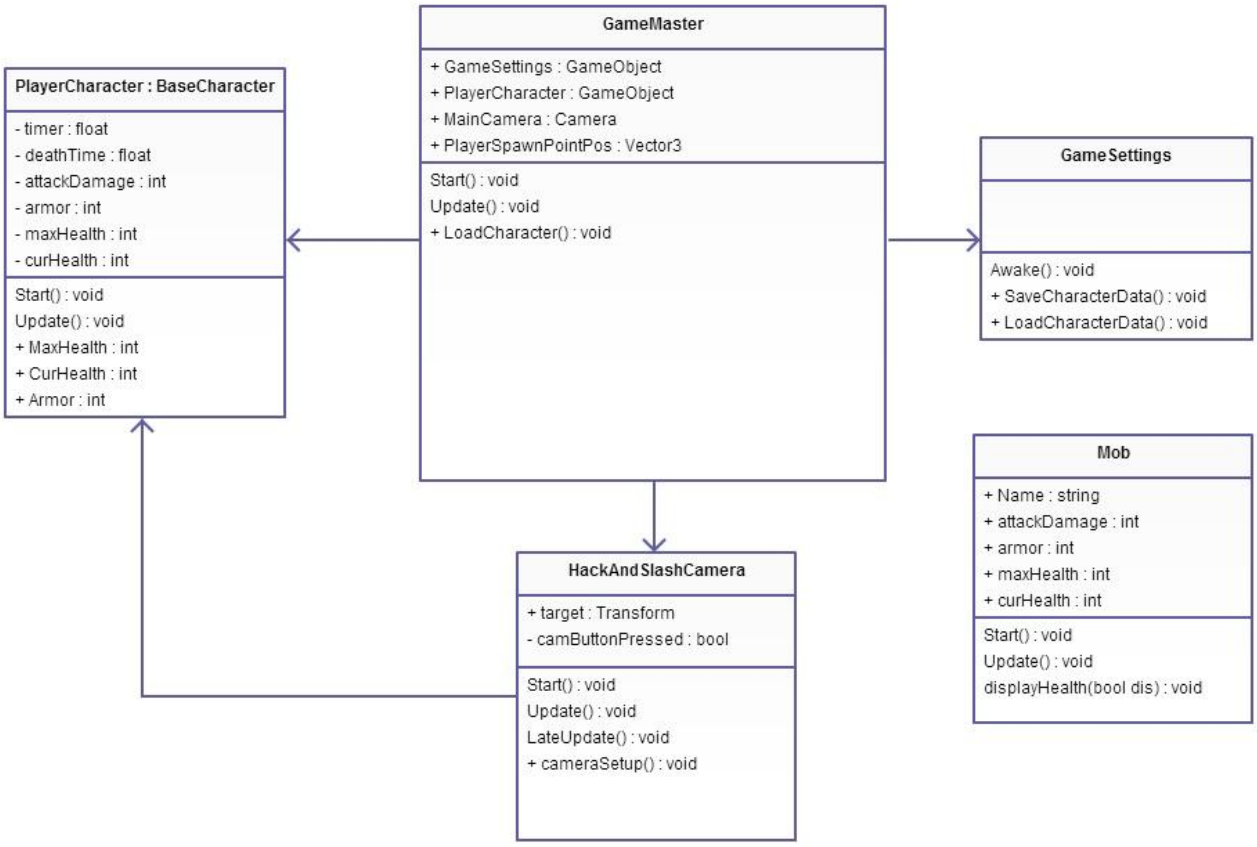

Illustration 18: Instantiation Class Design

To give a quick recap. The game master is responsible for instantiating the player character within the game world using the PlayerSpawnPointPos. Depending on a boolean set in the inspector the game master will also create an invisible object linked and floating above the player character.

If any data is saved from an earlier game setup the GameMaster will also Load that data. Note that the GameSettings are not affecting the used variables at the moment but is simply an extra feature saved for future development.

Furthermore the GameMaster will instantiate the HackAndSlash Camera and set it up for being the main camera. The Hack and Slash camera will then log onto the Prefab holding PlayerCharacter by using the tag “Player” to find the player prefab. If the earlier mentioned invisible object above the player have been created the camera will be positioned in regards to that while if it has not the camera will be positioned in regards to the player character behind the player.

Mobs are generally put into the game using their prefabs within the developing process and not during instantiation or during game play. Therefore the Mobs are handled outside the rest of the instantiation system.

7.2.2 Player Control

The UML diagram shown on Illustration 19 will show how the player character is controlled, as the last diagram some properties and/or methods might not be visible in the diagram.

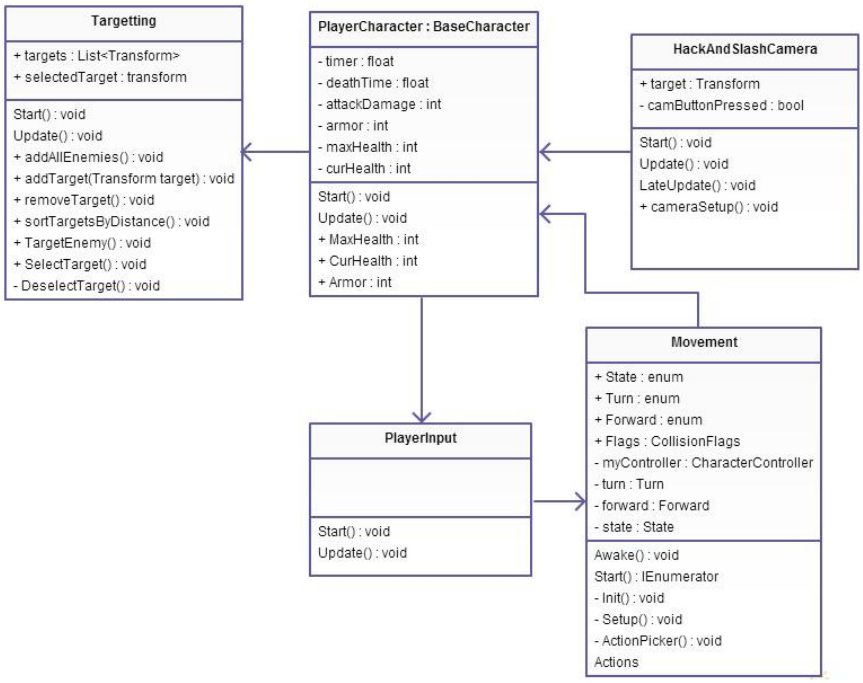

Illustration 19: Player Controlling Class Design

To sum up, as mentioned the player character is followed by the HackAndSlashCamera. If the camButtonPressed is true the player will be able to move the camera around the earlier mentioned invisible object. Once the camButtonPressed is set to false the camera will go back to its original position.

The PlayerCharacter furthermore uses the class Targeting which holds all transforms of all prefabs that have been labeled Enemy. The Targeting class controls all about these targets, the functionality for targeting them and the functionality for targeting the closest one.

At the same time the player character controls the PlayerInput that uses the built in Input function of Unity. The Inputs Update method simply looks for player Input and every time it finds a player input it changes a property within the FSM (Finite State Machine) class Movement. FSM is a mathematical computation model used to design sequential runtime logic.

Movement is a FSM that uses the values of its properties to affect the Prefab it is designated to with movement and animations. Movement is created in a way so that only the most basic animations are required and then other animations will only be called if the AI of the script calling it changes the values of the variables related to those animations.

7.2.3 Mob Behavior

Illustration 20 defines how mobs within the game work. In general it is a lot like how the player character is controlled but since a camera is not needed and since the mobs only have one GameObject that can be their target they won't need the Hack and Slash Camera script or the Targeting script.

