



Recruiting as a Game

Exploring Gameplay Design Patterns in the Development of a Recruiting Process

Master's thesis in Interaction Design and Technologies

SIMON LINDKVIST
JESPER OLSSON

MASTER'S THESIS 2018

Recruiting as a Game

Exploring Gameplay Design Patterns in the Development of a
Recruiting Process

SIMON LINDKVIST
JESPER OLSSON



Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2018

Recruiting as a Game
Exploring Gameplay Design Patterns in the Development of a Recruiting Process
SIMON LINDKVIST
JESPER OLSSON

© SIMON LINDKVIST, 2018.
© JESPER OLSSON, 2018.

Supervisor: Josef Wideström, Department of Computer Science and Engineering
Advisor: Future Skill AB
Examiner: Staffan Björk, Department of Computer Science and Engineering

Master's Thesis 2018
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Three screens from the recruitment tool developed as a part of this thesis.
Read more in Sections 5.3 and 5.4

Typeset in L^AT_EX
Gothenburg, Sweden 2018

Abstract

Recent years have seen, and future years will see an increased demand of talented programmers. This implies that companies at some point will need to reform their recruiting processes to be able to attract these talents and compete with rivaling companies.

This master thesis investigates how to utilize gameplay design patterns (GDP) as a design foundation to reform a general recruiting process, with the purpose of analyzing the impact of applying gameplay design patterns on a general recruiting process. To achieve this, a rigorous analysis of existing gameplay design patterns in a recruiting process has been made, as well as an investigation of what impact the addition of other gameplay design patterns to the process has. As a result of this, a concept design of a recruitment tool called *Operation Talent* was developed. Operation Talent was developed in order to illustrate an example of what a recruiting process could look like if it was designed as a game with a foundation of gameplay design patterns. From the standpoint of this concept design, arguments regarding the impact of applying gameplay design patterns on a recruiting process are made.

Furthermore, lessons learned and knowledge gained from working with gameplay design patterns as a design foundation and as an analysis tool are discussed. Future research within this subject could imply further developing the Operation Talent tool with other gameplay design patterns, evaluating the tool with real users or investigating the application of gameplay design patterns on other routine processes.

Keywords: Gameplay Design Patterns, Recruiting, Gamification

Acknowledgements

We would like to thank our supervisor Josef Wideström for guiding us through the jungle of writing a master's thesis through his clever and spot-on metaphors.

We would also like to thank Henrik Enström and his colleagues over at Future Skill and Software Skills for providing us with a very nice place to work, and continuous feedback on our progress. Not only is their office a great place to work, it is also a perfect place for mastering the art of ping pong, which, in addition to this thesis, is what we have done.

Contents

1	Introduction	1
1.1	Research Problem	1
1.2	Hypothesis	2
1.3	Delimitations	2
1.4	Intended Result	2
1.5	Outlining the Thesis	3
2	Background	4
2.1	Stakeholders	4
3	Theory	6
3.1	Games	6
3.1.1	Definitions of games	6
3.1.2	Gameplay	7
3.1.3	Gameplay design patterns	7
3.1.4	Mechanics, Dynamics, Aesthetics	8
3.2	Playing Games	8
3.3	Gamification	9
3.4	User Engagement	9
3.5	Viral Sharing	12
4	Methodology	13
4.1	Frameworks	13
4.1.1	Research through design	13
4.1.2	Gameplay Design Patterns	13
4.1.3	Playcentric design	14
4.1.4	Iterative prototyping	14
4.2	Methods	15
4.2.1	Inspiration	15
4.2.2	Ideation	16
4.2.3	Implementation	17
4.2.4	Evaluation	18
5	Design Process	19
5.1	Iteration 1: Designing a Recruitment Platform For Increased User Engagement	19
5.1.1	Inspiration	19

5.1.2	Ideation	20
5.2	Change of Focus	25
5.3	Iteration 2: Analyzing and applying gameplay design patterns on a recruiting process	25
5.3.1	Defining a Recruiting Process	25
5.3.2	Analyzing Gameplay Design Patterns in a Recruiting Process	26
5.3.3	Adding Additional Gameplay Design Patterns	31
5.3.4	Defining The Core Gameplay	33
5.3.5	Designing a Game Concept Based On Gameplay Design Patterns	34
5.3.5.1	Concept Creation	34
5.3.5.2	Lo-Fi Wireframes	36
5.3.5.3	Me-Fi Mockups	37
5.3.5.4	Hi-Fi Mockups	41
5.4	Iteration 3: Adding a theme to the design	50
5.4.1	Ideation for themes	50
5.4.2	Implementing a theme	52
5.4.2.1	Lo-Fi Sketching	52
5.4.2.2	Hi-Fi Mockups	55
6	Results & Findings	67
6.1	Impact on the Recruiting Process	68
6.2	Role of the Tool in the Recruiting Process	70
7	Discussion	71
7.1	Reflection on Results	71
7.2	Reflection on Process	73
7.2.1	Gameplay Design Patterns as an Analysis Tool	74
7.2.2	Gameplay Design Patterns as a Design Foundation	74
7.3	Further Work	75
7.4	Ethical Considerations	76
8	Conclusion	78
	Bibliography	79
A	Results of Brainstorming Features For Recruiting Platform	I
B	Programming Challenges Questionnaire	III
C	Gameplay Design Patterns in a Recruiting Process	VIII

1

Introduction

In this era of digitization, previously analog routine processes such as talent sourcing, income tax declaration and banking matters are moving into a digital space. This transformation entails an increased importance of research fields concerning interaction design, human-computer interaction and programming. How people perceive experiences of said processes lies in the hands of interaction designers, programmers and other specialists within the fields.

Furthermore, recent years have seen an increase in proliferation of software taking inspiration from games, commonly referred to as gamification. Gamification takes well-recognized concepts from human-computer interaction and combines them with concepts from game studies in an attempt to increase user engagement within the system [13].

Hence, not only are the previously analog processes being digitized, they are also being gamified. The fact that gamification is a relatively new phenomenon brings certain challenges when attempting to apply it to everyday processes. For example, the sample size of how gamification affects users' perception of routine processes is still small and no generalized guidelines or established theories exist regarding how to make proper use of it.

This master thesis intends to focus on one of the aforementioned routine processes, namely talent sourcing. A study conducted by Microsoft predicts that between 2010 and 2020, at least 1.2 million job openings in computing professions requiring at least a bachelor's degree will go unfilled [35]. But is this a question of a shortage of talented programmers, or ineptitude in reaching out to them? Assuming there's a challenge in finding talented programmers, it is of interest to investigate how interaction design in general and game design in particular can support the process of recruiting talented programmers. One could assume that different methods and approaches to recruiting and talent sourcing attracts people with different skills and mindsets. Using a gamified approach to talent sourcing will naturally attract people with an interest in games. With the trend of routine processes being gamified, the importance of attracting talented programmers with an interest in and understanding of games heightens.

1.1 Research Problem

The aim of this master's thesis is to investigate in what ways game design in general and gameplay design patterns in particular can affect the process of recruiting talented programmers. More specifically, this thesis tries to answer the following

question:

What impact do the application of gameplay design patterns have in the development of recruiting process?

by addressing the following subquestions:

- What gameplay design patterns exist in a general recruiting process?
- What additional gameplay design patterns would have an impact on users' engagement?

1.2 Hypothesis

This thesis presupposes that recruiting talented programmers with an interest in games is positive for the field of information technology and society as a whole, and that creating game-like recruiting methods will help in attracting such programmers. Not only can programmers with an interest in games help in swaying the trend of gamification in the right direction, but recruiting with game-like methods can help bring out the competitive fire that many potential candidates possess. If someone gets a high score on a game it usually indicates a very high skill level or persistent efforts, or a combination of both. This can be argued to be two of the most important qualities to look for in a candidate when hiring personnel for any type of job. The quality of giving persistent efforts can often be hard to manifest in normal recruiting processes.

Within the scope of this thesis, it is reasonable to believe that utilizing gameplay design patterns as a design framework to further develop the recruiting process will have an impact in attracting candidates with the aforementioned mindset, and facilitate them in showing the qualities of having a high skill level and giving persistent efforts. Furthermore, it is reasonable to believe that the use of gameplay design patterns will have a positive impact in regards to the engagement and persistency of these candidates.

1.3 Delimitations

This thesis investigates what affect the application of gameplay design patterns have in the development of a recruiting process. Some aspects of this investigation involves user engagement. However, this thesis does not provide user evaluations on this aspect, due to the fact that engagement is generally measured over a long period of time, which is not possible within the scope of this thesis. Instead, arguments as to how the application of gameplay design patterns can affect the engagement of users are made.

1.4 Intended Result

The result of this thesis is three-part result. The first part is an analysis of a general recruiting process using gameplay design patterns. The second part is a concept design of what a recruitment process could look like if these gameplay

design patterns were used in the development of a recruiting game. The third and last part of the result is the knowledge gained about the impact of working with and applying gameplay design patterns outside the context of games.

1.5 Outlining the Thesis

This section outlines the structure of the thesis, and briefly explains the contents of each chapter.

In the *Introduction* (Chapter 1), the overarching theme of gamifying routine processes is put into the context of interaction design, and the research problem is presented.

Furthermore, *Background* (Chapter 2), goes into more detail on the current state of recruiting, as well as what stakeholders exist for this thesis.

In the chapter *Theory* (Chapter 3), theoretical background regarding games and user engagement is presented, and these concepts and frameworks are later used to connect the research in this thesis with known theory.

Following the chapter on theory is the chapter on *Methodology* (Chapter 4), where broader methodological frameworks are brought up, as well specific methods that could facilitate achieving the goals of the thesis.

The next chapter, *Design Process* (Chapter 5), explains how the methods in the previous chapter were utilized and what results those methods yielded. Everything from the start of the work to the final outcome is presented in this chapter through three iterations with different goals and design decisions. Hence, parts of the result and partial answers to subquestions of the research question reside there.

The chapter *Results & Findings* (Chapter 6) tries to extrapolate the results yielded by the design process, and draw conclusions regarding the impact of gameplay design patterns on a general recruiting process by revisiting some of the concepts presented in the theory chapter.

Subsequent to presenting the findings, a *Discussion* (Chapter 7) of them is provided. Here, some interesting aspects of the findings are discussed, as well as a reflection of the process and thoughts on working with gameplay design patterns, opportunities for further work and discussing ethical aspects of the work done in this thesis.

To wrap the thesis up, the chapter *Conclusion* (Chapter 8) briefly summarizes important aspects of the preceding chapters.

2

Background

The field of recruiting is an evolving domain. Several companies are adopting a more digital strategy towards talent sourcing in order to attract interest from a wider range of candidates and also decrease the workload of the people working within recruiting. Today, there exist quite a few examples of companies providing digital platforms containing challenges and problems with the aim of sourcing talents within the programming field. Examples of such platforms are *Kattis* [27], *HackerRank* [22] and *Codility* [4]. What these three platforms have in common is that they make use of mathematical problems as a basis for their challenges, with little to no graphics or game elements. One could say that these companies are merely digitizing the otherwise analogue competence test found in many recruiting processes. While this is a proven and standard way of sourcing talent within the field of programming it misses out on the creativity and engagement that games can bring out of their applying candidates, by not utilizing the gamification possibilities that digitization brings.

Albeit not fully utilizing gamification possibilities, the aforementioned platforms do impact the recruiting process in some aspects. For example, they provide candidates with the privilege of taking competence tests outside the, for many, stressful context of a work place or interview room. They also, to a certain extent, provide recruiters with automated evaluation of a candidate's competence, enabling shorter feedback cycles between stages in the recruiting process.

2.1 Stakeholders

The main stakeholder, apart from Chalmers University of Technology, is a company called *Future Skill* [44]. Future Skill develops and sells an online recruiting platform focusing on sourcing talent within the fields of information technology and programming. Their platform consists mainly of two sections. One section is focused solely on standard coding tests, much like Kattis, HackerRank and Codility. The other section however, is focused on gamified programming challenges, where potential candidates compete against each other by creating algorithms to solve different kinds of gamified problems. The algorithms that the candidates write are directly translated into actions in a game, meaning that the code acts as a controller for a game. This section of the platform is however not heavily used for recruitment purposes as of today. It is used more to promote their platform to potential candidates, rather than as a tool for the recruitment process. Furthermore, this part of the platform is still young, and leaves room for further exploration and development. The

company Future Skill has an interest in this project because it has the potential to further gamify their recruiting process, as well as acting as an experimental project with little to no resources or risks involved.

The end users, meaning the companies who buy and the recruitees who use the platform are secondary stakeholders, since the research performed in this project has the potential to lead to a better experience for them. For the companies, this project might be a catalyst for increased engagement amongst recruitees, which in turn can lead to a better overall recruitment process. For the candidates, the insights gained in this project can lead to them getting further motivation to perform better during the recruiting process, which in turn can lead to them getting more job opportunities.

3

Theory

In this section, some, to the thesis, relevant theoretical frameworks and concepts are presented.

3.1 Games

Throughout the history of games, formal approaches to talking about games, developing them and designing them have been created. In this section, some relevant definitions of games are presented, as well as an outline of what gameplay is and some frameworks for analyzing games.

3.1.1 Definitions of games

One of the very first attempts at defining a game was made by Huizinga in 1938 [23]. He defined what he called play as being

"a free activity standing quite consciously outside 'ordinary' life as being 'not serious', but at the same time absorbing the player intensely and utterly. It is an activity connected with no material interest, and no profit can be gained by it. It proceeds within its own proper boundaries of time and space according to fixed rules and in an orderly manner. It promotes the formation of social groupings, which tend to surround themselves with secrecy and to stress their difference from the common world by disguise or other means".

Another definition was made by Bernard Suits [46], in which he stated that playing a game is a voluntary effort to overcome unnecessary obstacles. Furthermore, Salen and Zimmerman [43] said in their definition that

"a game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome".

These definitions are merely a selection from the abundance of existing game definitions and stand to show some of the diversity within the notions people have of games.

It is difficult to apply Huizinga's definition of play on gamified programming challenges for recruitment purposes, due to the fact that he states that no profit stands to be gained from playing, and that the programming challenges are means

of finding a job, which indirectly can lead to a profit. Salen and Zimmerman however, mentions nothing about profit in their definition, and all parts of their definition can be correlated to a gamified programming challenge. The *system* in this case can be thought of as the computer hosting the programming challenge. The *artificial conflict* is the question of who can provide the best solution to the problem given by the challenge, and the solutions are interpreted based on the *rules* given by the system. The *outcome* is *quantified* with a score based on how good the provided solutions are.