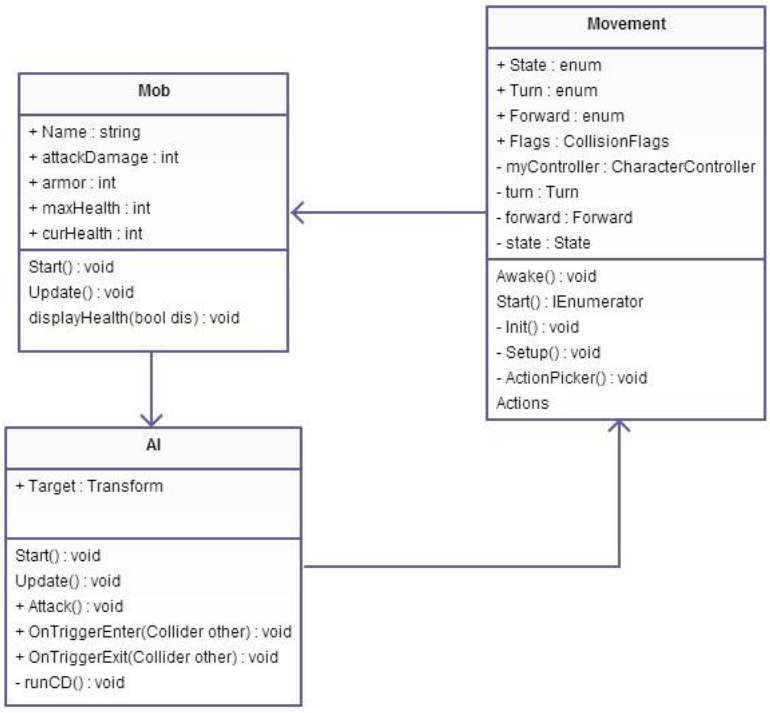

Illustration 20: Monster Behaviour Class Design

To sum up the mob has an AI which keeps a couple of Trigger Events. If the player character enters the creatures influence range the trigger event will set the player as the creatures target and the creature will enter a block inside its Update method. A graphical illustration of the influence range is shown on Illustration 21. At the moment the trigger event will look for a game object with the label

"Player" but could easily be further developed to look for different labels, and as such create a more dynamic setting.

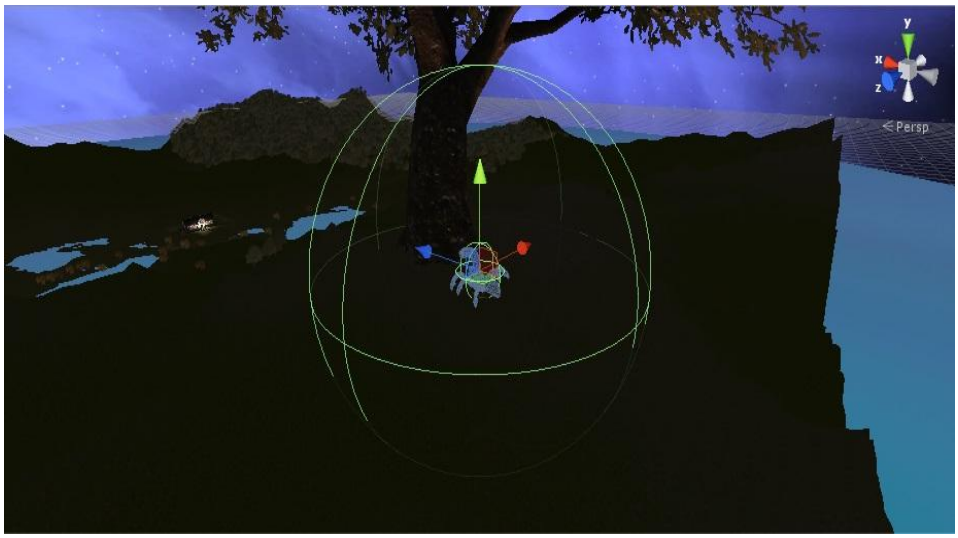

Illustration 21: Graphical Illustration of Sphere Collider

This block will then make calls to the movement script just as the Player Input class did. The movement script will then just as with the player Work as an FSM telling the Mob GameObject how to behave. Note that movement is designed so it can be used by both the mob and the player and any other instances of objects as long as they have the required animations.

7.2.4 Day and Night Cycle (Extra Feature)

The day and night cycle of the game is not a needed feature. However it is a nice one and one that has some interesting design decisions behind it and therefore the Illustration 22 explains how this cycle is designed.

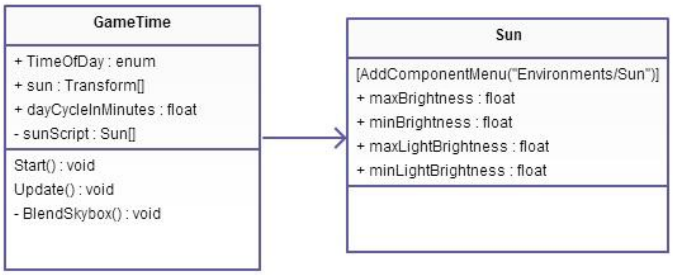

Illustration 22: Game Time Class Design

The GameTime script controls how long an in game day is. From this public variable it controls a skybox variable and

switches the graphics fluently for this skybox. For those not familiar with the term skybox, it is a graphical wrapper that lies around the entire scene showing the vast outside the game world. On Illustration 23 a scene without and with a skybox can be seen.

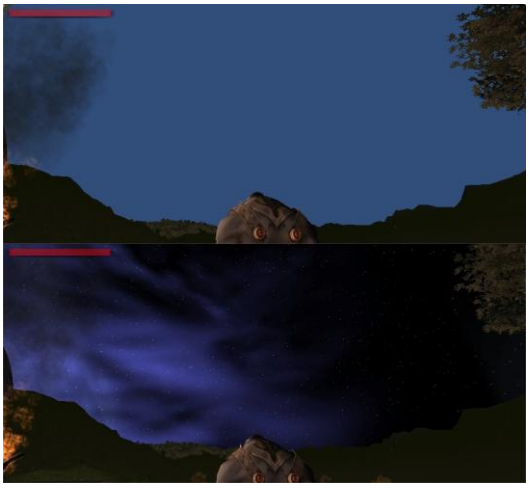

Illustration 23: Top Picture shows a Game without a Skybox. Bottom Picture shows a game with a Skybox

Furthermore GameTime keeps an array of Sun

objects which is simply a directional light source with the added features of the Sun implementation from unity. This light is then rotated around to make it seem like it is travelling around the world.

7.2.5 Individual Game Class

Some of the games need extra functionality which is shown in Illustration 24. The two games in question is The Closed Story and The Arena.

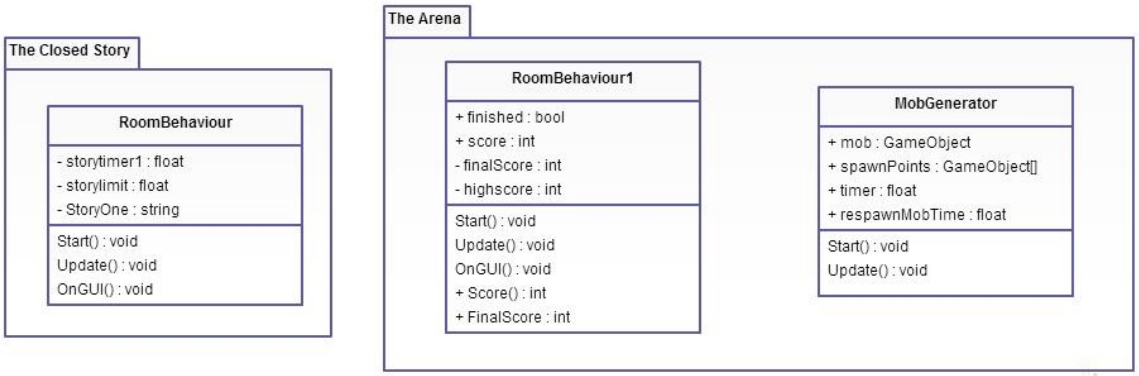

Illustration 24: Individual Game Class Design

The Closed Story needs a room behavior which can control when the different story messages should be displayed and also when certain game objects should be destroyed so that the player can progress onwards. Note that this class has many more properties and that the ones in the model are just to show the basic idea behind what the class does.

The Arena game need a RoomBehaviour as well to control the game timer and the scoring system. This game timer counts down from a starting value and until it reaches zero which is then the end of the game. The score is simply an integer which will have an addition of one every time an enemy dies and the final score will simply be set to the score at the end of the game. The high score is at the moment a predefined value, but with the implementation of a database it would be able to pull a high score from said database and add one as well.

At the same time The Arena is the exception of the rule that mobs are generated within the development tool. Instead the game needs creatures to be instantiated during game play. Therefore the game need a mob generator to handle this process. At this design level the mob generator simply holds a lot of spawn points and will then generate a mob every so often on a random designated spawn point.

7.2.6 Technical System Design Summary

This chapter set out to design a flexible and strongly organized system. It described the design of the system and the different important elements within. These included the base setup, player control, monster AI, day/night cycle and miscellaneous important classes for different games. With these elements it should be possible for anyone to design a relatively close

replica of the scripts used and use that to expand upon the experiment. The next chapter will describe the implementation of the system and how the most complicated aspects were created in more detail.

8 Implementation

With the previous chapter discussing the design of the system this chapter will describe the most important implementation decisions. Do remember that even though the design and implementation is split up in this report the process were a more iterative one, switching between design and implementation. Also do remember that a lot of basic classes have been implemented while following the tutorial made by (XX). For this project Unity was used as the game development engine. This decision was based on experience with that game development engine by the author and the fact that the Hack and Slash tutorial was made in Unity.