3.1.2 Gameplay

Almost every person attached to games, whether it be people playing games or people designing games talk about gameplay. However, when Fabricatore [16] investigated the theoretical landscape of games, he was unable to find a unified and formal definition of gameplay. He found out however, that players have a well-defined notion of what gameplay consists of. They refer to gameplay as

- what players can do in a game
- what other entities can do in response to a player's action (i.e. how the game responds to player's decisions).

Furthermore, Fabricatore claims that gameplay is the most important pillar of the game design activity and that a set of good core game mechanics and thought-through satellite game mechanics allow creating games with a simple, yet rich and entailing gameplay. Game mechanics describe how players can interact with the virtual world. Fabricatore defines satellite game mechanics as enhancements to already existing core game mechanics. An example of a satellite mechanic could be adding a telescopic sight to an assault rifle in a shooting game.

3.1.3 Gameplay design patterns

The introduction of game design patterns by Björk, Lundgren and Holopainen [2] was an attempt at giving game designers, game researchers and other people working in the game industry a unified vocabulary for talking about and understanding games. A pattern in their framework is described with a *name*, a *description*, a set of *consequences* for using the pattern, guidelines on *using the pattern* and what *relations* the pattern has to other game design patterns. The name should be short, descriptive and specific. The description should contain information on how the pattern affects the structural framework, while still being concise and also containing some examples of games in which the pattern has been identified. Consequences should describe the costs, benefits and trade-offs of using a specific pattern. The section on using the pattern should describe the common choices a designer is faced with when applying a pattern, often accompanied with examples of game elements from published games. The relations between patterns can come in three different shapes: superior, subordinate and conflicting. An updated collection of gameplay design patterns, compiled as a result of their research, can be found on [1].

3.1.4 Mechanics, Dynamics, Aesthetics

Hunicke et al. proposed the MDA framework [24], short for Mechanics, Dynamics and Aesthetics, as a formal attempt at understanding games, trying to bridge the gap between game design and development, game criticism and technical game research. This framework formalizes the consumption of games in three distinct components: rules, system and "fun". Mechanics, dynamics and aesthetics are the design counterpart of those components, meaning that the game designer controls what mechanics, dynamics and aesthetics exist in a game, while the consumer (player) experiences mechanics as rules, dynamics as the system and aesthetics as "fun".

Mechanics describe the particular components of a game, at the level of data representation and algorithms. Dynamics describe the run-time behaviour of the mechanics acting on player inputs. Aesthetics describe the desirable emotional responses evoked in the players when they interact with the system. Hunicke et al. use the taxonomy illustrated in Table 3.1 to describe the eight aesthetics of a game.

Name	Description
• Sensation	• Game as sense-pleasure
• Fantasy	• Game as make-believe
• Narrative	• Game as drama
• Challenge	• Game as obstacle course
• Fellowship	• Game as social framework
• Discovery	• Game as uncharted territory
• Expression	• Game as self-discovery
• Submission	• Game as pastime

Table 3.1: Taxonomy of aesthetics in games, as proposed by Hunicke et al.

3.2 Playing Games

People have been engaging in game-like activities of different shapes and forms for a long time, spanning from 5000-year-old board games [34] to modern computer games, but to what purpose? Why do people appreciate the idea of devoting their time to engaging in artificial conflicts? One heavily researched aspect of games is the aspect of having fun. Koster talks about this in his book about the theory of fun [30], mentioning that humans experience fun when learning and mastering something and that games provide opportunities for achieving this. Games offer a learning possibility in a context where there is no pressure from potential consequences. Furthermore, Koster talks about factors and aspects of games that can evoke boredom. Some examples would be that the player figures out how the game works too early on, that the game reveals patterns too quickly or too slowly or that the player masters the game entirely.

3.3 Gamification

Research on the subject of gamification exists to a certain extent. For example, Kiryakova et. al [29] talks about how to include gamification in the context of education in their paper and concludes that gamification elements can have positive effects on users' motivation, attitude and engagement. Furthermore, Liu et. al [33] analyzes gamification in intelligent environments and concludes that the main functionality of the service or product is still the most important aspect, but that gamification can help in increasing user engagement.

Both these papers talk about general gamification elements, such as points, levels and badges, but none of them mentions anything about a structural framework from which these elements are taken.

Furthermore, not everyone agrees that gamification is a positive trend. Professor and video game designer Ian Bogost [3] wrote an article titled "Gamification is Bullshit", in which he outlines ways in which the current trend of gamification gets games wrong. He critiques the whole idea of gamification for "mistaking incidental properties like points and levels for primary features like interactions with behavioural complexity" and states that businesses only use gamification to keep up with competitors and make easy sales.

3.4 User Engagement

Alistair Sutcliffe proposed "user engagement" (UE) as a term to describe users qualitative judgments of an interactive product [47]. This is generally a complex and context-based process, but Sutcliffe saw in his studies signs of aesthetics, metaphors and interaction having higher importance in games and other entertainment media in order to maintain the UE, as opposed to the more serious field of applications where utility and usability seemed to be of higher importance. When speaking about the field of games, it is also important to adjust the difficulty level, especially in games where a player can become better at an action by practising it. In cases where the difficulty level is not adjusted, the user will generally get bored and the UE will drop [48]. Keeping users engaged by finding a balance between the challenge and the user's skill often results in something called *flow*.

Flow is today a relatively common term, but was first coined in 1975 by Mihaly Csikszentmihalyi [10]. There are three important factors contributing to achieving flow [12]:

- Clear goals
- Balance between challenges and skills
- Instant feedback

An illustration of the balance between challenges and skills can be seen in Figure 3.1, where eight mental states have been divided into a challenge-to-skill ratio diagram. The flow state is only reached by having a certain amount of challenge, while simultaneously possessing a certain skill level. Having a high level of challenge and a low level of skill will result in a state of *Anxiety*. Csikszentmihalyi also argues that it is not possible to reach a flow state by for example watching TV, since skills are

not being utilized in an attempt to solve a challenge [11]. Instead, he calls this state *Apathy*.

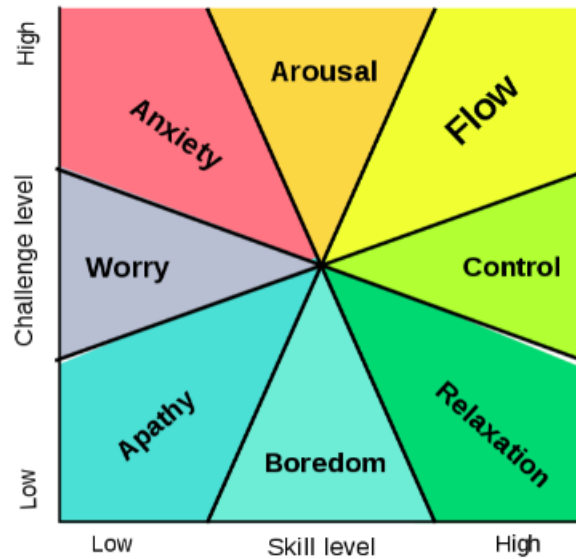


Figure 3.1: A redesign of Csikszentmihalyi’s flow model of a person’s mental states.

Furthermore, one could describe the set of existing emotions as either positive or negative. James A. Russel did in one of his research projects a *model of affect* where he placed 28 *affect words* or verbs in a coordinate system consisting of the x-axis “misery-to-pleasure” and the y-axis “sleepiness-to-arousal” [42]. Some examples of the negative emotions would be those on the negative x-axis side, thus on the misery side of the scale (see the right column of Table 3.2 for an example set of negative emotions). On the positive side of the x-axis, emotions striving towards pleasure are found (see the left column of Table 3.2 for an example set of positive emotions).

Positive	Negative
• Aroused	• Angry
• Excited	• Afraid
• Happy	• Sad
• Satisfied	• Depressed
• Relaxed	• Bored
• Calm	• Tired

Table 3.2: Example of positive and negative emotions (derived from Russel’s model of affect)

Another representation of emotions is Robert Plutchik’s work, where he created an illustration of what he came up with was the eight core emotions of humans [40]. See Figure 3.2 for an illustration of these eight emotions. The illustrated model has different levels of intensity; the distance to the center of the model describes

the strength of the emotion, with closer to the center meaning stronger. Just like in Russel’s model, it is also possible to divide some of these emotions into the two categories positive and negative emotions.

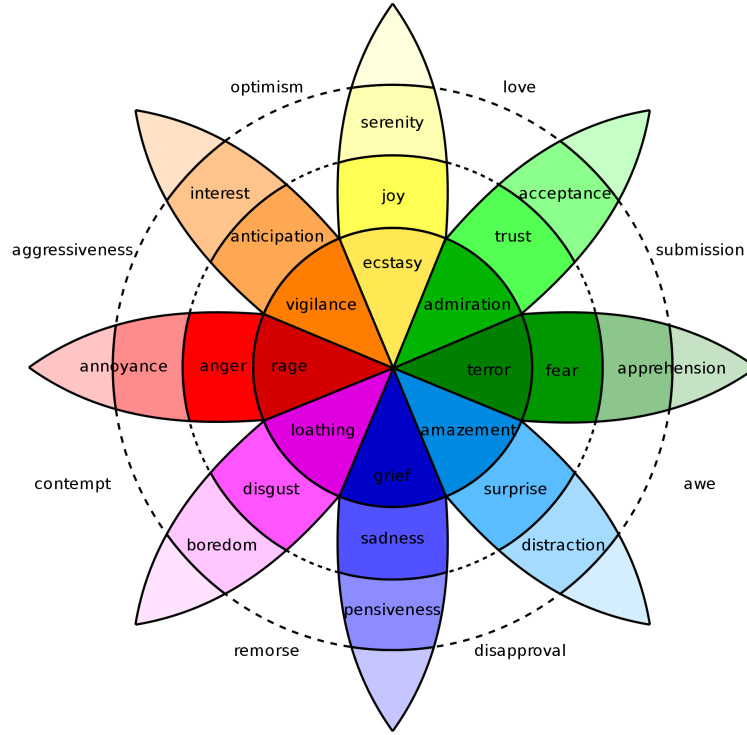


Figure 3.2: Plutchik’s wheel of emotions.

Why is positive versus negative emotions of interest? In Nico H. Frijda’s laws of emotions, emotions are seen as responses to meaningful structures of situations [19]. The emotional responses are complex and rely on many variables in the situation. Hence, it is important to understand that it is not always best to design solely towards the positive list of emotions, because in some contexts, the negative emotions also lead to a positive experience in the end.

The emotion frustration is a frequently used example, especially in games. A lot of games challenge the player by having some kind of difficult action, an example being the mobile game Flappy Bird [36]. The task in Flappy Bird is quite simple. The player is supposed to fly through the holes between pipes. The game’s challenge lies in that the steering is difficult, which after several crashes and attempts at beating the high score, often results in a frustrating feeling for the player. The point here is that when the player finally achieves a new high score, the reward will most likely be greater than without the challenge and frustration. The reward after being frustrated could come in form of a positive emotion - satisfaction.

This way of using one or multiple negative emotions to achieve a stronger positive emotion creates a so called rich experience. Fokkinga and Desmet [18] have developed a three step process for designing such rich experiences, consisting of:

- Emotion selection - Choose an appropriate emotion for the situation
- Emotion elicitation - Find ways to evoke the chosen emotion

- Emotion reversal - Make the negative emotion reversible and transformed into a positive emotion

3.5 Viral Sharing

The phenomenon occurring when material is being massively self-spread through social media could be described as *viral sharing* [5]. Elizabeth L. Cohen conducted a study about what makes games go viral. The study attempted to answer the question about viral sharing by identifying player characteristics and experiences when playing the game *Darfur is Dying*. One of the main findings of the study was that positive emotions increase the odds of players sharing the game during their early phase of playing the game, while negative emotions did, on the other hand, increase the odds of players sharing the game in the week after their initial phase of playing the game [6]. Worth to notice is that the study only involved one game and a specific target group - traditional college students [7] - in the analysis and it is therefore uncertain if the findings are only applicable to that context.

4

Methodology

In any project, using suitable frameworks and methods as a basis for the work process is a key to success. Working with design as a method for research is challenging, and integrating game design methods with methods for interaction design comes with challenges of its own.

4.1 Frameworks

This section presents the overarching methodologies, frameworks and concepts that the work in this thesis has been based around. Section 4.2 describes the concrete methods.

4.1.1 Research through design

In 2007, Zimmerman et al. [49] presented their model of *Research through design*, outlining how interaction designers can contribute to human-computer interaction (HCI) research by facilitating different disciplines. This model allows the HCI research community to engage with wicked problems [41], as well as allowing interaction designers to make contributions to research by utilizing the skills they possess. Research through design is not a design methodology in and of itself, but acts more as a statement articulating that research and design are intertwined disciplines that stand to gain valuable information from each other.

In his article about research through design, Gaver [21] talks about design as a research practice being generative; it is not describing the world as it is, but rather investigating how to create new ones. Gaver continues, stating that design is underspecified by theory in the sense that many aspects of a successful design will not be captured by given theories. What Gaver concludes from his investigation of the research through design concept is that practitioners should have moderate expectations of creating verifiable theory when pursuing research through design. Gaver suggests, rather than focusing on accepting or rejecting research through design as a valid science, reflect on how to pursue the research on its own terms and for what it is.

4.1.2 Gameplay Design Patterns

Gameplay design patterns, introduced in Section 3.1.3, is generally a framework used for analyzing and discussing games and elements of gameplay. However, little has

been published about using gameplay design patterns as a foundation of a design process. In this thesis, gameplay design patterns as a framework acts as the basis for the entire design. It provides an enclosed, but yet powerful and dynamic design space. By having a fixed amount of patterns to utilize, the framework puts an explicit limit on what can be created with it, but as a result of the abstract nature of the patterns, the interpretations and implementations, as well as the combinations of patterns can vary to a high degree.

Gameplay design patterns as a design method can be implemented to communicate desired events, behaviour and experiences of a design. In combination with the MDA framework (introduced in Section 3.1.4), gameplay design patterns provide the designer with an opportunity to both categorize and express elements of a design.

4.1.3 Playcentric design

The playcentric design approach, as proposed by Fullerton [20], is a design process specifically applicable to designing games. The approach consists mainly of four larger phases: conceptualization, prototyping, digital prototyping and playtesting, which are all performed iteratively. One of the focal points of playcentric design is involving players throughout the whole process, from conceptualization through completion. Fullerton proposes achieving this involvement by establishing *player experience goals* early and constantly checking their existence in the game being developed. Player experience goals are what they sound like: goals set by the game designer describing the intended experiences a player should have while playing the game.

4.1.4 Iterative prototyping

Iterative processes, as opposed to waterfall processes, are becoming the standard way of working in creative fields. There are an abundance of iterative processes for design purposes, and incorporating one into the design process is key to continuously be updated on the current state of the project as well as what potential users think of the project. IDEO proposes an iterative design process with the phases inspiration, ideation and implementation [25]. The inspiration phase involves research activities such as interviews, ethnographic studies and other approaches toward understanding existing solutions or attempted solutions to a certain problem. The findings and learnings from the inspiration phase is consequently fed into the ideation phase. During the ideation phase, several brainstorm methods are undertaken in order to generate a collection of ideas, which after being narrowed down are fed into the implementation phase, where a subset of the ideas take the form of prototypes. These prototypes are then tested and evaluated, after which the process is repeated several times in order to refine the results.

4.2 Methods

4.2.1 Inspiration

One approach to getting inspiration at the start of a project is to take a look at existing solutions to approximately the same problem and evaluate them. In the case of a project being a further development of an existing solution, evaluating the current solution is of high importance. This can be achieved using various methods, depending on the context.

Interviewing is a tool to get qualitative information for research purposes. Qualitative information is about an individual user's perspective and feelings, where designers can get more details out of deeper questions and therefore get very descriptive data. In-depth interviews are better at giving details about a specific group rather than giving generalized facts [32].

Baxter et al. [26] make a distinction between unstructured, structured and semi-structured interviews. Semi-structured means that the interview is structured in such a way that the interviewer has prepared questions, but is also flexible in the way that it is more open-ended, allowing the interviewer to alter the path of the interview.

One drawback of interviews is that they are time-consuming, both in the data gathering phase and in the data analysis phase. Consequently, interviews will indirectly result in smaller sample sizes in comparison with less time-consuming methods. Another challenge with using open-ended questions is that they are harder to compare, since they will probably generate a wide range of answers.

While interviews gather qualitative data, there are also methods that gather quantitative data. One of those methods is questionnaires. The use of questionnaires often aims to get large sample sizes in order to facilitate the data being transformed into numbers, which is oftentimes more easily analyzed. Albeit the method's possibly larger sample size, it does not mean that questionnaires provide more information, since they have a limitation in its in-depth data.

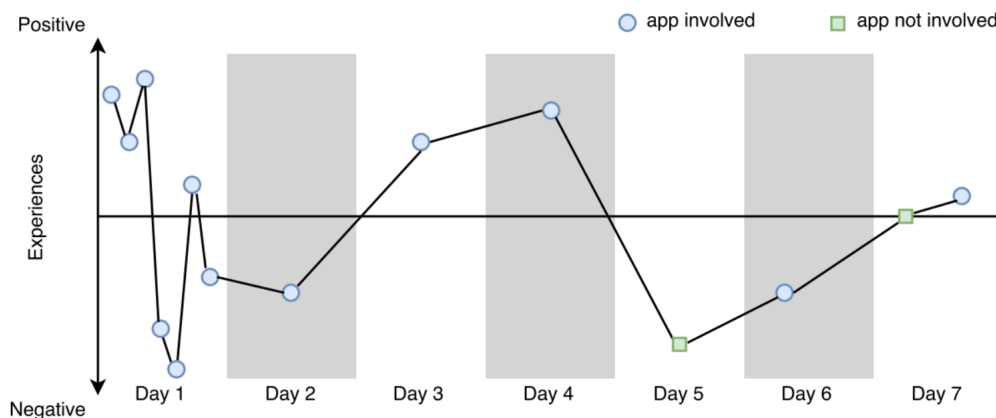


Figure 4.1: Example of a UX curve over an interval of one week.

Creating user experience (UX) curves is a method for collecting data about long-term user experience with a product or service [31]. The method is performed by

giving the test subjects a time line with an additional y-axis, which has a scale of positive to negative experience, see Figure 4.1. The test subjects are then supposed to fill in their experiences over time in this diagram in the form of data points, which have a positive to negative value and also an approximate value of the time of day in which the test subject had the experience. To get more qualitative data, the test subjects are also expected to add notes for each data point, commenting on the specific context and nuances of the experience.

4.2.2 Ideation

A substantial amount of ideation methods exist for the design team to conduct after obtaining domain knowledge from the inspiration phase. Arguably the most common method of ideation is some form of brainstorming. Some methods of brainstorming are more tailored towards generating completely new ideas, while others are tailored towards fine tuning existing ideas or selecting the best idea from a set of ideas obtained from an earlier phase of brainstorming.

One method of selective brainstorming, introduced by Faste et al. [17], is called cheatstorming. The cheatstorming paradigm proposes that *"no new ideas are necessary for further ideation to occur"*. Given a set of random ideas from previous brainstorm sessions, the cheatstorming concept bypasses the generation phase of traditional brainstorming and jumps straight to voting on which ideas to keep for the next stage.

Personas is a method for user-focused design. Creating a persona means creating a fictional person based on data that have been gathered from and about the real target group of users [37]. This means that the personas could be persons that actually exist, but are in fact merged characteristics of persons representing the different types of users. Incorporating personas then means that the design work can be tailored towards said personas and thus also towards the real users.

Furthermore, when having personas as a tool to elicit user requirements and needs, it is possible to put the personas into scenarios. Scenario-based product development with personas involved can merge the ideas of designers with the needs of users to create a valuable product [38].

Alan Cooper claims that personas help with avoiding the mistake of designing a product for everyone [8]. Cooper states that designing to satisfy everyone's needs will lead to a product no one wants, because it will consist of a clutter of every imaginable feature. These personas and their scenarios also work as a way of communicating design choices efficiently between designers, developers and stakeholders [8] [39].

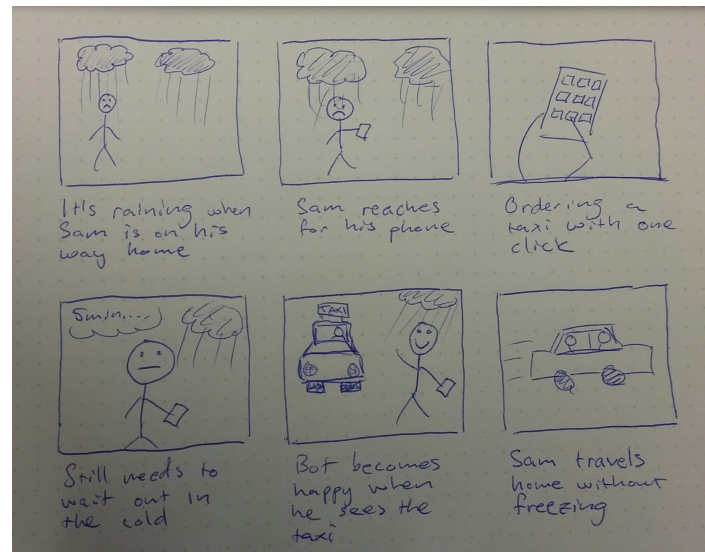


Figure 4.2: Example of a storyboard.

Another method of expressing scenarios is storyboarding. See Figure 4.2 for an example of a storyboard. It is important to remember that it is not the look of a perfect drawing that is valuable, but rather how well the scenario is expressed. Putting too much details into the early storyboards could just be a waste of time and money [8].

4.2.3 Implementation

One common mistake when designing new products is jumping to the final idea and implementation too early in the process. It is a mistake because it will result in not being able to discover the valuable, early-on findings that instead could have been discovered by applying prototyping early on in the process. It has been proven that prototyping in game design can help designers with finding out if the gameplay and its mechanics delivers the desired user experience before fully implementing the game [20].

Low-fidelity (Lo-Fi) prototypes are rough prototypes that by no means need to look perfect. The focus when creating low-fidelity prototypes for games lies in the interaction, game mechanics and how the game works as a whole. It could also be a good idea to prototype smaller parts of the whole gameplay to get an even more time-efficient prototyping process, since it is faster to do small changes directly while prototyping, rather than changing something after it has been fully implemented [20].

Paper prototyping is included in the area of low-fidelity prototyping. Testing the game design on paper instead of directly programming it can help the designer with not getting too attached to the solution at an early stage. Paper prototypes are fast to make and are also easy to throw away if needed, while programming something requires more time and effort, which therefore makes it harder to realize that it is best to start over [20].

It can be hard to understand how paper prototypes are supposed to be used when prototyping real-time games, since it is impossible to mimic the real-time

movements in a digital game using paper prototypes. On the other hand, Fullerton argues that even if the prototype may not capture the full user experience, it will still give feedback on game mechanics on a higher level such as strategic planning, weapon balancing and territorial controlling [20].

When developing a turn-based game, paper prototyping makes even more sense and can in turn generate even more valuable insights. The turn-based mechanism makes it easy to physically prototype, since the prototype basically becomes an ordinary board game.

Between low-fidelity and high-fidelity (Hi-Fi) prototypes, medium-fidelity (Me-Fi) prototypes exist. According to Engelberg and Seffah [14], Me-Fi prototypes should be fairly detailed and complete, and are best utilized to test and evaluate most interactive aspects. Me-Fi prototypes provide the advantages being sufficient for usability testing without having to work out every single detail of the interface. On the other hand, Me-Fi prototypes are in most cases not communicating the look and feel of the final product.

If designing for a digital platform, at some point in the process, the design has to move into a digital space. Eric Todd (cited in [20]) divides the digital prototyping of a game into four categories: game mechanics, aesthetics, kinesthetics and technology. Fullerton [20] argues that an important aspect to consider when prototyping game mechanics digitally is to not include all mechanics at once, but rather test mechanics separately, at least in the beginning. Aesthetics are the visual and dramatic elements of a game, and Fullerton continuously argues that they should not be worried about at a prototyping stage. However, this rule can be broken when moving in to digital prototyping, because adding some visual and sound elements can help expressing game mechanics better. The kinesthetics of a game can be explained as how the game feels and moves, and as opposed to the mechanics and aesthetics, must be tested in a digital medium.

4.2.4 Evaluation

When conducting evaluations during and after the design process, the methods used in the inspiration phase (presented in Section 4.2.1) are suitable to apply, especially when redesigning an existing solution. Using approximately the same methods that were used to evaluate the existing solution will lead to comparable results when looking to evaluate the changes made in the redesign.

In order to both evaluate results during the design process and after the design process has been finished, activities have to be performed to ensure that valuable feedback on the design can be collected. One such activity, in accordance with the playcentric design framework, is playtesting. Fullerton claims that playtesting is the single most important activity a game designer engages in [20]. Fullerton continues, stating that many designers fail to grasp that there is more to playtesting than just sitting down and playing the game and gathering feedback. Playtesting is something that the designers should incorporate throughout the whole process of developing a game, to gain insights into whether or not the player experience goals (introduced in Section 4.1.3) are being achieved by the game.

5

Design Process

The design process of this thesis is split into three different iterations with different focuses. The first iteration is a little bit of an outlier, since the main focus of the thesis was changed subsequent to performing this iteration. The first iteration proceeds from the focus question "What are the general guidelines to follow for creating an engaging user experience in programming challenges from a gameplay perspective?", while the following two proceed from the focus question given in Section 1.1. Overall, the second iteration focuses on eliciting gameplay design patterns from the recruitment process described in Section 5.3.1 in combination with other gameplay design patterns that could be interesting to explore in a recruitment context, as well as a concept developed based on these gameplay design patterns. The third iteration focuses mainly on finding and implementing a theme for the concept developed in the second iteration.

5.1 Iteration 1: Designing a Recruitment Platform For Increased User Engagement

The first iteration proceeds from a user standpoint, with some user research and mapping involved. The overarching goal of this iteration was to get to know the potential users, create personas based on the user research, as well as coming up with ideas as to what functionality these personas would want in a recruitment platform for programmers.

5.1.1 Inspiration

The first step in the inspiration phase was to scan the landscape of programming challenges, trying to understand how different types of programming challenges were perceived by programmers. This was done through a questionnaire, which can be seen in detail in Appendix B. The questionnaire was distributed through various programming-related social media channels and received 50 responses. The focal point of the questionnaire was to find out to what reason people participated in programming challenges, what they found interesting about them, and what they did not like about them. The results of the questionnaire showed that a majority of the respondents participated in programming challenges for entertainment purposes, to learn new skills and to improve existing skills. Furthermore, the three most important factors for the respondents regarding their continuous engagement in

programming challenges were that the problem is challenging, that the difficulty adjusts as they become better and that it is easy to get started.

A second step in the process was to gauge the landscape of the other side of the spectrum: the recruiters. What do they think about online recruiting methods and what do they look for in specific candidates? Can these qualities be demonstrated through online programming tests? The interview was semi-structured with these topics as a basis, but the interviewee was free to treat other aspects of recruiting as well. Subsequent to interviewing an expert within this field, some general conclusions could be derived regarding her view on online recruiting in general and programming challenges in particular. It was deemed that programming challenges mainly attract junior developers and younger people in general due to them having more spare time to put on completing these types of challenges. This is not ideal for recruiters, as they generally want a larger span of candidates. The expert recruiter also mentioned that generally, programming challenges do not give a hint about important qualities of a programmer, for example flexibility, since the candidates generally only need to know one programming language to complete the challenges.

As a third step in the phase of gaining inspiration, gameplay design patterns were elicited from some existing programming challenges that the stakeholder company Future Skill had created. This was done in order to get knowledge of what gameplay design patterns already were integrated in the challenges. This in turn should facilitate the development of new challenges, as one would know what patterns would introduce new functionality and dynamics to the challenges. As a fourth and final step, a list of patterns that would be interesting to introduce in the programming challenges was compiled.

5.1.2 Ideation

Moving from the inspiration phase to the ideation phase, this section describes the process of generating ideas for how the programming challenges could be made more engaging. This does not necessarily mean coming up with ideas for completely new challenges, but rather how the platform could be altered or redesigned to achieve the goal of increased engagement amongst users.

In order to limit the design space, and as a result of the questionnaire answers, three personas were created to represent three different types of users who are all interested in taking programming challenges for different reasons. The first persona, see Figure 5.1, represents a young, overconfident male who seeks to destroy his opponents.