The chosen IDE for this project was MonoDevelop, due to the fact that the system organization is handled by Unity, the software development part of the project were mostly about writing the actual scripts. Therefore a more advanced IDE like Visual Studio were not necessary and MonoDevelop was chosen.

The chosen programming language to create the project fell upon C#. C# has a strong class library to support the programmer, which helps to create many functions. The author is experienced in several programming languages but the main experience lies within C#. Therefore since C# can fulfill any task needed by Unity it was chosen as the programming language.

As the complete implementation of the project spans over countless lines of code, obviously the entire implementation will not be covered in the report. As such, this chapter will try to outline the different algorithms and programming methods used within the project, so as to give anyone else, who has some knowledge of working in unity, the best chances possible of copying the product for further testing.

The entire implementation can be found on the attached DVD.

8.1 Game Master

The first method this chapter will describe is the Start() method of the GameMaster class. This is due to the fact that the game master is the top organizer of the system and because with its Start() method it instantiates the classes used in any of the games.

As seen on Code Snippet 1 the method begins by finding the player spawn point inside the GameSettings prefab, and if there is no spawn point it will create one.

```
void Start () {
    GameObject go = GameObject.Find (GameSettings.playerSpawnPoint);
    if (go == null) {
        go = new GameObject(GameSettings.playerSpawnPoint);

        go.transform.position = playerSpawnPointPos;
    }

    _pC = Instantiate (playerCharacter, go.transform.position, Quaternion.LookRotation(new Vector3(90, 0, 0))) as GameObject;

    _pC.name = "Player Character";
    _pCScript = _pC.GetComponent<PlayerCharacter>();
    _pC.transform.Rotate (0, 90, 0);

    if(CameraFocusOrNot)
    _cameraFocus = Instantiate (new GameObject (), new Vector3 (_pC.transform.position.x,
                                                                _pC.transform.position.y + 5,
                                                                _pC.transform.position.z),
                                Quaternion.identity) as GameObject;
    else
        _cameraFocus = Instantiate (new GameObject (), new Vector3 (_pC.transform.position.x,
                                                                    _pC.transform.position.y +1,
                                                                    _pC.transform.position.z),
                                    Quaternion.identity) as GameObject;
    zOffset = -2.5f;
    yOffset = 1.5f;
    xRotOffset = 12.5f;
    yRotOffset = 90;
    mainCamera.GetComponent<HackAndSlashCamera> ().target = _cameraFocus.transform;

    LoadCharacter ();
}
```

Code Snippet 1: Start Method of GameMaster Class

Afterwards it instantiates a Player Character on the spawn point within the game world. The player character prefab is added in the inspector to the public property. The reason for the rotation is a quick fix to make the playing character rotated in the correct direction.

The if/else statement decides where the focus point of the camera should be located. The CameraFocusOrNot property is defined depending on which scene the game is in. Then a new empty game object will be instantiated and the difference between the two methods is how high the game object should be placed.

Afterwards the offset for the camera is set and in the end the camera's target will be set to be the empty game object. In the end the character data will be loaded so a saved character can be loaded in.

8.2 Player Character and Vital Bars

With the instantiation described this subchapter will describe the functionality of the vital bars and the player character. This might not seem like the most important functionality within the system but it uses the functionality of the Messenger class found on a webpage written by (Wolffelt M., 2012) which is important to describe since it might be a possibility to use in new developed classes. Code Snippet 2 and Code Snippet 3 show the OnEnable() and changeHealthBarSize methods of the VitalBar class.

```
public void onEnable(){
    if (isPlayerHealthBar) {
                Messenger<int, int>.AddListener ("Player Health Update", changeHealthBarSize);
            } else {
                toggleDisplay(false);
                Messenger<int, int>.AddListener ("Mob Health Update", changeHealthBarSize);
                Messenger<bool>.AddListener ("Show Mob Vital Bars", toggleDisplay);
            }
}
```

Code Snippet 2: OnEnable Method of the VitalBar Class

```
public void changeHealthBarSize(int curHp, int maxHp)
{
    curBarLength = (int)((curHp / (float)maxHp) * maxBarLength);
    display.pixelInset = calculatePosition();
}
```

Code Snippet 3: ChangeHealthBarSize method of the VitalBar class

OnEnable checks whether the bar is a player or mob health-bar. If it is a player health bar it will look add a listener. This listener will listen for messages broadcasted with the name Player Health Update and two integers attached. If one such message is intercepted it will call the changeHealthBarSize method using the two integers. The changeHealthBarSize method simply calculates how long the health bar should be.

The process is nearly the same with the mobs health bar. However in that case the health bars display status will first be set to false, since that should be the default for all monsters' health-bars. Furthermore a listener will be added to check for a boolean. If a boolean with the value of true is passed from a broadcaster the current health bar will be displayed, until it receives a false message.

On Code Snippet 4 the Update method of the player character class is described. This method shows how to use a broadcaster that will work together with the listeners created in the before mentioned listeners.

```
void Update()
{
    if (curHealth < 0)
                    curHealth = 0;
    Messenger<int, int>.Broadcast ("Player Health Update", curHealth, maxHealth);

    if (curHealth <= 0) {
        SendMessage("IllDie", true);
        timer += Time.deltaTime;
        if(timer > deathTime && GameObject.Find("GameMaster").GetComponent<RoomBehaviour1>() == null)
            Application.Quit();
        else if(GameObject.Find("GameMaster").GetComponent<RoomBehaviour1>() != null){
            GameObject.Find("GameMaster").GetComponent<RoomBehaviour1>().FinalScore =
            GameObject.Find("GameMaster").GetComponent<RoomBehaviour1>().Score;
            GameObject.Find("GameMaster").GetComponent<RoomBehaviour1>().finished = true;
            GameObject.Find("GameMaster").GetComponent<RoomBehaviour1>().destroyMobs();
        }
    }
}
```

Code Snippet 4: Update Method of the Player Character Class

The update method starts out by setting current health to 0 if it is below 0. The reason for this is so that the vital bar won't suddenly be displayed in a negative direction. Afterwards a message is broadcast carrying two integers. Note that the name of this broadcast has to match the name of the receiver taking the message.

Afterwards if the current health of the player character is equal to or below 0 the player character will use a SendMessage method to tell the Movement class that it should play the movement animation. This method will be described more in detail later in this chapter.

Afterwards a timer will start, if the scene is the arena scene the game will instantaneously enter the scoring screen. To check whether the current scene is the arena or not it simply searches for the class RoomBehaviour1 within the GameMaster object. If it is not the GameMaster object the application will quit once the timer has run out.

8.3 Player Input

With the messaging system handled this subchapter will describe how the SendMessage method mentioned before, works and also how to use the input manager of unity. Code Snippet 5 shows a part of the PlayerInput class' Update method. Herein lays both SendMessage calls and Input calls.

```
void Update () {
    if(Input.GetButton("Vertical"))
    {
        if(Input.GetAxis("Vertical") > 0)
            SendMessage("MoveMeForward", Movement.Forward.Forward);
        else
            SendMessage("MoveMeForward", Movement.Forward.Back);
    }
    if(!Input.GetButton("Vertical"))
       SendMessage("MoveMeForward", Movement.Forward.None);

    if(Input.GetButton("Horizontal"))
    {
        if(Input.GetAxis("Horizontal") > 0)
            SendMessage("TurnMe", Movement.Turn.Right);
        else
            SendMessage("TurnMe", Movement.Turn.Left);
    }
    if(!Input.GetButton("Horizontal"))
        SendMessage("TurnMe", Movement.Turn.None);

    if (Input.GetButton ("Run")) {
                SendMessage ("ToggleRun", true);
        } else
                SendMessage ("ToggleRun", false);
```

Code Snippet 5: Part of the PlayerInput Class' Update Method

The Update method is a long list of if statements which checks for different button inputs. If a button input equals true it will use the appropriate SendMessage to tell movement what action to take. SendMessage simply sends out a message to all classes within the same game object as the one PlayerInput resides in. If any class for example holds a method called ToggleRun that takes a boolean as input that method will be called with the input sent from SendMessage.

Illustration 25: InputManager as seen from the Unity Inspector

The different Input options can be seen and edited within the inspector. Illustration 25 shows the input manager and a list of methods. When creating a new one there are two options, either add an input to the end of the list by increasing the size, or rename one of the already created inputs.

With the SendMessage and Input functionality explained, the next subchapter will describe the movement class.