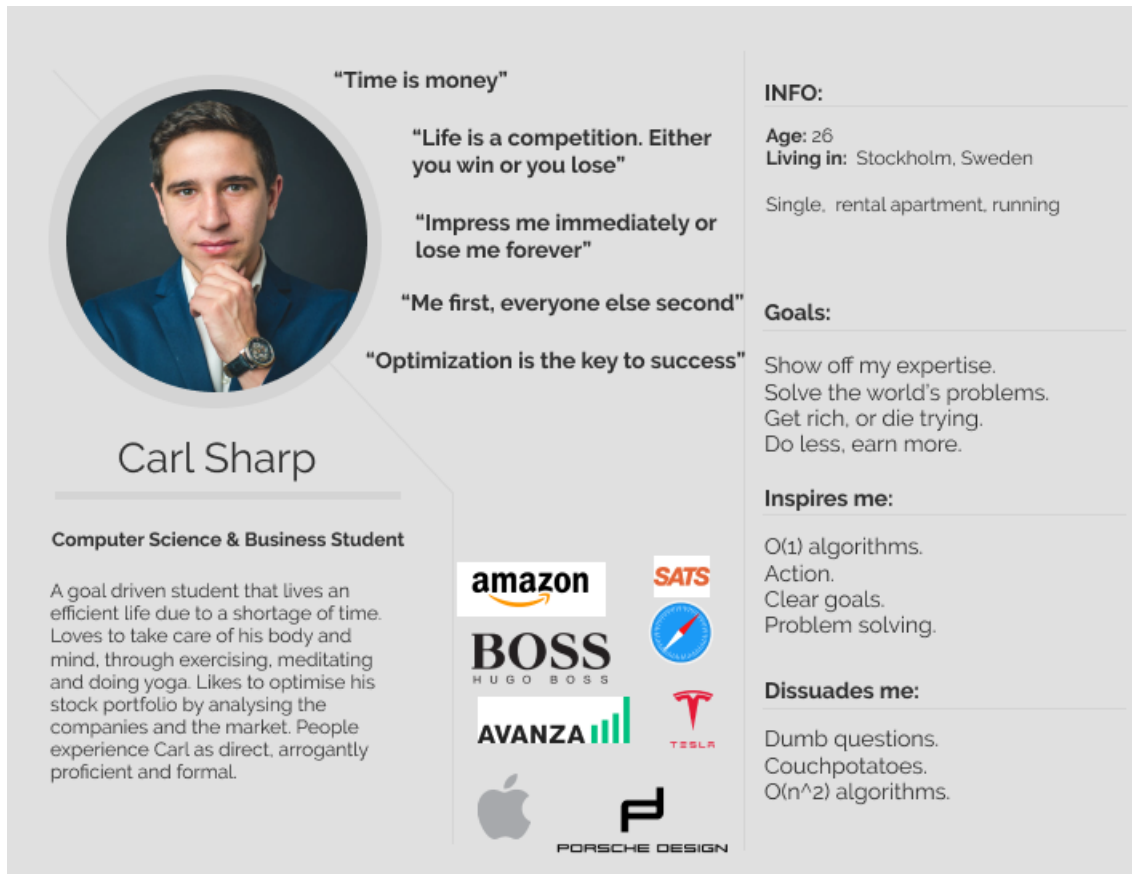


Figure 5.1: Overconfident persona called Carl Sharp

The second persona, see Figure 5.2, represents a more selfless and considerate user, looking to help her peers rather than compete against them.



Figure 5.2: Selfless and considerate persona called Allie Sembler

The third and last persona, see Figure 5.3, represents an experienced, but lonely user looking to socialize and explore new challenges in life in general and programming in particular.



Figure 5.3: Experienced and lonely persona called Hans Askell

Subsequent to defining the personas, several brainstorming sessions were held in order to generate ideas on how to increase user engagement on a platform with programming challenges.

The first brainstorm session consisted of two parts: one five-minute brainstorm on regular features, and one ten-minute brainstorm on crazy features. The distinction between a regular and a crazy feature can be described as a regular feature being reasonable and somewhat standard, while a crazy feature is more out of the box. The brainstorm on regular features resulted in eight distinct features:

- Friend list for matchmaking and/or challenging
- An overview visualization with details on demand
- An overview of the challenges that the user has taken with detailed info on number of submits, time spent, lines of code, and ranking over time.
- Visual statistics of progress and ranking over time
- Avatar
- A tag (for example "Looking for a job within frontend", or "Just here to have fun")
- Achievements (for example "Completed five challenges")
- Badges (for example "Won tournament in Space Challenge")

A general consensus of these features is that giving the user information and statistics of his or her performance on the platform is of high importance.

The brainstorm session on crazy features resulted in 13 distinct features:

- Collectible items and lootboxes
- The character traverses a world, beating different types of challenges in a story line with side quests
- Building a character - level up, distribute attribute points and play with this characters in challenges
- Daily quests
- Different named rankings, for example "rookie" and "code monkey"
- Tinder-like swiping for finding interesting challenges and/or people to challenge
- Showing statistics on how many people the user has beaten
- Let users rate others' solutions
- Measure $O(n)$ on the user's solutions and display an average
- Let the top ranked players pick the jobs of their liking
- Add an element of gambling, letting users bet on winners
- Add a money incentive
- Add an in-game currency: "CodeCoins"

A lot of these features are merely brought in from other games and put into the context of programming challenges. They are not necessarily out of the box or crazy, but the mindset when conducting this segment of the brainstorm session was more open and allowed for these types of ideas.

As a result of the brainstorm, a bunch of ideas for potential features were generated. As a step towards generalizing these features into broader concepts, a variation [45] of the KJ method [28] was applied. This process resulted in six categories of features, namely:

- Community / Social tools
- Acknowledgement of success
- The big picture
- Competitive measures
- Personal description
- Diegetic aspects
- Monetary allurements

The first four of these categories received votes in the subsequent balloting, and the reasoning behind these four receiving votes were largely based on the preferences of the personas. Hans Askeff is looking to widen his social circle, which is why *community / social tools* received votes. Carl Sharp likes to prove his worth against other opponents, hence *competitive measures*. Allie Sembler is uncertain of her abilities as a programmer, and would like to get *acknowledgement of success*. Having put names on the broader categories of features that were desired additions to the current platform, it was only natural to continue the process with a brainstorm session focused on these four categories. This session resulted in many new ideas for features. A list of these ideas can be found in Appendix A. After the idea generation phase was completed and before the start of a possible concept development phase, a discussion was held within the group, as well as with a supervisor, and as a result of those discussions, it was determined to make a change of focus for the thesis, as described in Section 5.2. This is not to say that the work done up to this point

was a waste, because the main body of work could still be utilized as inspiration for the new main focus. Even though the new focus was mainly about applying gameplay design patterns on a recruitment process, and not increasing user engagement in programming challenges, some ideas from previous brainstorming sessions stemmed from gameplay design patterns and could therefore be utilized when designing the recruitment game as well.

5.2 Change of Focus

Subsequent to the first iteration of the design process, a decision was made to change the focus of the thesis. This decision was made due to the earlier focus not being aimed at the recruiting process in general, but rather at specific elements of a recruiting process, namely the programming challenges. It was deemed more interesting to focus on analyzing and applying gameplay design patterns on the recruitment process as a whole rather than specific parts of it. The programming challenges are one aspect of moving towards a more digital and gamified recruiting process, and improving them could in and of itself help increase user engagement to a certain extent. However, using programming tests and challenges are relatively widespread methods in modern recruiting processes. It was deemed more innovative to look at the entire recruiting process from a game standpoint, rather than gamifying parts of it that were already closely related to games.

5.3 Iteration 2: Analyzing and applying gameplay design patterns on a recruiting process

This section describes the second iteration in the design process overall, and the first iteration with the new focus. In short, this iteration consists of defining a recruiting process to base the design on, analyzing this recruiting process in order to identify existing gameplay design patterns, adding additional gameplay design patterns that would be interesting to explore in a recruiting game process, categorizing all the patterns in the collection as either a mechanic, a dynamic or an aesthetic, as well as developing a game concept based on this collection of gameplay design patterns.

5.3.1 Defining a Recruiting Process

It is difficult, if not impossible, to find a common recruiting process between companies within the IT sector. Everybody does things slightly differently, and studies on what process is the most effective becomes subjective. In this thesis, having a specific recruiting process to base reasoning and argumentation off is not of high importance, since at an abstract level, they are all similar. However, an argument can be made that introducing a concrete recruiting process to base the thesis on can be beneficial for the sake of clarity, and that is the purpose of this section.

Basically, one can identify three main stages of a recruiting process in the context of programming and software development. The first stage consists of the candidate

creating a resume, writing a cover letter and sending in an application to a company. Between this first stage and the second stage, the recruiting company gets a chance to either approve or disapprove the candidate from continuing the process. If approved, the candidate is invited to an interview, which mainly focuses on personality, but also scrapes the surface of the candidate's competence. This is the second stage. The third and final stage in this simplified model of a recruiting process is the competence test. This is where candidates really get a chance to show that they possess the required competence for the role in question. A visualization of this recruiting process is shown in Figure 5.4.

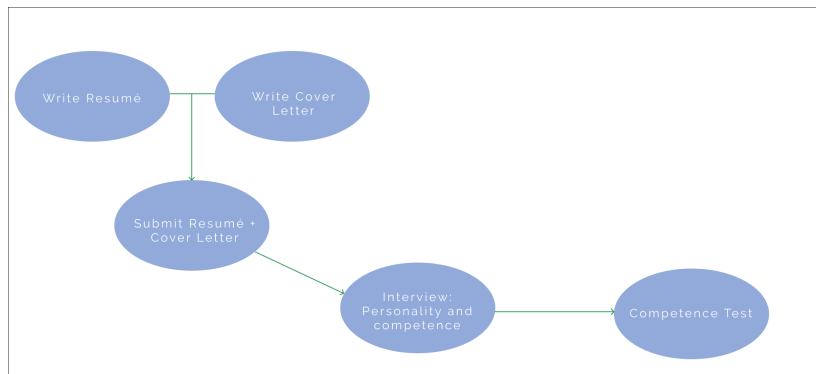


Figure 5.4: An abstract recruiting process

5.3.2 Analyzing Gameplay Design Patterns in a Recruiting Process

If one was to look at the process of recruiting as a game, which is the point of this thesis, one would see that it is possible to identify gameplay design patterns (a framework introduced in Section 3.1.3) in such a process. The following list represents a selection of significant gameplay design patterns that have been identified in a general recruiting process (such as the one introduced in 5.3.1), and is later used as a design foundation (see Section 5.3.5). The full list of identified patterns is found in Appendix C. The structure of the list is as follows: The bold title represents the name of the gameplay design pattern in question ((taken from [1]), the first bullet point represents a quote of a one-line description of the pattern taken from [1] and the second bullet point represents a description of how the gameplay design pattern in question correlates to the recruiting process. The noun within parentheses describes at what level in the MDA framework (introduced in Section 3.1.4) the pattern is considered to belong.

Abilities (Mechanic)

- *"Actions that agents can do which allow players to affect game states"*
- Every candidate has a set of abilities, and using / showing them will affect how they perform in the recruiting game.

Agents (Mechanic)

- *"Diegetic game elements that can be interpreted as having goals in game worlds, and working towards those goals"*

- Both candidates and recruiters can be seen as agents, as they both have their own goals, and continuously work towards achieving them.

Arithmetic Progression (Dynamic)

- *"A linear relationship between the effort put into an action and its potential reward or risk"*
- Generally, the more effort a candidate puts into the recruiting process, the better the performance / reward. While maybe not always linear, as recruiting is subjective, one could still see a correlation between effort and success.

Asynchronous Gameplay (Dynamic)

- *"Gameplay that does not require - or demands - that not all players are playing at the same time"*
- A recruitment process does generally not require all candidates to be actively participating simultaneously. Rather to the contrary. A recruiting company can not handle all candidates at once. People can drop in and out, people are at different stages in the process etc.

Campaigns (Dynamic)

- *"A series of designed gameplay experiences for a game that are intended to be played in sequence"*
- The different stages of a recruiting process can be correlated to the designed gameplay experiences. A recruiting process is generally progressed through in sequence.

Challenging Gameplay (Aesthetic)

- *"That players experience the gameplay as difficult or challenging their abilities and skills"*
- A recruiting process is generally intellectually challenging for the candidate, as the whole point is proving their abilities, both technically and socially.

Characters (Mechanic)

- *"The abstract characteristics of diegetic persons"*
- The diegetic persons in this case is the candidates and recruiters, which all have different characteristics.

Character Development (Dynamic)

- *"Changes in characters' abilities, skills, or powers as part of gameplay"*
- In the eyes of the recruiter, the candidate is building its character throughout the recruiting process, as the recruiter gets more information about the candidate.

Checkpoints (Mechanic)

- *"Locations in game worlds which signify game state changes directly related to the progress of some goal"*
- In the case of a recruiting process, the checkpoints are not necessarily locations by definition, but rather metaphorically. Progressing from one stage of the recruiting process to the next can be viewed as a checkpoint. The state changes significantly, as new tasks become available when reaching the next stage.

Death Consequences (Mechanic)

- *"Gameplay consequences of avatars or characters dying"*
- The death of a character is in the case of recruitment is of course metaphorical. A candidate being denied further progress in a recruiting process can be viewed as a character dying in a game. And the consequences of this death is that the candidate generally can no longer apply for the same job, at least within the foreseeable future.

Delayed Effects (Dynamic)

- *"The effects of actions and events in games do not occur directly after the actions or events have started"*
- This rings true for a recruiting process as well. The effects of taking an interview or doing a competence test are not revealed immediately, but will rather take some time to get feedback on.

Early elimination (Dynamic)

- *"The ending of players' game sessions well in advance of the end of game instances"*
- This depends on how one defines a game instance, but the game instance from a recruiter's point of view might be the process until a role is fulfilled by a candidate. This means that for most of the candidates, the game session will end in advance due to them being eliminated at an earlier stage of the recruiting process.

Enemies (Mechanic)

- *"Game elements that are actively hindering players to complete game goals"*
- One could correlate the recruiter to an enemy in the game. They are constantly performing checks to see if a candidate is a good fit for the role. This could be seen as them actively hindering candidates from completing the goal of getting the job.

Extended Actions (Dynamic)

- *"Actions that take so long to complete that they require players to miss opportunities to perform other actions in order to complete them"*
- If an action is looked at as going through a recruiting process at one company, that action might hinder candidates from performing other actions such as going through the recruiting process at another company.

Finale Levels (Mechanic)

- *"Levels or other enclosed gameplay areas that are constructed so they are experienced as the final part of a narration arc and resolves that arc"*
- The final interview, a final competence test or such can be looked at as the final level of the recruiting process.