8.4 Movement, FSM

As mentioned this subchapter will go into details about the movement class, how it works as an FSM and how it can work with both the AI class and the PlayerInput class depending on whether it lies in a monster or a player game object. This is due to the nature of movement being the most clear example of a strongly organized system, and due to the fact that it is an example of how to create an FSM. Code Snippet 6 shows the Start method of the Movement class.

```
// The FSM
IEnumerator Start () {
    while (true) {
        switch(_state){
        case State.Init:
            Init ();
            break;
        case State.Setup:
            Setup();
            break;
        case State.Action:
            ActionPicker();
            break;
        }
        yield return null;
    }
}
```

Code Snippet 6: Start Method of the Movement Class

This start method will keep running and due to the nature of the IEnumerator and the yield within the while loop it will run every frame. The starting state will be set to Init and each state will then make the State property into the next one until the state is set to Action. Once it is every frame it will call ActionPicker. That method can be seen on Code Snippet 7 and will be described below.

The ActionPicker method will in general check whether to call different methods by using the parameters which have been set by either the PlayerInput class or the AI class.

```
private void ActionPicker(){
    Rotate ();
    if (myController.isGrounded){
        moveDirection = new Vector3(0, 0, (int)_forward);
        moveDirection = myTransform.TransformDirection(moveDirection).normalized;
        moveDirection *= moveSpeed;

        if(_attack && _die == false){
            Attack ();
        }

        if(_forward != Forward.None && _die == false){
            if(_run){
                moveDirection *= runSpeed;
                Run ();
            }
            else {
                Walk ();
            }
        }
        else
        {
            Idle();
        }

        if(_jump && _die == false){
            moveDirection.y += jumpHeight;
            Jump ();
        }

        if(_die){
            Die ();
        }
    }
    moveDirection.y -= gravity * Time.deltaTime; //Makes sure the world has gravity
    if (_die == false) {
                _collisionFlags = myController.Move (moveDirection * Time.deltaTime);
    }
}
```

Code Snippet 7: ActionPicker Method of the Movement Class

First it will always call the Rotate method which will rotate the character if needed. Afterwards a check is made to check whether the controller is grounded. This is so it is not possible to move or jump in midair.

Afterwards a vector symbolizing which way the character will move will be created. Afterwards the ActionPicker method checks to see whether the character is currently set to be doing any movement like Attack, Run, Walk, Idle, Jump or Die. If so the method's calling the animations of the character are called. The last couple of lines make certain that the character is affected by gravity so it can't float around in space and that the character moves if it is not dead.

8.5 Mob AI

The last part of the implementation is the Trigger methods inside the AI class. These are used to find out whether the player gets close to an enemy and if so awakes that enemies AI.

```
public void OnTriggerEnter(Collider other){
    if (other.CompareTag ("Player")) {
        target = other.transform;
    }
}

public void OnTriggerExit(Collider other){
    if (other.CompareTag ("Player")) {
        target = null;
    }
}
```

Code Snippet 8: Trigger Event methods of the AI Class

The two methods shown in Code Snippet 8 are used by the sphere collider of the object. If another game object enters the sphere collider OnTriggerEnter will be called and if it leaves OnTriggerExit will be called. Both methods checks whether the object entering or leaving has the tag player. And if so will set the target to that object or remove it. Inside the mobs update method nothing will happen unless the mob has a target so in some way the mob is awakened by the nearby presence of a game object with the tag of player.

8.6 Summary

This chapter has described some of the implementations of the project. Parts of the classes GameMaster, PlayerCharacter, VitalBar, PlayerInput, Movement and AI have been explained. Thereby the more advanced and confusing concepts have been explained. With the design and implementation chapters complete it should be possible for anyone with Unity experience to duplicate the product creation process and thereby be able to test the problem statement themselves. With this chapter complete the next chapter will describe the test process and the results of said test.

9 Test

With the two previous chapters concluded and thereby the explanation of the design and implementation concluded this chapter will describe the test of the problem statement. The goal was to compare a player's player profile model with that players rating of different developed games each with their own model.

Null-Hypothesis: *There is a difference between how a player will rank two created games and how closely that player's profile model is correlated to said games' profile models.*

As mentioned back in chapter 5.3 the player model has to be filled up with as precise data about the player as possible. Therefore a preliminary test section will be performed to ensure that the player profile model is as precise as possible before it is compared to the game models and the rating of the games.

9.1 Method

Both the preliminary test and the final test is completed in one quantitative test. This made the testing procedure for the tests as a whole last only a small week. The test used a convenience sampling method with the added condition that people would have had to have played games recently. The reason behind using convenience sampling is that the model should be able to work for all players and as such using an extensive amount of time on an ethnographic limitation would not benefit the test. The convenience sampling is followed by sending out the developed games and the questionnaire used for testing (Rueløkke J., 2014) on different game forums, facebook groups related to game development and gaming, and to friends and family frequently gaming. This approach added a spread over different kinds of gamers to the test participant group as can be seen in the later mentioned player profile models. The answers from the questionnaire can be found on the DVD inside the folder results. The reason it is not in the appendix is because of the table sizes. If the tables were split up to make them fit inside an A4 page much of the understandability would be lost. The test is done as within subject test design. The within subject test design was followed by sending each test participant through the exact same process to make sure that conditions for the results were the same. Thereafter using the test data to find distinctions between subjects rather than groups.

As mentioned two tests have to be made. The first test is performed to ensure that the way players input data into their player profile model skewer with the results of the final test as little as possible. The first test is created as a base for finding out which parameter inputting strategy, described in (5.3 Adding values), should be used. If the used method of adding data to the player profile is not precise enough the created player profile model will not be valid and therefore usable by the final test.

The second and final test tries to answer the problem statement by investigating whether players will rank the developed games in the same ranking order as they should from what their player profile model tells. The reason to use a ranking instead of a rating is the fact that because it is small developed games people might rate them from different criteria. Some might rate them from the criteria that they are small prototypes, while others might rate them compared to triple-A titles thereby corrupting the test data. The ranking system is therefore used to remove this bias. To find this relation two calculations will have to be iterated over for each test participant. The first calculation will have to compare two player profiles or in this case a player and a game profile. The second calculation will have to turn this comparison into a rating of the game.

9.2 Procedure

As mentioned both tests have been conducted on the same test participants with both tests being conducted in the same test setup. Each participant starts out by answering a few simple demographic questions; afterwards they create their own player profile. As a help the test participant was asked to look at the pdf file attached to the folder with the three games. The pdf file contained a short explanation of each parameter and can be seen in appendix 13.3 Explaining The Player Profile Model Parameters. The player profile are as mentioned earlier the player's own personal filled out player profile model. While the game profile is a model filled out for the game. One such player profile can together with a game profile be seen on Illustration 26.

As the next step the test participants are asked to play each of the three games. Since the arena game does not give for test participants to get used to the control any test with that as the first played game would be problematic. Therefore it was decided to stay away from randomizing the order of the games being played. After each play session the users were told to rate the game from 1-10 and give a description of the reason behind the rating.

After they have played the three games the test participants are asked to name a game they play at the moment and fill out a game profile for said game. This step has to be completed for three games.

In the end the test participants have been be asked to rate 23 commonly known games from 1-10 or if they haven't played the game chose an added option called haven't played. The games are chosen from a list of different well known games that effect different parameters within the player/game profile model. Each game profile have been chosen based on personal objectified experience by the author. This has been done by extensive game play with at least twenty hours spent on all but the most simple of the games. This game play can be from years back but with an extensive knowledge about the game and an objectified look it have been possible to give fair values to the games within the model. If a number in a game is not satisfying to the reader it is possible to open the result spreadsheet on the DVD and change that number and everything from that point in the results will be changed accordingly.

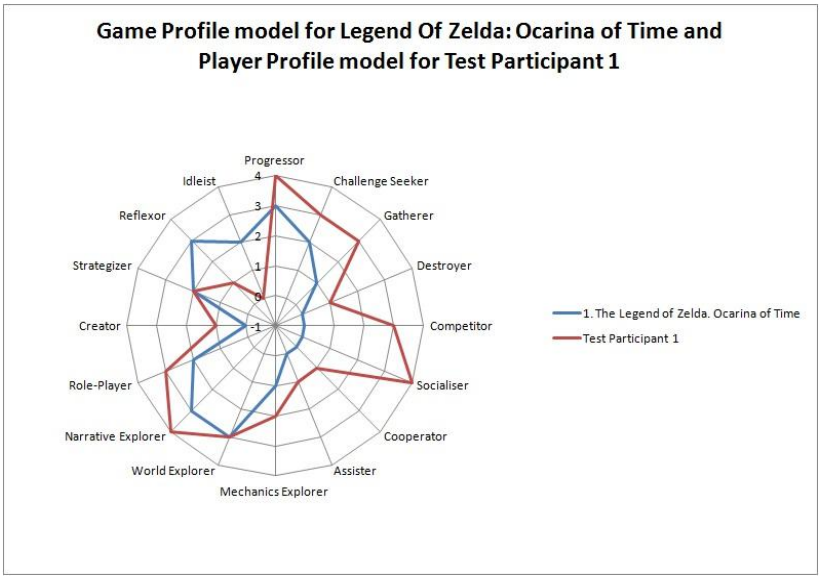

Illustration 26: Comparison between Test Participant 1 and The Legend of Zelda within the Player Profile Model

The following step will explain the procedure of the preliminary test in general while following a concrete example.