Fog of War (Dynamic)

- *"Hiding information about parts of game worlds that are not being observed or have not yet been explored by players"*
- The candidates do generally not have too much information about what will happen later on in the process.

Further Player Improvement Potential (Dynamic)

- *"That players have the possibility to increase their skills in handling the gameplay"*
- As candidates gather experiences in different recruiting processes, they will most likely become better at performing them next time.

Game Masters (Mechanic)

- *"Facilitators of game worlds, and of players' interactions with these worlds"*
- The recruiters could be correlated to game masters, as they, to a certain extent, facilitate candidates interactions with the recruiting process.

Game Over (Mechanic)

- *"The event which makes a player no long[sic] able to participate in the gameplay of a game instance"*
- Failing to proceed in a recruiting process generally means that the candidate can not apply again during the same instance of the recruiting process.

Last Man Standing (Dynamic)

- *"Gameplay where players or teams are actively trying to eliminate each other to be the last survivor"*
- While candidates might not feel that they are actively trying to eliminate other candidates, in fact, that is what they are trying to achieve. They want to be the last man standing in the recruiting process and get the job.

Levels (Mechanic)

- *"A level is a part of the game in which all player action takes place until a certain goal has been reached or an end condition has been fulfilled"*
- A recruiting process generally consists of different stages or levels, in which candidates can fulfill certain goals and either advance or get eliminated.

Main Quests (Mechanic)

- *"Quests whose completion provides the main winning condition of games"*
- Each *level* in a recruiting process generally consists of one or more main quests that the candidate must complete in order to advance towards the next level.

Penalties (Dynamic)

- *"Effects on the game state that are negative to players regarding their position, progress, or abilities"*
- Each *level* in a recruiting process generally consists of one or more main quests that the candidate must complete in order to advance towards the next level.

Player Characters (Mechanic)

- *"Characters in games that are under players' direct control or represent the players role in the gameplay"*
- An argument can be made that people choose to represent themselves as different characters depending on which job they are applying for. One could say that when people are creating their resumes and writing their cover letter, they are in fact creating a character, which is under their direct control throughout the recruiting process.

Player Elimination (Mechanic)

- *"The forced endings of players' game sessions before game instances are finished"*
- The elimination can be correlated to a recruiter telling candidates that they will not advance to the next stage of the process, because they have not fulfilled the required goals.

Player-created Characters (Mechanic)

- *"Characters whose creation players' have had some influence over"*
- The same reasoning as for *Player Characters* works here. It is the candidates that are creating the characters, which explicitly means that they have had influence over them.

Possibility of Graceful Surrender (Mechanic)

- *"The ability for players to leave gameplay or surrender to other players without negatively affecting the gameplay for all other players of a game instance"*
- A character electing to leave a recruiting process will not affect the other candidates' experiences negatively.

PvE (Mechanic)

- *"Gameplay where game systems, rather than other players, provide challenges to players"*
- Since the candidates are never really interacting with their opponents, one could say that the recruiting process is a PvE game, as the system (the recruiting company) provides the challenges for the candidates.

Quizzes (Mechanic)

- *"Collections of questions asked to players as part of gameplay."*
- Questions are asked to the candidates during for example interviews, and often as part of the competence test.

Races (Dynamic)

- *"Gameplay goals that need to be worked against under pressure, often but not necessarily in competition against others"*
- Oftentimes, the competence test has some sort of time-limit and could be seen as a competition with the other candidates.

Real World Knowledge Advantages (Dynamic)

- *"Games where players can make use of specific real world knowledge to their advantage"*
- This one is almost trivial. It is evident that the candidates are using their real world knowledge to their advantage in the recruiting process.

Real-Time Games (Mechanic)

- *"The progression of game time during play is tied to the progress of real time"*
- This one is also trivial. The time during the recruiting process progresses in the same way as real time.

Role Selection (Mechanic)

- *"Selecting what gameplay abilities one will have by choosing from a limit[sic] number of roles"*

- If one looks at applying for a specific role at a company as limiting the abilities a candidate will have during the recruiting process, this pattern makes sense. The candidate will only be able to utilize the abilities required for the role during the process.

Skills (Dynamic)

- *"Representation of how likely diegetic agents are to succeed with a type of activity that can be improved through experience"*
- As with everything, candidates can have different skill levels which will determine how well they will perform in the process.

Solution Uncertainty (Dynamic)

- *"Uncertainty in a game which stems from not being aware of an existent solution to a challenge"*
- The challenges provided to the candidate in a recruiting process oftentimes don't have an obvious solution. For example, knowing what the 'correct' answer is to an interview question is unintuitive.

Tension (Aesthetic)

- *"The feeling of caring about the outcome of actions or events in a game without having full control over them"*
- This is true for most candidates. They care about getting the job, but really has no control over that outcome, since it is the recruiter's subjective assessment of the candidate's performance that determined whether the candidate advances in the process or not.

Time Pressure (Dynamic)

- *"That gameplay imposes a sense of need for complete[sic] actions or goals quickly"*
- For example, in an interview, the candidate oftentimes feels a sense of time pressure, as they don't have unlimited time to answer a question. Also, the competence tests generally comes with a time limit.

Winner Determined After Gameplay Ends (Mechanic)

- *"Games where the winner(s) are determined after gameplay ends"*
- The game, for a candidate, ends when all stages of the recruiting process have been completed (or when the candidate has been eliminated). The recruiting company decides, after a candidate has completed all stages, whether or not that candidate gets a job. This means that the winners are determined after candidates' gameplay are over.

As mentioned earlier in the section, this list is merely a selection of the gameplay design patterns identified in a recruiting process. These patterns are presented here because they were selected as the basis for the design concept described in Section 5.3.5. These patterns were selected based on the subjective opinion that they describe the essence of a recruiting process.

5.3.3 Adding Additional Gameplay Design Patterns

Subsequent to the analysis and elicitation of gameplay design patterns in a general recruiting process, a decision was made to add additional patterns to the collection.

This decision was made in order to diversify the design foundation. One could argue that if only the patterns that were identified in a recruiting process were utilized to develop the game concept, then that concept would merely be a digital reflection of the analogue recruiting process. Once again, the collection of gameplay design patterns, available at [1], was traversed in order to find interesting patterns, which were not included in the list detailed in Appendix C, to explore in a recruiting game concept. The thought process when sieving patterns was to find patterns which met one of the following criteria:

- It should have some relation to the process of recruiting, or at least make sense from a recruiting standpoint
- It should be innovative and provide unheard of experiences in the context of recruiting

The result of this elicitation is shown in the list below. The list is structured as follows: The heading describes the name of the gameplay design pattern (taken from [1]), the first bullet point is a quote from [1] representing a one-sentence definition of the pattern, and the second bullet point describes the reasoning behind why the pattern was regarded as interesting for further exploration. The noun within parentheses describes at what level in the MDA framework (introduced in Section 3.1.4) the pattern is considered to belong.

Near Miss Indicators (Dynamic)

- *"Means of providing players with explicit information about how close they or others were to achieving a goal when they have failed to achieve it"*
- Providing the players with information about how close they were to achieve a goal, when they failed to achieve it could be an interesting way of potentially increasing engagement. This relates to the lack of feedback in many recruiting processes.

Progress Indicators (Dynamic)

- *"Information about players' current progress towards closures in addition to the configuration of game elements involved"*
- Something often missing in a recruiting process is an overview of one's progress towards the goal of getting the job.

Scores (Mechanic)

- *"Numerical values used in games to determine winners"*
- In a recruiting process, the recruiters might put an internal score on candidates, but this score is seldom revealed to the candidate. By introducing a score, the candidates can get an idea of how well they performed in the recruiting process.

High Score Lists (Dynamic)

- *"The storing of scores after games have finished so they can be compared"*
- Having a public list of scores could potentially increase the competitive part of the game. It is seldom, if not never that the candidate gets this information about other participants in the recruiting process.

Achievements (Mechanic)

- *"Goals whose fulfillment is stored outside the scope of individual game sessions"*

- Something that one almost never gets from a recruitment process is achievements, for example "Completed the first interview at company X". It is in general very binary; either candidates get the job or they do not. By introducing achievements in the recruitment process, candidates can get some sort of reward for participating and completing certain stages of the recruitment process, even though they might not get the job in the end.

Penalties (Dynamic)

- *"Effects on the game state that are negative to players regarding their position, progress, or abilities"*
- Introducing penalties in the game could work as a method of further solidifying the tension and the challenge that players experience.

Chat Channels (Mechanic)

- *"Text-based message channels in games"*
- Communication with other candidates in a recruitment process is very rarely occurring. It could be interesting to explore whether or not this would be something that the candidates would utilize, and what kind of communication would occur in these channels.

Storytelling (Dynamic)

- *"The act of telling stories within the game"*
- Packaging the recruiting process in a story with a common thread is an interesting aspect to explore, especially in the sense of increased user engagement. One of the biggest factors as to why people watch movies, read books and to a certain extent play games is due to the authors narrating an interesting story. If this can be incorporated in a recruiting process, the engagement amongst users has the potential to skyrocket. Also, putting the recruiting process in the context of a fictional story can help dedramatize a for many nervous and stressful process.

Thematic Consistency (Dynamic)

- *"That the characters and objects depicted in a game world are thematically consistent, as is their behaviour"*
- Goes hand in hand with *Storytelling*, and goes a long way towards keeping users engaged during the whole process, and not question the validity of the story that is being told.

These patterns were merged with the collection of identified patterns, and this merged list makes up the design foundation for the concept development.

5.3.4 Defining The Core Gameplay

Subsequent to the elicitation of a design foundation in the form of a collection of gameplay design patterns, the patterns that would make up the core part of the gameplay were selected. This selection was made as a measure to facilitate future design decisions, and to know what patterns to focus most on implementing in the concept. The result of this selection is shown in Figure 5.5

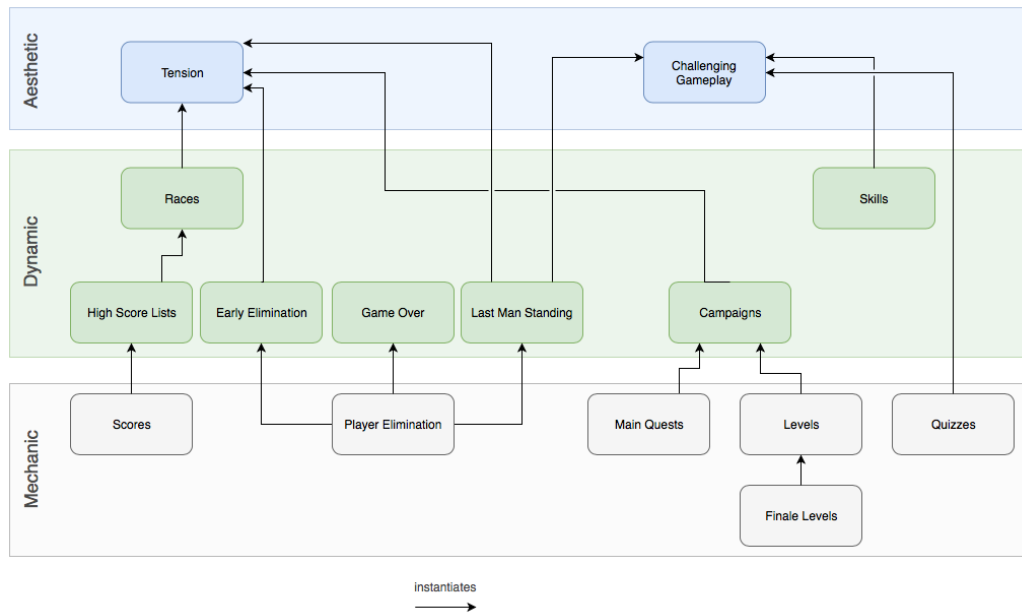


Figure 5.5: The gameplay design patterns that make up the core gameplay of the concept, and their relation to each other.

In accordance with the MDA framework [24], looking at Figure 5.5, the boxes with grey backgrounds represent mechanics, the boxes with green background represent dynamics and the boxes with blue background represent aesthetics. The arrow in between two patterns state that a relation exists between the patterns, and that the pattern where the arrow goes out from instantiates the pattern where the arrow goes in to, meaning that adding the first pattern will lead to the second pattern being introduced in the game. What relation patterns have to each other is context-dependent, and can vary between different designs, depending on how the patterns are used. Hence, the relations in Figure 5.5 are not an absolute truth; they could look different in another concept. The graph merely represents the *desired* interactions of patterns within the design, and not necessarily the *actual* behaviour.

5.3.5 Designing a Game Concept Based On Gameplay Design Patterns

As a result of analyzing existing gameplay design patterns in a recruiting process, as well as coming up with additional patterns that would be interesting to explore in a recruiting context, a long list of gameplay design patterns was compiled. As a next step in this iteration of the design process, ideas for a concept were developed, based on the list of gameplay design patterns.

5.3.5.1 Concept Creation

As a start in the development of a game concept, the earlier mentioned recruiting process (see Figure 5.4) was transformed and abstracted into the steps of an "ordinary game" (see Figure 5.6) through the use of the collection of gameplay design

patterns described in Section 5.3.2. Looking from the top down, there are different stages of a recruiting process that a candidate goes through before getting an offer. This is conceptualized in this game by utilizing the pattern *Levels*. The user of the system is now seen as a player and plays along a game story. The GDP *Levels* was used as an analogy for the different stages in a recruiting process, where each level has its own goal. The levels are coherent and are played sequentially, hence the concept also implements the gameplay design pattern *Campaign*.

Level one was created together with the *Player Characters* pattern in mind. The players are here supposed to create a character to represent themselves. This connects to the first step in a recruitment process, to create and send a resume. This is conceptualized as creating a character in a game.

Generally in games, after the player has created its character, the gameplay starts. This is the case for this game as well. The idea of level two was that after creating a character, the player should continue on by building on its abilities. The patterns that realize the building level are *Character Development* and *Abilities*. Level two also reinforces the game story by preparing for the player's first raid. Level two therefore becomes a metaphor for the first interview in a recruiting process. The candidates get a chance to start proving that they were not lying on their resume with some initial competence tests.

While the second level gauged the contenders from the pretenders with simple tests, at level three is where the first real test comes. The players have now built and practiced with their characters and it is time for the first raid. The test in this level differs from the one in level two by being more complex and involving several different steps toward a larger goal. The first raid implements the pattern *Main Quests*, hence must be completed before being able to continue to the next level.