The preliminary test uses the information gathered from the rating of the commonly known games together with the players self defined player profile, and the same players chosen game profiles.

The gathered data is then calculated, and analyzed upon. The first calculation to do is to take the players three chosen game profiles and calculate them into a second player profile. By having the second player profile the two can be compared on all parameters which is the reason to create it. This is done by finding an average for each parameter over the three games and adding that average as the value of the player profile. In Table 8 it is possible to see the creation of this second player profile which from this point forward will be mentioned as the calculated player profile while the one the player chose all by themselves will be named self defined player profile.

Parameters	First Chosen Game	Second Chosen Game	Third Chosen Game	Player Profile Model
Progressor	2	2	3	2
Challenge Seeker	2	3	3	3
Gatherer	3	1	3	2
Destroyer	0	1	0	0
Competitor	0	4	1	2
Socializer	0	0	0	0
Cooperator	0	3	2	2
Assister	0	2	1	1
Mechanics Explorer	1	1	1	1
World Explorer	3	0	3	2
Narrative Explorer	4	0	3	2
Role-Player	0	0	0	0
Creator	0	0	2	1
Strategizer	1	3	2	2
Reflexor	2	3	2	2
Idleist	0	0	0	0

Table 8: Creation of second Player Profile Model for test participant 24

Afterwards the total difference between a player profile model and each of the game models is calculated. This is done by going through each parameter and if the player profile has the highest value the difference will be added to a total. If the game has the highest value half of the difference will be added. The reason for making a difference between these two cases is that if a player wants some type of game play and a game does not offer it, it is worse than if a game offers a type of game play that the player does not want. The player will often be able to avoid that type of game play. However the reason to add anything when a game offers more than a player wants is because with a lot of building aspects in a game the player might feel forced into playing a way he or she does not want to.

Parameters	Self Defined Player Profile	Calculated Player Profile	Word feud Game Profile
Progressor	2	2	0
Challenge Seeker	2	3	3
Gatherer	3	2	0
Destroyer	0	0	3
Competitor	2	2	3
Socializer	1	0	1
Cooperator	3	2	0
Assister	1	1	0
Mechanics Explorer	0	1	0
World Explorer	3	2	0
Narrative Explorer	3	2	0
Role-Player	0	0	0
Creator	2	1	0
Strategizer	2	2	4
Reflexor	2	2	0
Idleist	1	0	4

Table 9: Player Profiles for Test Participant 24 and Game Profile for Word feud

On Table 9 it is possible to see the player profile models of Test Participant 24 and the game profile of Word feud. The difference between the self defined player profile and Word feud's game profile is 24 and the difference between the calculated player profile and Word feud's game profile is 20,5.

Once this has been done the next step is to calculate these differences into a rating of 1-10 where the higher the difference the lower the rating. The highest difference possible would therefore be if a game had 0 in all values and a player had 4 in all values, in which case the difference would be 64. With 30 people and 23 games it would be unrealistic to think that no one would give a rating of 1 or 10 to any single game. Furthermore it is unrealistic that participants will be the maximum possible distance away from any games or that they will have 16 parameters in common with a game. Therefore calculation that can keep the calculated rating scope between 1 and 10 will have to be created. To do this all differences between player models and game models are used. The minimum distance is found and subtracted from all distances. Thereby the minimum distance will now be zero. Afterwards the max distance is found to find the range of distances that the ratings will have to be designated within. The formula for finding the predicted ratings is then.$Rating = 10 - \left(-\frac{ProfileDifference}{HighestProfileDifference*1.1}\right) * 10$ As seen the calculation starts out with a rating of 10 and then subtracts from that. The rest of the formula is a simple percentage calculation with the twist that it has to multiply the highest profile difference with 1.1 to ensure that the rating can't be zero.

To continue with the calculation on test participant 24 the lowest and highest value will have to be added. Note that these values are found from searching through all differences between all players and games for each method. The lowest value for the self defined player profile difference is 7,5 which brings the difference down to 16,5, while the lowest value for the calculated player profile difference is 5 which brings the difference between the player profile and Word feud's profile down to 15,5.

With the new values it is possible to find the rating by adding the highest difference in each model to the earlier mentioned formula. The highest difference for the self defined player profile model is 33 while on the generated player profile it is 42,5. Thereby the calculations are

$$RatingSelf = Rounded(10 - \left(-\frac{16,5}{33*1,1}\right) * 10)$$
$$RatingGenerated = Rounded(10 - \left(-\frac{15,5}{42,5*1,1}\right) * 10)$$

While RatingSelf is equal to 5 RatingGenerated is equal to 6. Test participant 24 gave Word feud a rating of 6. So while one method hits the correct rating the other method is one away from the target.

To be able to see how precise the player profiles have been in rating games the same way the test participant did an average for each test participant has to be calculated. This is done as a way to find a difference between the created player profile model and the participants' actual opinions about games. After that the average of those differences for each person is found. In the end the mean and the standard deviation of all the averages is found for both methods to find out which one is more precise.

The second test will be processed quite like the first one. First the method chosen from the first test is used to predict how each player would rate the three developed games. For each person the ratings will then be gone through and the highest rating will get a ranking of 3, the second highest 2 and the lowest 1. If two games are rated equally high they will get the same ranking and the next ranking will be one lower. Each game comparison is then looked through. This process is shown in Code Snippet 9 which is an Excel statement.

```
=IF(AND(AT131>AU131; AT35>AU35);1;IF(AND(AT131<AU131; AT35<AU35);1;IF(AND(AT131=AU131; ABS(AT3-AU3)<2);1;IF(AND(AT35=AU35; ABS(AT99-AU99)<2);1;0))))
```

Code Snippet 9: Game Ranking Calculation

Simplified this statement reads the following statements

- IF(A1>B1 and A2>B2) return true;
- ELSE IF(A1<B1 and A2<B2) return true;
- ELSE IF(A1=B1 and Math.Abs(A3-B3) < 2) return true;
- ELSE IF(A2=B2 and Math.Abs(A4-B4) < 2) return true;
- return false;

This statement checks the rating difference on the players rating choices. At the same time it checks the difference on the ratings the player profile would create. If the two models agree about which game the player has rated and should rate the highest the method returns true. If not it returns false. If one of the sides have the same ranking for two games the ranking would have to be unfairly precise therefore the equal operators are meant to make a small room for errors so that a rating difference of one can also be taken as the same ranking. When all comparison pairs have been calculated it is possible to gather results to check whether the model were successful in predicting game ranking.

9.3 Results

The following subchapter will outline all the results gathered from the different tests. However it is not possible to show all results and midways calculations here. For that look for the .xlsx file appended on the DVD called Results. The results are not in the appendix because of the table sizes which are 30x16 cells big. These tables would therefore have to be split up to fit inside an A4 page, and this would destroy the understandability of the different tables.

9.3.1 Preliminary Test

The goal of this test was to find out whether players should choose their own player profiles or whether it should be created by how they play their favorite games.

The first method which is explored, looks at the difference between the ratings given by test participants to a list of 23 games and the rating the participants self defined player profile should give. On Illustration 27 it is possible to see a histogram of the average distance between the given and predicted ratings from the test participant's self defined player profile.