Level four follows the *Further Player Improvement Potential* pattern and should allow the player to once again prepare and improve their character before the last level is reached.

Level five is the last and biggest challenge, and it implements the pattern *Finale Levels*. The player should have an experience here that gives a feeling of "if level five is completed, the game is won". Completing all levels would lead to the final goal of the game, to get a job offer.

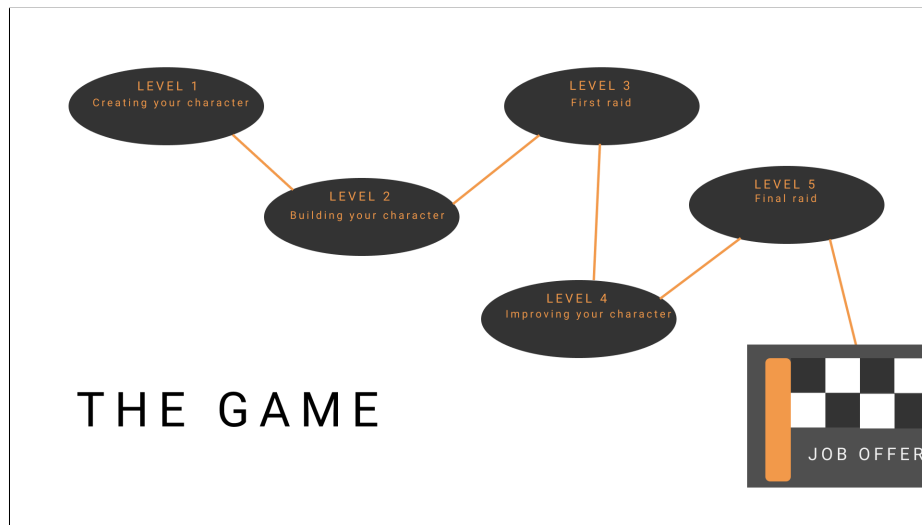


Figure 5.6: Concept model

5.3.5.2 Lo-Fi Wireframes

With a concept model and description developed from gameplay design patterns as a foundation for the design, low fidelity wireframes were used to ideate around possible illustrations of the concept. Figure 5.7 is a wireframe of the first screen that the player would have encountered in the game. It only consists of a picture, job description and a play button to start the recruitment game.

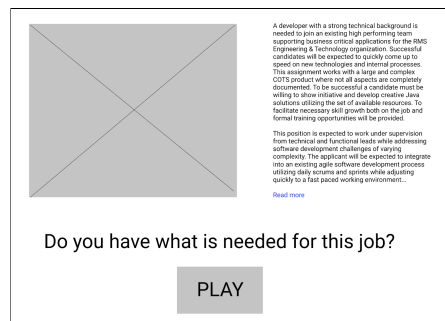


Figure 5.7: Early stage wireframe idea, showing the entry point of the game.

The next wireframe screen, see Figure 5.8, shows the first idea of level one. The player creates a character by adding a picture and a name. The patterns *Progress Indicators* as well as *Checkpoints* were merge and visualized as a progress bar with circles at the top. The lines that connect the circles symbolize the different levels the player needs to complete and the circles symbolize the different checkpoints.

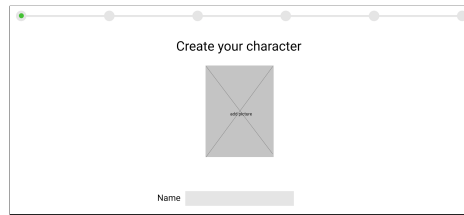


Figure 5.8: Early stage wireframe idea, showing the creation of the character.

As the player continues to the next level, the progress bar and checkpoints are filled in, see Figure 5.9. An additional idea for level two that was evoked at this wireframe stage was that the player should receive a rank and a level, which is visualized in the left half of the screen. The rank would show the player's placement in the game compared to other players and if a player do not maintain a high rank, then that player is eliminated. The idea was that if a player is rank one by the end of the game, then that would imply winning the game. This works as a metaphor for the recruiting process of a job, only one can get it and others will get removed from the candidate list if they do not meet the expectations. These ideas together implements the patterns *Last Man Standing*, *Player Elimination* and *Early Elimination*. Furthermore, on the right half of the screen, the *Abilities* pattern is visualized. This area allows the players to develop their characters in different categories of abilities.

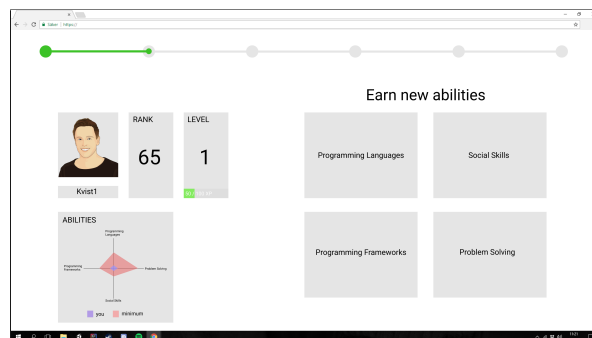


Figure 5.9: Lo-Fi mockup of level two, showing a dashboard.

5.3.5.3 Me-Fi Mockups

As a next step in this design iteration, a medium fidelity (Me-Fi) design was created. The Me-Fi design focused on developing the content of each level, as it was important that all levels together resulted in a coherent game story. Together with this, the addition of a look and feel and a colour theme was made. The specific colour theme (seen in Figure 5.10) has dark grey as the main colour and a high contrast orange as secondary colour. This colour theme was supposed to create a more gamified look and feel as opposed to the formal colours of light grey and white. Although the same gameplay design patterns as in the concept model and earlier wireframes are implemented, a change in the creation of the character was made. To reinforce the *Character* pattern, the player can now actually create the look of the character by

choosing a face, a hair style and eyes. For recruitment practicalities, more details about the player were added as a form to be filled in.

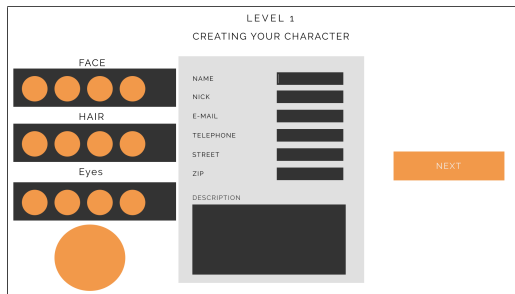


Figure 5.10: Me-Fi mockup illustrating the creation of a character.

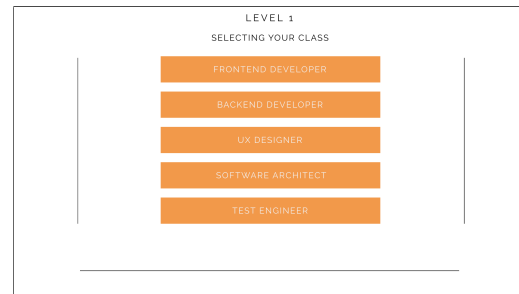


Figure 5.11: Me-Fi mockup illustrating the selection of a class.

To be able to tailor the upcoming tests in the game, level one was developed further with the selection of a class, see Figure 5.11, which works as a metaphor of the different roles one can take in a software development team. Selecting a class means that the player selects a set of abstract characteristics these roles tend to have, hence classes supports the already implemented *Character* pattern.

The function of developing abilities has been introduced in section 5.3.5.2 and an improvement on the *Ability* pattern was here done by adding a screen where the player gets to choose their top five abilities, see Figure 5.12 and Figure 5.13. This does not only give data to the system (to tailor the game), but it also reinforces the building of a character.

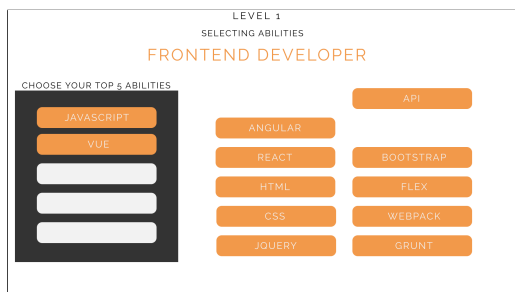


Figure 5.12: Me-Fi mockup illustrating the selection of abilities.

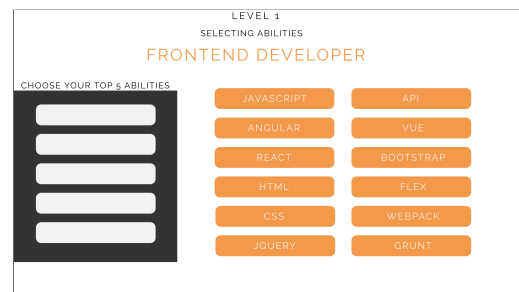


Figure 5.13: Me-Fi mockup illustrating selected abilities.

It is not until after the character is created, and the class and abilities are chosen that the player enters the actual gameplay. This is where the players practice and prove their abilities. Figure 5.10 shows how the title "main quests" works as a metaphor for the different competence tests a candidate could face from a recruiter. This implements the pattern *Main Quests*. Furthermore, the pattern *Side Quests* is implemented as well, by having non-mandatory quests. An addition in this mockup is also the experience points (XP) that the player receives as a reward from playing the different quests. The XP received depends on the performance of the player, but also the difficulty of the quest. XP is a feature implementing the *GDP Rewards*.

The total XP is also the players measurement on how well they have performed during the entirety of the game, hence it also implements the *Scores* pattern.



Figure 5.14: Me-Fi mockup illustrating the building of a character.



Figure 5.15: Me-Fi mockup illustrating the first raid.

When entering level three, players are notified whether or not they have accomplished the goals of the level or if they have been eliminated due to of other players gaining more XP than them. The quest in level three is visualized with a larger square, as an attempt at showing that it contains a larger test (therefore named raid to fit the game metaphor), as seen in Figure 5.15. The player can also see that there is more XP to gain from the raid than the quests from level two.

Subsequent to completing the raid, the players will again be evaluated on their total XP in order to eliminate players that have not reached a certain amount. The players that succeed to stay in the game move on to level four, where they get a chance to once again practise and improve their abilities. This is realized by providing them with multiple tailored quests, see Figure 5.16. The tailored quests should work as a warm up before the last and most challenging level.

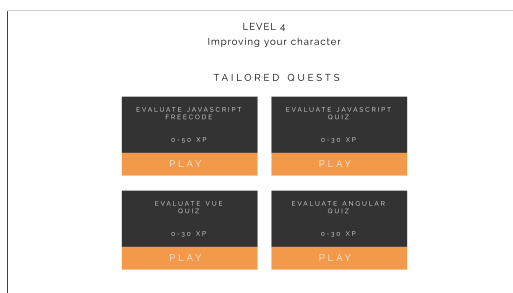


Figure 5.16: Me-Fi mockup illustrating improving a character.

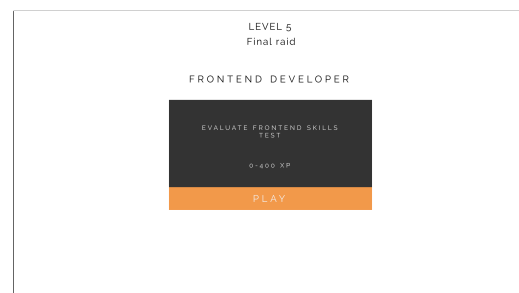


Figure 5.17: Me-Fi mockup illustrating the final raid.

Level five contains the final raid. This level should contain a test of substantially higher difficulty. It combines the technologies from earlier quests into one test and evaluates the player on the behalf of the class that is played. The player can see that the XP reward is even higher on this raid, see Figure 5.17.

The player with the most total XP after completing this final raid will get a job offer.

Another part of the game design was created in parallel with the Me-Fi mockups, and this part focused on developing the content for each level. The parallel work

focused on the overview of the game state by for example visualizing progress, rank and position, as seen in Figure 5.18. The left column contains the player's details that was filled in during the character creation. The progress bar (circles on a line) works as described in the Lo-Fi wireframes section 5.3.5.2. The square below the "Rank" title shows the player's current rank, total gained XP and at the bottom there is a hint about how close the player is to rank up. The hint implements the *GDP Near Miss Indicators*. The bar below the "Position" title shows a visualization of the player's current position. The dark grey area symbolizes all the players that are playing the game (other players are seen as enemies and implements the *GDP Enemy*), the light grey area symbolizes the enemies that have been eliminated and at last the small orange vertical line symbolizes the players position. Furthermore, the top five abilities that was chosen on level one is shown in this overview section.

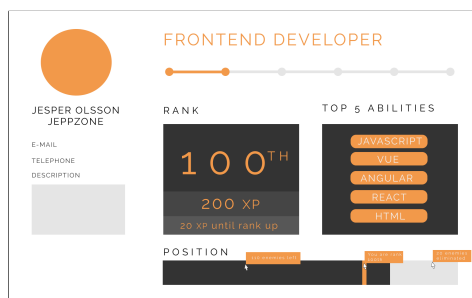


Figure 5.18: Me-Fi mockup showing progress, rank and position visualizations as well as hover feedback.

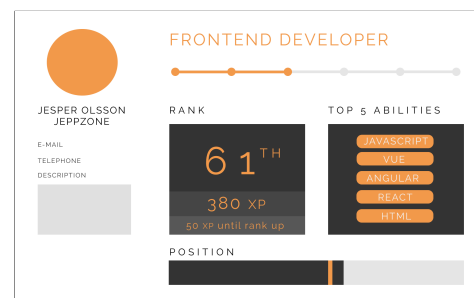


Figure 5.19: Me-Fi mockup showing progress, rank and position visualizations at rank 61.

Figure 5.19, Figure 5.20 and Figure 5.21 shows a possible scenario if the player manage to stay in the game and succeed through the levels. Figure 5.19 shows that the player is on level three (the progress bar is filled up to three circles), but even if the player has reached level three the position visualization indicates that the player is close to getting eliminated (orange vertical line close to the light grey area). This data puts a higher emphasis on the elimination system and could possibly evoke the *GDPs Races* and *Tension*.

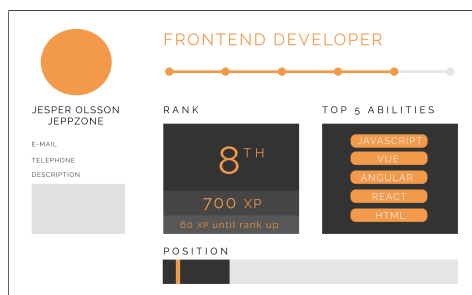


Figure 5.20: Me-Fi mockup showing progress, rank and position visualizations at rank 8.