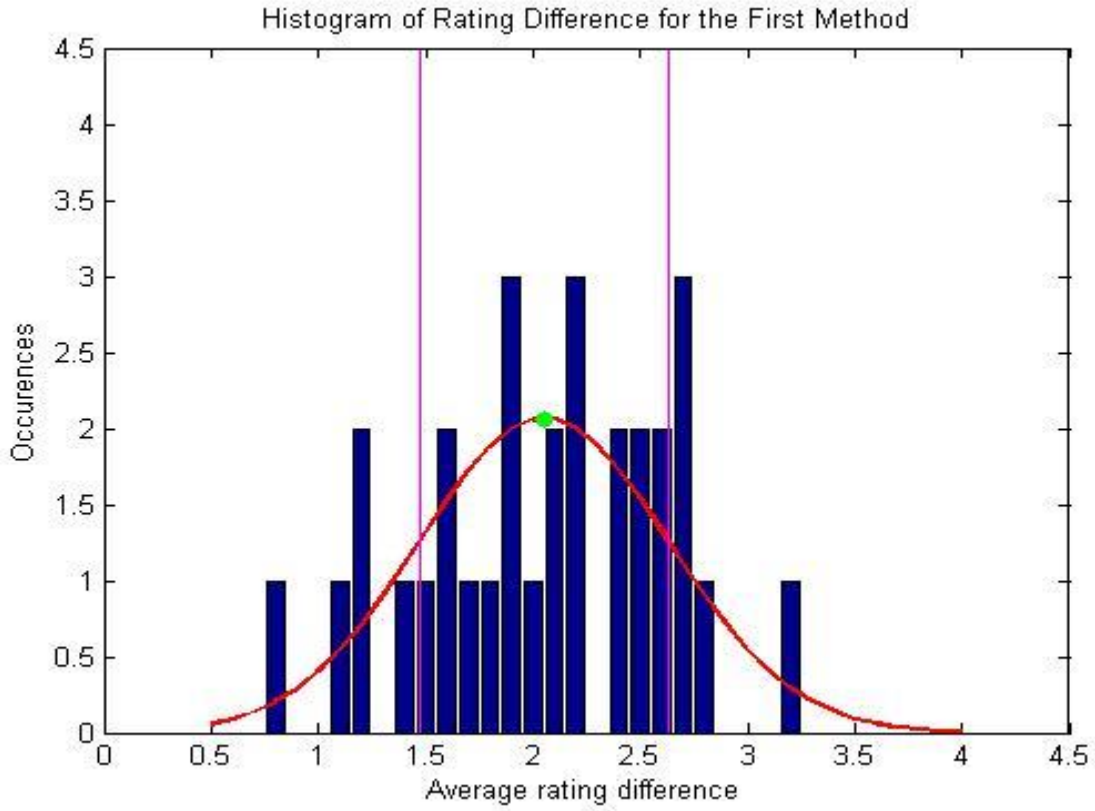

Illustration 27: Histogram of Rating Difference between the Actual Ratings and the Predicted Ratings based on the Self Defined Player Profile

The average amount the first method rated wrong was 2,05 and the ratings were as mentioned earlier given and calculated from 1-10. The respondent that had the closest distance between the two rating sets had a distance of 0,83 average. Meanwhile the respondent who had the

predicted rating that missed by the most in average missed by 3,21 in average. The standard deviation on this data is 0,5808.

The second method which is explored, looks at the difference between the ratings given by test participants to a list of 23 games and the rating the player profile, calculated from the test participants favorite games, should give. On Illustration 28 it is possible to see a histogram of the average distance between the given and predicted ratings from the player model calculated based on favorite games. Again each count within the table is an average of a test participant's difference in ratings between predicted and actual ratings.

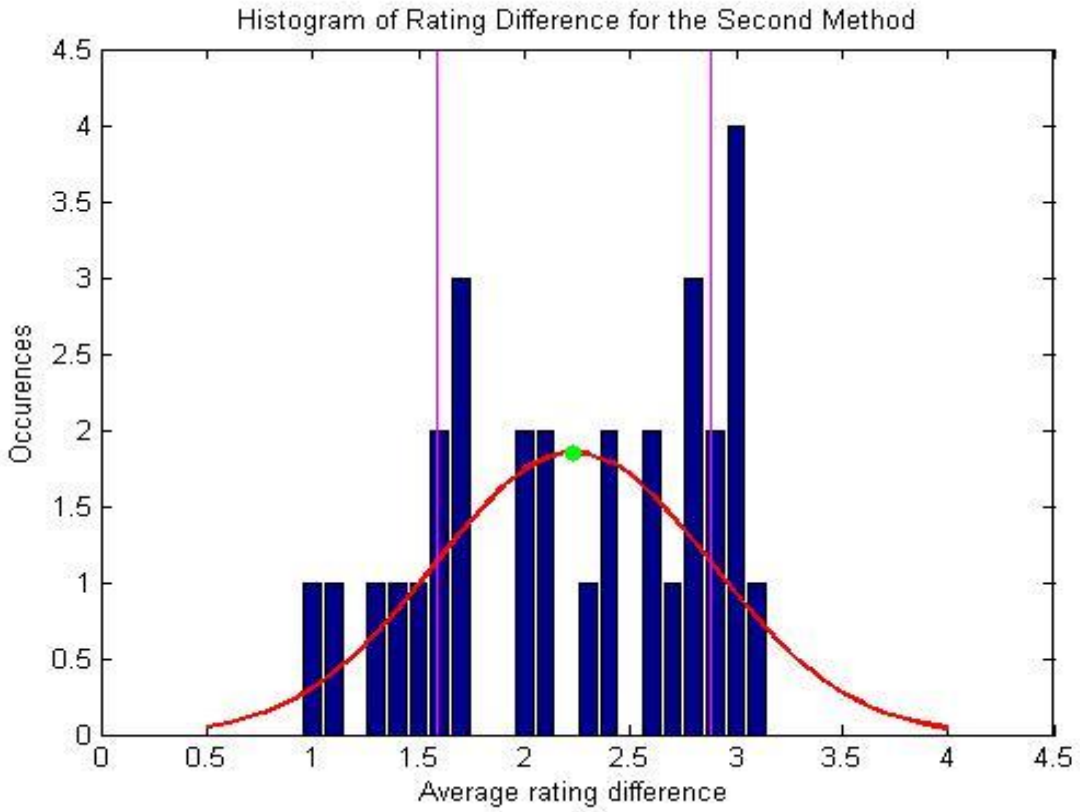

Illustration 28: Histogram of Rating Difference between the Actual Ratings and the Predicted Ratings based on the Player Profile Generated by Favorite Participants Favorite Games

The average amount the second method rated wrong was 2,2 and the ratings were as mentioned earlier given and calculated from 1-10. The person where the model rated closest was 1,00 from the given rating in average and the person where the model was the furthest away in predicted rating was 3,14. The standard deviation on this data is 0,6485.

From the above results the parameter input method of letting test participants chose the parameters for themselves was decided upon for the final test. This was due to a slightly less error margin in average and that the errors were a small amount less spread out but stayed around 2 in the average distance.

With a scale of 1-10 an average error margin of 2 might seem high. In chapter 10 Discussion some of the reasons behind this error margin will therefore be explained.

9.3.2 Final Test

The final test was created to answer the null hypothesis. Therefore the created games would have to be ranked against each other. To do this the rating that test participants gave the games were used as a base for the rankings. In Table 10 is an example of four different scenarios in how the ranking were followed.

	Game 1	Game 2	Game 3
Rating Scenario 1	7	9	3
Rating Scenario 2	4	5	3
Rating Scenario 3	3	3	4
Rating Scenario 4	8	5	8
Ranking Scenario 1	2	3	1
Ranking Scenario 2	2	3	1
Ranking Scenario 3	2	2	3
Ranking Scenario 4	3	1	3

Table 10: Examples of ratings being turned into rankings

As seen from the table the amount one game is above another does not matter for this test, only the fact that it is above matters for the ranking. The table also show what happens when two games get the same rating. If they do they both get the next rank left and the next game after those two gets two ranks lower than the two equally ranked ones.

The job of the player profile model is then to predict whether for example Game 1 will be rated higher than Game 2 by test participant 13.

First games 1 and 2 were compared against each other. Here 30 participants had ranked both games. Of those 30 participants the predicted model had guessed 11 of their ranking correlations which gives a precision of 36,67%.

In the comparison between games 1 and 3, 30 participants also ranked both games. Of the 30 people the Profile model were able to guess the correct ranking correlation in 20 of the cases which is 66% of the 30.

Finally between games 2 and 3, 30 participants were also ranking the games. Of the 30 people the Profile model were able to guess 15 of their ranking correlations which is a 50% correct guess rate.

	Game 1 and 2	Game 1 and 3	Game 2 and 3
Lower than	24	19	11
Equal	2	5	4
Greater than	4	6	15
Predicted Lower than	2	11	19
Predicted Equal	11	11	9
Predicted Greater than	17	8	2

Table 11: Shows whether the first named game is ranked lower than, equal to or greater than the second game named

From these data it would seem that game 1 has an error that affects the results therefore a final calculation were made so it could be used in the upcoming discussion. This final calculation uses the rating data. It counts how many respondents prefer game 1 above game two. It also gets the number of respondents that should like game 1 above game 2 according to the player profile model. The calculation does this for all comparisons between the three games and those comparisons can be viewed on Table 11.

As seen the best percentage of correct predictions were 66%, which is not enough to disprove the null-hypothesis.

9.4 Summary

This chapter have described the testing process. As seen from the results the model has not been precise enough in the final test to be able to disprove the null hypothesis which from the test requirements needed a precision of 80% to be approved by the author. The discussion chapter that follows will discuss the project as a whole and thereby also which effects could have influenced the results in such a way.

10 Discussion

To review the results, a summary of the analysis, the design, the implementation, the testing process and their strength's and shortcomings, is necessary

First off, the analysis discussed and compared the most important research articles within the subject of player types. Furthermore this was complimented by information gathered from focus groups and experience from everyday life. In general the created model covered the subject of how people play games very well.

The first change that might be required from within the analysis is the inclusion of Fog's Destroyer. It is still a discussion whether Fog's destroyer is a significant enough group that it should be included in the model, but the author of this report would like to acknowledge that the problem of missing coverage is there. Since none of the games affected Fog's destroyer parameter in any way this missing category would not have any affect on the test results.

The second part of the analysis that has to be discussed is the choice of representation chart. As mentioned in the chapter discussing the subject, the radar chart was chosen especially because of its compressed structure compared to the bar chart. As long as the chart will only have to show one or two profile models simultaneous, it should still be the strongest chart for visualizing player types. But for each profile model shown simultaneous the radar chart gets worse and worse at displaying the information and at four or more models the bar chart should be used since it is stronger at showing many models. This is a problem for the visualization but not a problem that affected the test results in any way.

The last and biggest error within the analysis has to do with the conceptualization of a player type as a whole. In the analysis it was quickly distinguished that the report would focus on making a model of how people play and leave the subject of why people play to future perspectives. While this is a good approach the report should have remembered other aspects that are important when the subject is on whether players like games or not. For example both World of Warcraft and Star Wars The Old Republic have a Narrative Exploration element of 4 with big open stories that can be discovered and investigated. However a player might rate one game 1 and another 10 because they do not like the Sci-Fi setting versus the fantasy setting or the opposite. These are not within the how we play model, but can be seen as reasons why the model is unable to predict how well people like different games better than it is.

The design chapter was split into two parts, one designing the games in general and one designing the system. The system design is strong, organized and flexible so it would be easy to further develop on the game for future experiments. However at the same time the game design can be an influence into the test result for two reasons.

The first reason is the way the three games fill out there model. Even though they each influence two parameters greatly that the other games does not, the fact is that none of the models affect more than 5 parameters on the X-axis. This could make the three games predicted ratings be relatively equal and as seen from Table 11 more than a third of the games are ranked equal. It would be hard to change this fact without a significant increase in production power since each parameter added will have to be designed and implemented for the game it is added to. It is therefore acknowledged that this is a bias, but it is not a bias that can be fixed at the current state of the project.

The other big problem lies with the hard requirement of ease of learning. Most of the system seemed to be fairly easy for test participants to use, however one mechanic left most participants frustrated. The targeting system was not well enough designed or implemented and therefore many people had a hard time figuring out how to attack enemies within the games. This might have created a negative bias on ratings with especially the first game since it was the game where people had no experience with the targeting system. The order of the games will be discussed in just a bit.

The implementation of the product followed the design and was thereby a strong part of the project. The only problem with the implementation was the targeting system, but as it was also mentioned this was more a design problem than an implementation problem.

The last part discussed in this chapter, before discussing the project overall, is every aspect of the test. The first part has to do with the preliminary test. As seen in the test chapter the player profile model directly chosen was a bit more precise than the model created from the player's favorite games. This could be the cause of three games not being able to cover a player's interests within gaming enough. However adding more games would make the test time longer for each game, which makes it a fine balance on how many games could be added.

As mentioned earlier in the chapter participants had many problems considering the combat system. Therefore a usability test rotation with focus on ease of learning during the implementation process would have been beneficial as that would have been able to catch the errors within the targeting system. The usability problem was one affecting all three games however especially the first game were from the comments influenced by the missing understandability. Thereby fixing the usability the first game would perhaps get better ratings and thereby also be aligned more with the model.

The three games were all tested in the same order for every test participant. This meant that it was the same game that every test participant used to get used to the controls. This might have hurt the rating of the first game. Therefore a randomized order of the experimental conditions could be considered. The problem with this would however be that The Open World does not have a lot of combat while The Arena is all about combat. So if it hurts the ratings of The Open World to be the first tested game then the amount of problems for people playing The Arena as the first game would make it impossible to use that data. Therefore there are two options to solve the problem which should be combined. First of all the targeting system should be upgraded to a more user friendly version. This should be combined with a training wheel game. This game should not be used for test results but should solely be used to learn the player the game thereby removing the bias.

In general the model were fulfilling of all the other reviewed studies with some explained differences. However it was not possible to create a model that could be used to predict whether a player would like a game or not. It was however possible to add a model to the current research area that adds some new ideas to the research field of player types. First of all the models most important feature compared to other models is that it distinguishes between how and why people play games and focuses on how people play. This distinction is an important one, since the two effects are so very different. Ask yourself how you do anything and why you do it and the answer will not often be the same. Because of this distinction it is possible to add both games and players within the model which no other model seen, have been able to do. While it did not mean the model could be used alone in regards to the problem statement, it could still be used to influence game design as a checklist for both researchers and game developers alike.

While the project did not prove a similarity between how players rate games and how they play games in general, it was an ambitious project that opened the author's eyes to the difficulties of creating a functional player profile model.

11 Conclusion

In this project the possibility for creating a functional player profile model was investigated. The null hypothesis of the project were on the subject of whether the created model would be able to predict which games people would like to play. The model created did not fulfill the null hypothesis. However it has added some important aspects to the research area of player types. These additions include an incentive to get other researchers to think more about the difference between how people play games and why they play games. And also about creating models that could work for both games and players. The model can be used as an ethnographic tool to map player behavior to more detailed extent than most other models currently invented.

12 Bibliography

Bartle R. A. (1996). *Hearts, Clubs, Diamonds, Spades: Players who suit MUDS.* Hentet fra http://mud.co.uk/richard/hcds.htm

Bartle R. A. (2003). *Designing Virtual Worlds.*

Bostan B. (2009). Player Motivations: A Psychological Perspective.

Briggs K. C., M. I. (1962). *Myers-Briggs Type Indicator.*

Csikszentmihalyi M., N. J. (2002). The Concept of Flow. I L. S. Snyder C. R., *Handbook of Positive Psychology.*

Dixon D. (2011). Player Types and Gamification.

Fog H. S. (2011). The Player Engagement Process - An Exploration of Continuation Desire in Digital Games.

Kilkku V. (2013). *Bartle player types, Yee's motivations, and self-determination theory.* Hentet fra http://www.kilkku.com/blog/2013/08/bartle-player-types-yees-motivations-and-self-determination-theory/

Laliberte P. (2010). *Hack & Slash RPG - A Unity 3D Game Engine Tutorial*. Hentet fra Burgzerg Arcade: http://www.burgzergarcade.com

Malone T. W., L. M. (1987). Making learning fun: A taxonomy of intrinsic motivations for learning. I F. M. Snow R.E., *Aptitude, Learning, and instruction: Conative and Affective Process Analyses.*

Marczewski A. (2013). User Types 2.0.

Papadakis E. P. (1983). Why and What for (Four): The Basis for Writing a Good Introduction.

Rueløkke J. (2014). *How You Play*. Hentet fra Google Docs: https://docs.google.com/forms/d/1-8omPXhZJkP2cFN2lBrxVSyUhj_i3dNL9d8CrIsghcM/viewform?usp=send_form

Shell J., K. C. (2006). Why People Play Games, An Industry Perspective.

Sweetser P., W. P. (2005). GameFlow: Amodel for evaluating player enjoyment in games. I *ACM Computers in Entertainment 3* (s. 1-24).

Turbosquid. (2014). Hentet fra Turbosquid: http://www.turbosquid.com

Unity Technologies. (2014). *Asset Store*. Hentet fra Unity3D: https://www.assetstore.unity3d.com/en/

Wolffelt M. (2012). *Unity Community*. Hentet fra Unity3D: http://wiki.unity3d.com/index.php?title=CSharpMessenger_Extended

Yee N. (2002). *Bartle's Types.* Hentet fra http://www.nickyee.com/facets/bartle.html

Yee N. (2008). Maps of Digital Desires: Exploring the Topography of Gender and Play in Online Games.