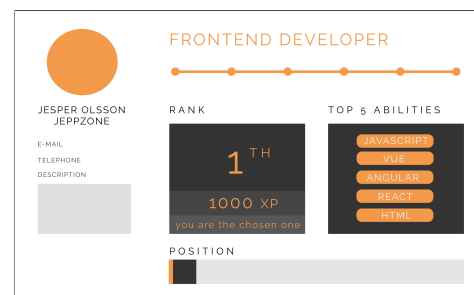


Figure 5.21: Me-Fi mockup showing progress, rank and position visualizations at rank 1.

Figure 5.20 demonstrates that the player has reached level five (the final raid) and has here gained a better position than previously on level three.

Figure 5.21 visualizes a winning position. The player has reached the last step on the progress bar and is ranked number one.

5.3.5.4 Hi-Fi Mockups

The medium fidelity mockups got developed further and many iterations later resulted in a series of coherent, high fidelity mockups that together represent the whole game and connects to the very first concept. The step from Me-Fi to Hi-Fi is here quite large and many changes have been made in between. This section will walk through the design and point out what changes have been made, although there are a lot of features still remaining from earlier stages.

First of all, the design has been put into the web browser to represent a web application, see Figure 5.22. This Hi-Fi design is made to fit into a "community-like" website, where users can do different kinds of programming challenges, discuss with like-minded, view statistics and then play this recruitment game. The top right corner shows who is logged in on their community account, here "Jeppzone" is the user. The bell placed immediately to the left of the name is for eventual upcoming notifications.

New metaphors have been added, one of them is "Servers", which has been added because the first thing a person does when searching for a job is actually choosing what company to apply for a position at. This choice is conceptualized by letting a player select what server to play on. The servers are split into two different sections or game modes: Adventure and Campaign. If a player does not care about game mode, the ability to select a server at random exists.

The Adventure game mode represents applying for a consulting company, since the candidate will get to explore a lot of different assignments as a consultant. An adventure in games generally means exploring the game world in a somewhat more free way than in normal campaign modes.

The Campaign game mode represents applying for a product company, since at a product company, the candidate is likely working towards the same goal all the time, with different sub-goals of course. A campaign mode in a game generally means playing to reach a certain goal by solving smaller tasks that in the end leads to fulfilling this goal.

Selecting a server at random in this game means applying for a recruiting company. The candidate is being evaluated by the recruiting company, which in turn will decide what type of company they think the candidate is best suited for. In games, selecting a random game mode is commonplace as well.

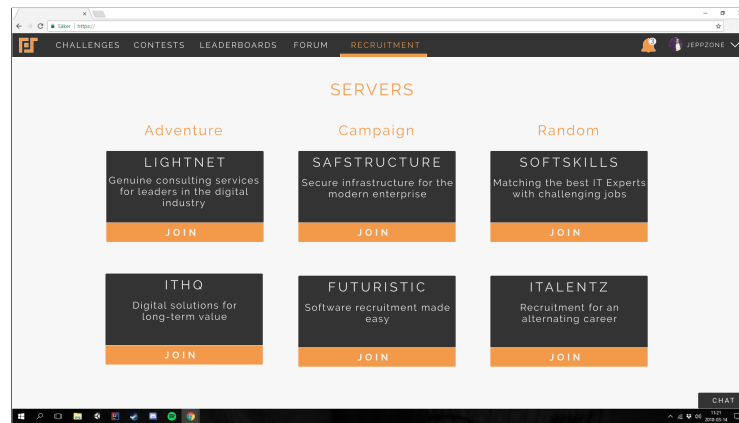


Figure 5.22: Hi-Fi mockup of the starting view on the recruitment tab, showing different servers to join.

In Figure 5.23, earlier Me-Fi parts have been put together to form a more polished user interface. It is here, at level one, that the player creates their character after joining a server. One thing that has been added is the time countdown, which limits the amount of time the player has to finish each level, which implements the GDP *Time Pressure*.

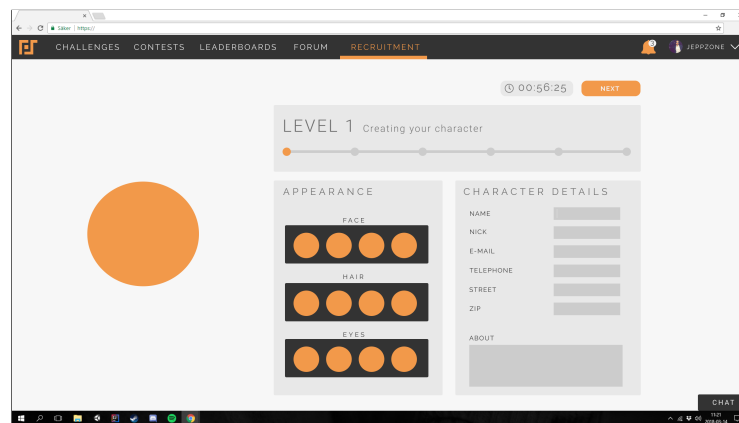


Figure 5.23: Hi-Fi mockup showing the creation of a character.

The next two views, see Figure 5.24 and Figure 5.25, still represents level one. The player chooses a class to play and presses next.

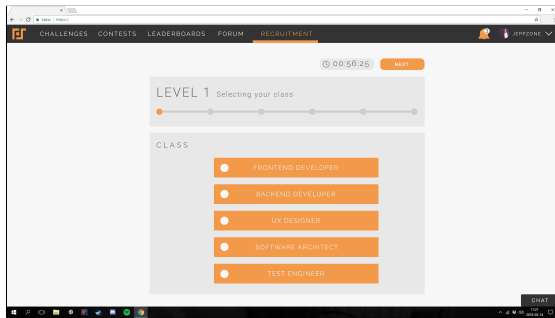


Figure 5.24: Hi-Fi mockup showing the selection of a class.

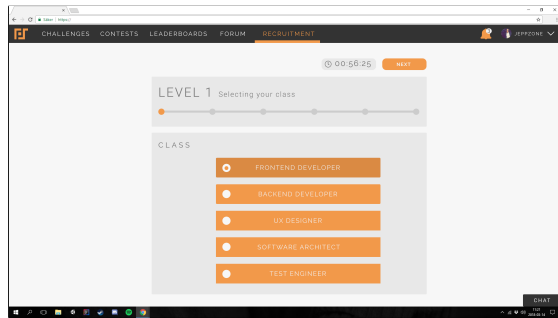


Figure 5.25: Hi-Fi mockup showing a selected class.

The player proceeds by choosing its top five abilities by dragging and dropping the suggestions into the slots, see Figure 5.26 and Figure 5.27.



Figure 5.26: Hi-Fi mockup showing the selection of abilities.

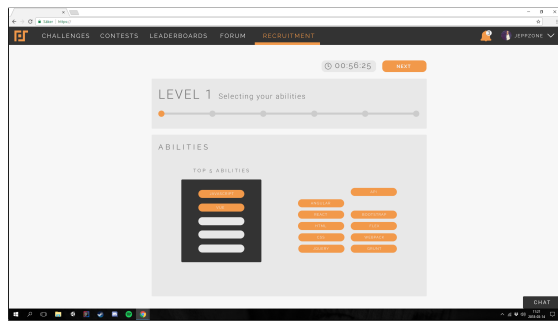


Figure 5.27: Hi-Fi mockup showing selected abilities.

Once again, Me-Fi designs have been merged together into a Hi-Fi mockup, see Figure 5.28. The only change here is the topics. "Server" and "Class" has been added in the overview area. The previous topic "Rank" has been changed to "Position" and earlier "Position" was renamed to "Placement". This change was made to highlight the difference between the two elements.

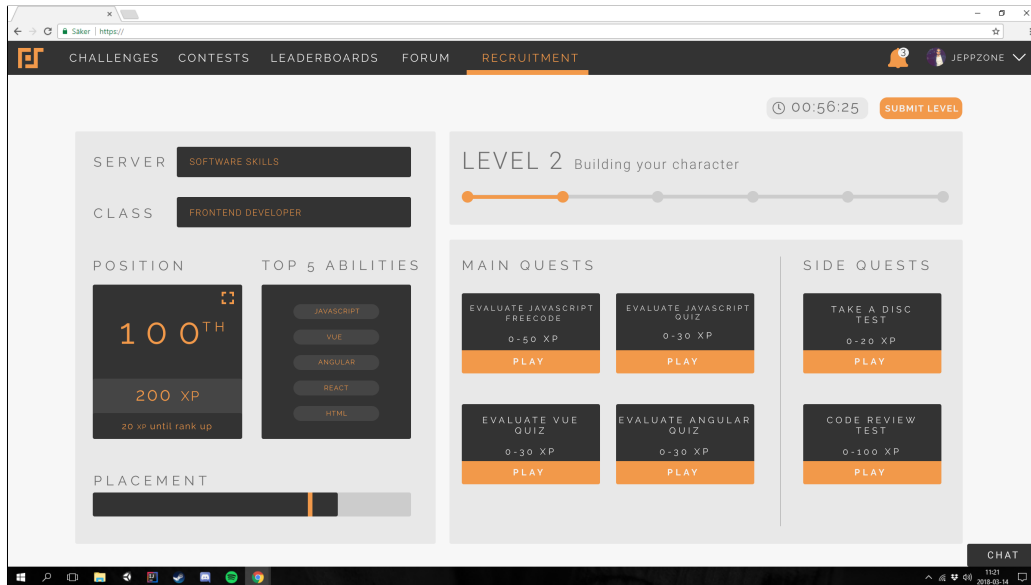


Figure 5.28: Hi-Fi mockup illustrating the building of a character.

As seen in Figure 5.29, it is possible to hover over the circles in the progress bar to get more information. The player has an overview, but can get details on demand. The time countdown also contains hover feedback, see Figure 5.30



Figure 5.29: Hi-Fi mockup showing progress bar and hover feedback.

If the player does not submit the level before the time countdown reaches zero, a penalty is yielded. This is demonstrated in Figure 5.31 and makes use of the *Penalties* pattern.

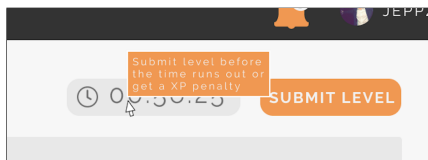


Figure 5.30: Hi-Fi mockup showing time countdown and hover feedback.

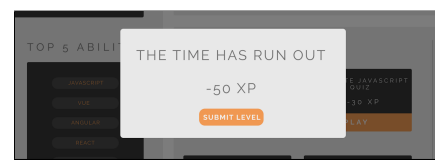


Figure 5.31: Hi-Fi mockup showing the time out penalty.

If the play button on a quest is clicked, the player will be taken to a quest view, see Figure 5.32. This view consists of a code editor and quest details, where the player should write code that solves the quest. The functionality of this quest is much like what the Future Skill platform provides (introduced and described in section 2.1).

The player can write code and test it to see its output until the time for the quest has run out. The quest time is visualized by the blue shrinking bar in between the "run code" and "submit" buttons. There is no way for the player to get away from the quest except by submitting it. This is a metaphor to the scenario when a candidate is tested live, hence the candidate can not cancel as easily as in the digital world.

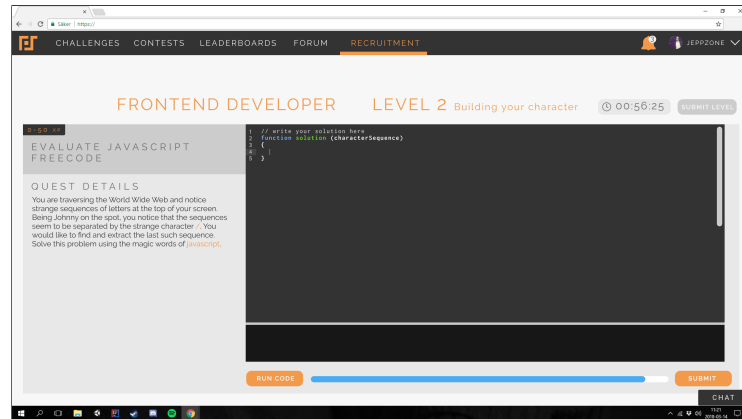


Figure 5.32: Hi-Fi mockup which illustrates playing a quest.

When a quest has been submitted, the player is taken back to the main view. The recently completed quest is updated in the "main quests" section with blink feedback, see Figure 5.33. The player receives an amount of XP depending on the correctness of the solution and the quest is marked as "completed". The position, total XP and rank up hints are updated similarly as seen in Figure 5.34.

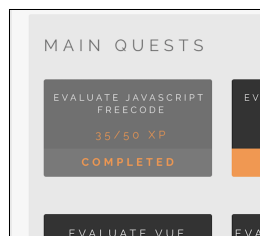


Figure 5.33: Hi-Fi mockup showing a quest update and blink feedback.

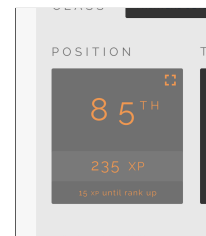


Figure 5.34: Hi-Fi mockup showing position and XP updates and blink feedback.

Another type of a quest is seen in Figure 5.35 (also derived from the Future Skill platform) and is formulated as a quiz, where the player reads questions or statements and answer them respectively by choosing from the list of options. This quest implements the *GDP Quizzes*.

Figure 5.36 also illustrates a quiz-like quest. This is a side quest that contains a personality test.

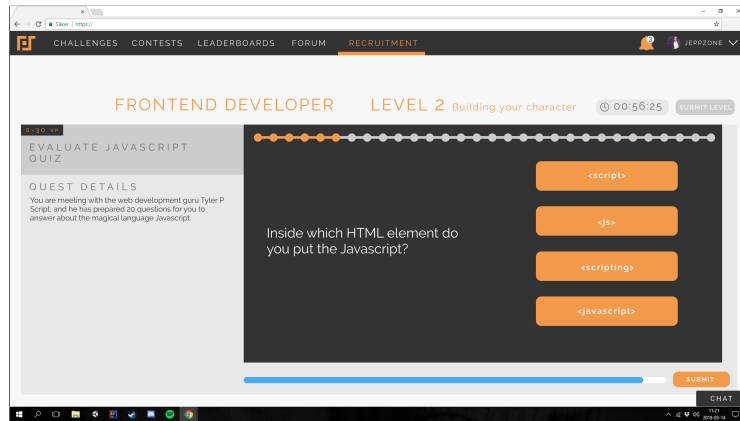


Figure 5.35: Hi-Fi mockup illustrating an example of a quiz quest.