13 Appendix

13.1 Table Summarizing Player Types

Bartle First Model	Killer
	Socializer
	Explorer
	Achiever
Bartle Second Model	Politician
	Networker
	Friend
	Griefer
	Planner
	Scientist
	Hacker
	Opportunist
Marczewski Model	Free Spirit
	Achiever
	Socializer
	Philanthropist
	Disruptor
	Player
Yee Model	Advancement
	Mechanics
	Competition
	Socializing
	Relationship
	Teamwork
	Discovery
	Role-Playing
	Customization
	Escapism
Shell and Kurg model	Competitor
	Explorer
	Collector
	Achiever
	Joker
	Director
	Storyteller
	Performer
	Craftsman
Fog model	Solving
	Sensing
	Interfacing
	Exploration
	Experimentation
	Creation
	Destruction
	Experiencing The Story
	Experiencing The Characters
	Socializing
	Achieving
	Progressing
	Completion

13.2 Focus Group Notes

First focus group

Casual vs hardcore 1 min in
Mind vs moving
11 min: Do whatever and then story
12 min: Boring stuff then do more interesting sidestory, side get off track
13 min: Boring then sidetrack

Bartle

16 min: like the model explorer
18 min: doesn't like the model too compromised, both explorer and non explorer
19 min: likes the model
20 min: games might facilitate a certain type of game
21 min: like the model but again more affected by the

Marczewski

28 min: streamers next to being a gamer,
31 min: more open model
32 min: explore the achievement
33 min: more accurate

Nick Yee

36 min: Best model
41 min: The game will enforce how you play,
42 min: A little bit extreme model
43 min
55 min: Different for games,

Created Model

58 min: Escapist should be added it is also a how immersionist
59 min: Idleist might miss something, timewaster, casual enjoyer, roamer
61 min: Roleplayer, Character explorer, narrative explorer
66 min: Minecraft lorejunkie
68 min: Narrative explorer and character explorer
70 min: The how is determined by the why
72 min: Rate each thing
79 min: Questionnaire or not
83 min: Define from games not their own
89 min: Use the gamers to model gamemodels

Second focus group

6 min: Exploring, collecting, progression, idiot, pvper
7 min: Singleplayer even in MMOs, completionist used as more someone seeing everything,
8 min: Skipping the story
9 min: The social aspect, completionist, trophy hunter
12 min: Depending on the game, star wars quest, world of warcraft group play
13 min: Numbers in an excel ark, ego online
14 min: Helping others thereby getting one self some enjoyment
15 min: MMOs kill the important aspect by people doing the same quest
16 min: Everyone wants to be the biggest star!, Story makes you believe you are part of something, EVEs economy
18 min: Making gamers feel more special
22 min: Making the challenge harder for yourself

Bartle

25 min: One can see it completely, another says it depends on the game
26 min: Killer does mostly fit
27 min: Depends on the game
28 min: Too narrow

Marczewski

30 min: Marczewski worse than the previous model
36 min: Believe that we are everything

Yee
38 min: Likes Nick Yees model more than Bartle and Marczewski
Created Model
46 min: Lore-Junkie vs World Explorer perhaps too closer
47 min: Lore-Junkie terrible term History, Story, Gamesetting, Reflex bad term
49 min: Roleplayer, Destroyer might overlap griefer, better term?
54 min: Model close to the Mayas Briggs
58 min: Narrative explorer perhaps instead of lorejunkie,
59 min: Collector part of the
60 min: Completionist wrong word, Challenge seeker, use completionist
61 min: Group them up, vizualisation
62 min: Graphics and sound in some way, or not if its how you play alone
64 min: Sound and graphics perhaps
Previous groups notes
67 min: Collector vs. Completionist maybe the same broaden the definition Hunter or Gatherer
76 min: Way to answer the model will make it crash and burn or be relevant

Third Focus Group

1 min: Depends a lot on the game, hitman tries to be killer, strategy game strategic
2 min: How the game intends to be, optimal way to play the game, achieve
3 min: A few times just explore
4 min: How I feel like playing at the moment, the mood
5 min: To benefit me, finish the game quickly if you go for it, sidequests for exploring, story
6 min: Moral options in games, different ways new times you play
7 min: Fitting the character to the environment (Character explorer)
8 min: Usually play for the story
9 min: Feel for the fictional children
Bartle
13 min: Just level, play pve, pvp, completionist
14 min: Pet battles achievement points
18 min: Explorer
20 min: PVP killer or achiever
22 min: Identity seeking/knowledge seeking
23 min: 3 kinds person
25 min: Like the setup not the idea
27 min: Depends on the game
28 min: Aged concept, pick another game or pick another aspect of a game
32 min: Depends on the game
34 min: When playing mindcraft = explorer, when playing hitman achiever
Marczewski
35 min: Situational,
38 min: Fit more games
Yee
41 min: Nick Yee Not for every game
43 min: More flexible model than the others, cant really be used for anything
Created Model
53 min: Roleplayer, Lore-people and roleplayer perhaps together?
54 min: Assister <- Big sister
55 min: Griefer bad term
56 min: Big exstensive model, variating results
57 min: 0-7 scale
59 min: Trying to fit it into minecraft
60 min: Story progressor, narrativist
61 min: Achiever
62 min: Builder and crafter
64 min: Group up, next to each other much more,
66 min: Challenge seeker
67 min: Too big model too many points, others think its too close
68 min: Builder/crafter = creator

68 min: Categories
69 min: Mechanic instead of mechanics explorer
71 min: Challenge seeker if I can see myself completing
76 min: Kind of challenge
80 min: Differences in the model, sorted a bit more
82 min: Why would be important as well
83 min: Game added
84 min: 5 points perhaps
85 min: Only choose some point
Previous groups notes
86 min: Selling and research spiderweb model, our own purpose just points
87 min: Spiderweb or not.
89 min: Less categories
92 min: Idleist working or not?
93 min: Confusion about progressor, completionist, collector and challengeseeker
95 min: Thinking more about why than how
100 min: Too specific model

13.3 Explaining The Player Profile Model Parameters

Progressor
The progressor continues through game content in a fashionable speed. They do not like being stuck in the same place for too long and doesn't usually sidetrack too much while doing a piece of content.

Challenge Seeker
The challenge seeker want's to get their skills tested. The challenge seeker will often play games within their game genre that are known for their high difficulty or play against people that are hard to beat. You wont often see the challenge seeker stop playing because of dying.

Gatherer
The gatherer collects all the things. Whether it is achievement points, pokemon or vegetables. The gatherer wants to get them all.

Destroyer
The destroyer wants to annoy other people as much as possible and kills them, destroys their creation and generally does everything he or she can to get people angry.

Competitor
The competitor wants to play against others in as fair a setting as possible.

Socialiser
The socialiser uses games as a social platform. Whether it is to talk with people inside the game or playing the game just to be able to follow the discussion with ones friends the player does it for the social aspect.

Cooperator
The cooperator wants to work together with others to complete a common goal. Whether it is to defeat some game created challenge or some challenge these players create themselves is not necessarily important as long as its a challenge they have to work together to complete.

Assister
The assister uses a lot of time on helping others. Whether it is by telling them where to go or whether it is by actively helping them kill some monster the assister just uses time on this aspect.

Mechanics Explorer
The mechanics explorer looks for all the small extra features in the mechanics of the game. This can be as simple a thing as to find out how much a headshot hurts, compared to a shot in the leg. Or something as extensive as finding the optimal build in an MMO.

World Explorer
The world explorer sees a tall mountain far away and straight away wants to go there. Even though the game might have sent the person to some other corner of the world, the world explorer wants to see all the caves, mountains, fields and so on.

Narrative Explorer
The narrative explorer is the guy or girl who knows every background story of every monster in the game. He will read every quest text, talk with all the NPC's, and read every single book of lore in the game. Some might even go outside the game to look up background stories of characters and monsters within the game. Lesser scores within the narrative explorer would be reserved for people who like to read the story but might not go all out on finding out everything.

Role-Player
The roleplayer is the character instead of playing the character. The roleplayer might take an action that for the gameplays sake is obviously worse but for the sake of the character seems more realistic. For example if a roleplayer has chosen that their character is scared of spiders instead of killing the spiders they might stun themselves in terror.

Creator
The creator adds things all around them. Whether it is buildings or other things affecting the landscape around them, or costumes and items that can be equipped.

Strategizer
The strategizer plays with their mind. They try to figure out the optimal way on completing a challenge and might even go think about how it can be completed without even playing.

Reflexor
The reflexor challenges their cognitive abilities as much as possible. This can be their reflexes, their memory, or any other ability. They improvise alot from how the game acts on them.

Idleist
The idleist has the game running but for the most part the computer works harder than the idleist. The idleist does other things while playing and their focus might not be all on the game. Games that can be said to fulfill this role are facebook games like Farmville or Mafia Wars but also big games like world of warcraft have idleist elements like fishing.