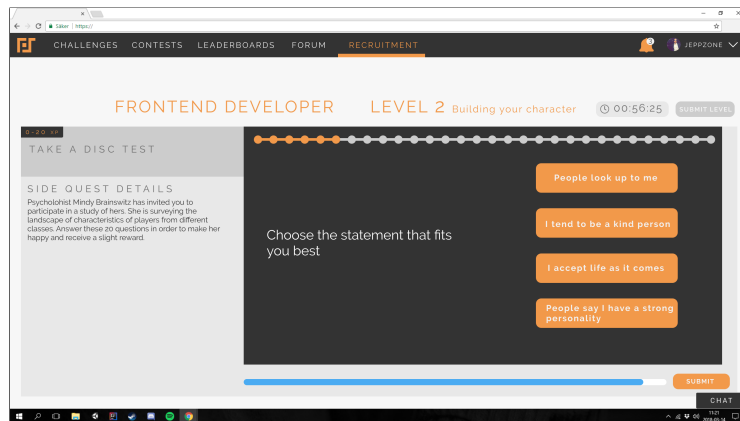


Figure 5.36: Hi-Fi mockup illustrating an example of a side quest.

Figure 5.37 visualizes how it looks when the player has completed two quests. The player can continue gaining XP by doing more of them, but the same quests can not be redone.

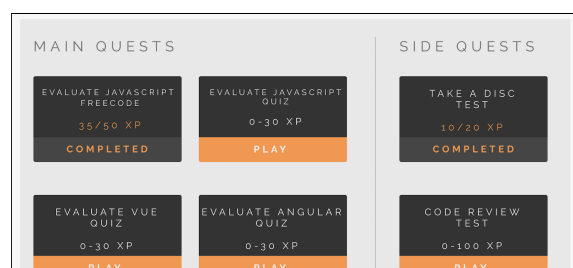


Figure 5.37: Hi-Fi mockup which shows that two quests are completed.

Under the topic "position", as seen in Figure 5.37, a small maximize icon can be found in the top right corner. If the player clicks here, a top list will open up, as seen in Figure 5.38. The top list shows a couple of other players whose positions are just above or under the player's. The list also always shows the top three players. The top list makes use of the GDP *High Score Lists*. The player can get back to the

previous view by clicking the minimize icon in the top right corner of the dark grey area.

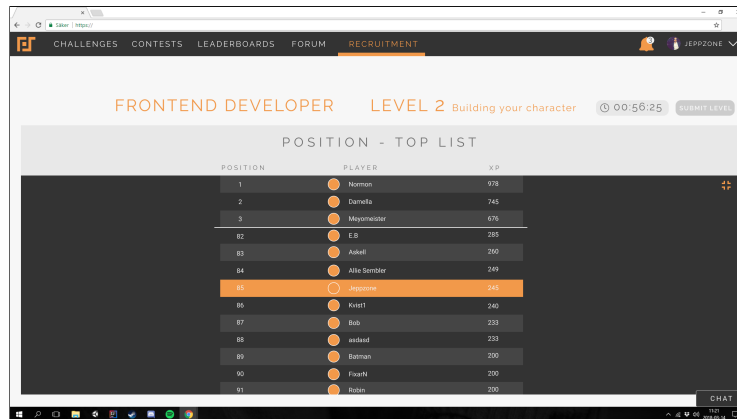


Figure 5.38: Hi-Fi mockup with an example of top list positions.

If the player continues and succeeds to level three, the view in Figure 5.39 is shown. The functionality is the same as in the Me-Fi Section 5.3.5.3, but if the player presses play here the system will open the view in Figure 5.40. The idea is the same as earlier, the player should face a bigger programming challenge than in the earlier, smaller quests.

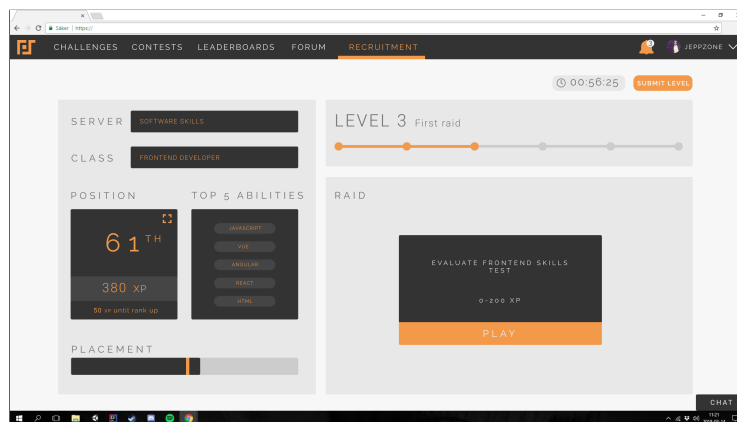


Figure 5.39: Hi-Fi mockup which shows the view of level three.

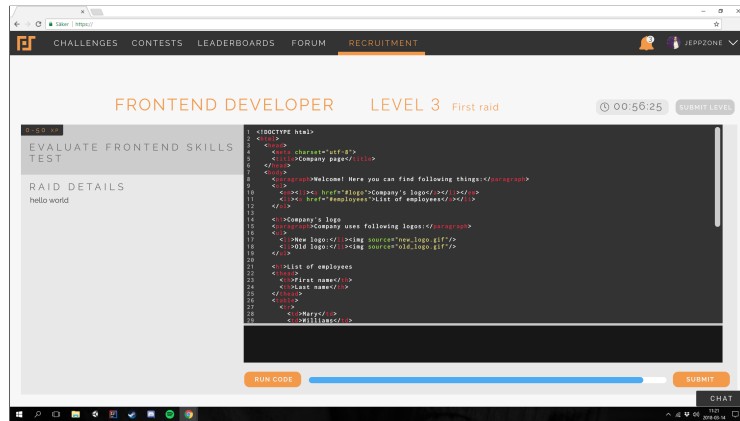


Figure 5.40: Hi-Fi mockup which gives an example of a first raid.

The views in Figure 5.41 and Figure 5.42 adds in level four and five respectively into the Hi-Fi design.

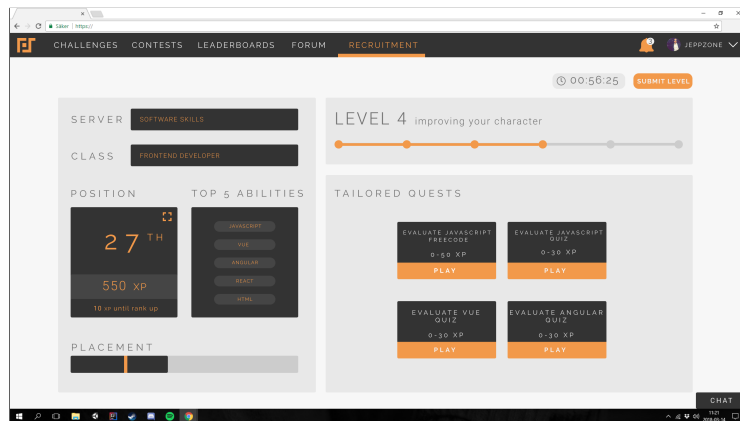


Figure 5.41: Hi-Fi mockup which shows the view of level four.

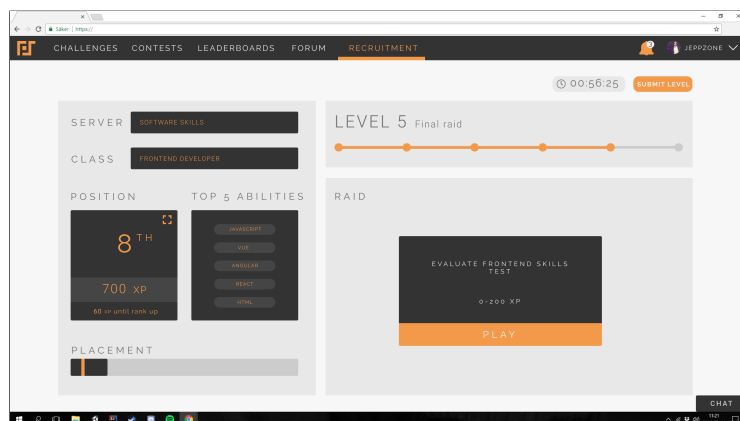


Figure 5.42: Hi-Fi mockup, which shows the view of level five.

The winning state and the losing state of the game are shown in Figure 5.43 Figure 5.44 respectively. If the player wins, an opportunity of getting hired is given.

If the game is lost, the player can choose between joining another server or playing another class. On the game over view, the "placement" visualization shows that the vertical orange line has been pushed out of the dark grey area, hence indicating how close the player was to stay in the game.

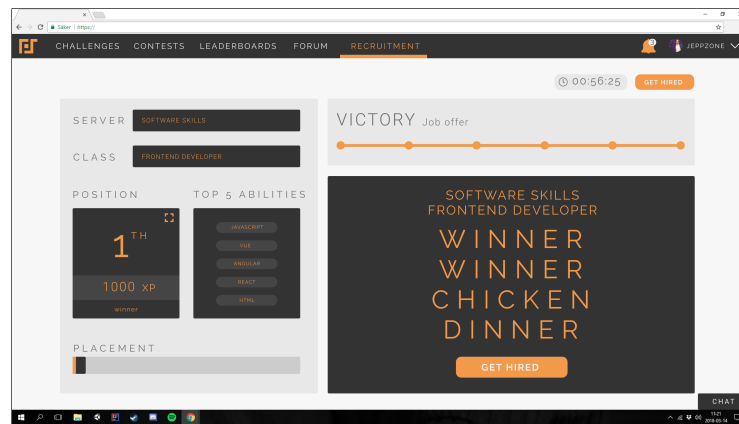


Figure 5.43: Hi-Fi mockup which gives an example of the winning state.

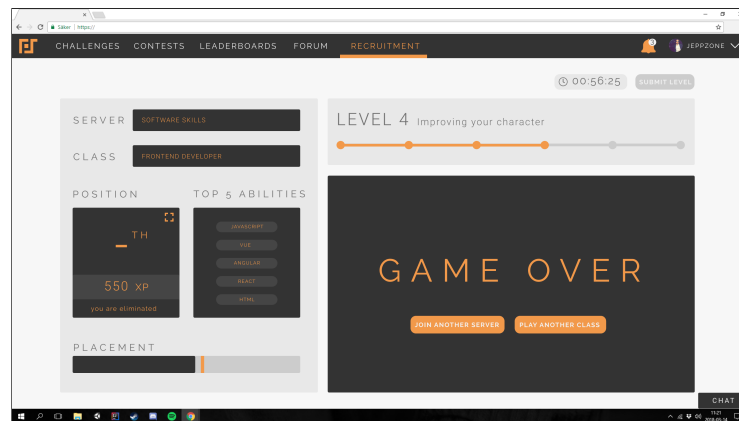


Figure 5.44: Hi-Fi mockup, which gives an example of the losing state.

In this section, the game as a whole has been walked through, but there is one aspect that exists outside of the scope of the game. That is the profile page, see Figure 5.45. The player gets here by clicking on their nickname in the top right corner. The profile page contains statistics that are summarized over all played games.

The statistics shown are:

Title	Description
• Wins	The total number of won games.
• Top 10	The number of top 10 (positioned 1-10) placements.
• Beaten	Number of enemies that have been eliminated before the player.
• Total XP	Total gained XP and weekly based staple diagram visualization on how much XP was earned the previous five weeks.
• Ability reliability	Visualization on how well the player has performed on each ability.
• Achievements	List of received achievements.
• Lines of code	The total lines of code that has been written in the quests.

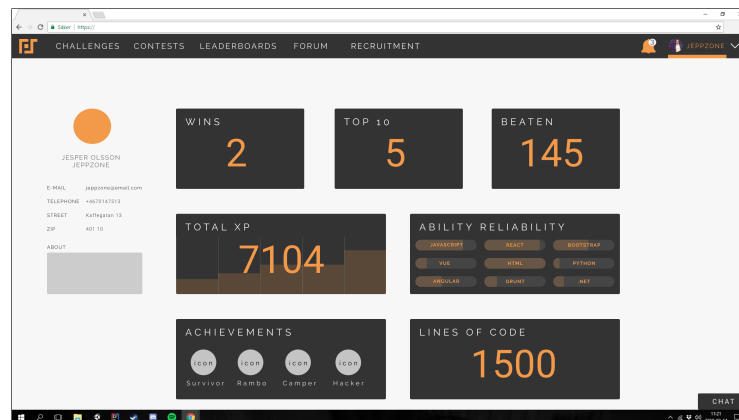


Figure 5.45

5.4 Iteration 3: Adding a theme to the design

The design created in the second iteration contained most of the necessary building blocks for a complete design, but design decisions such as element naming, look and feel and story were merely based on intuition. This is why a new theme for the whole design was developed. Most of the gameplay design patterns that were described and implemented during the second iteration (see Section 5.3.5 still exist in the game, and mentions of specific gameplay design patterns in this section will only be made if they were added, altered or removed in this redesign.

5.4.1 Ideation for themes

As a start for the theme ideation phase, three different core topics were chosen. These three were "Agent", "Soldier" and "Hacker". All of these puts the role of the player in focus; is the player an agent, a soldier or a hacker? The three alternatives all fit together with the features designed in iteration two. A brainstorm session was

held, see Figure 5.46, and the theme "Agent" received the most amount of votes, and hence was selected for further development.

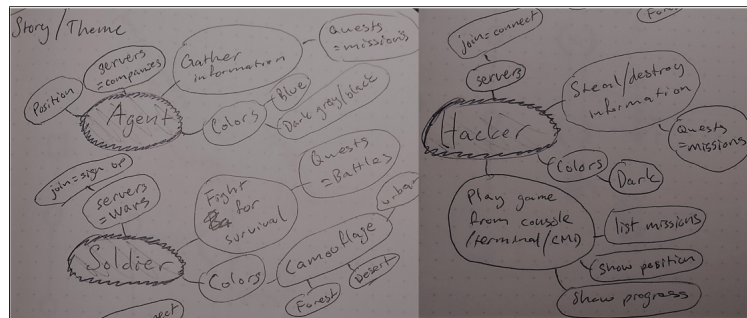


Figure 5.46: Brainstorm notes when trying to find alternative themes to introduce.

To further develop and define the agent theme, a holistic concept map was made (see Figure 5.47) in order to connect the naming, the look and feel and the overarching story to the new theme. As seen in the concept map, all of the small changes resulted in an even more coherent design.

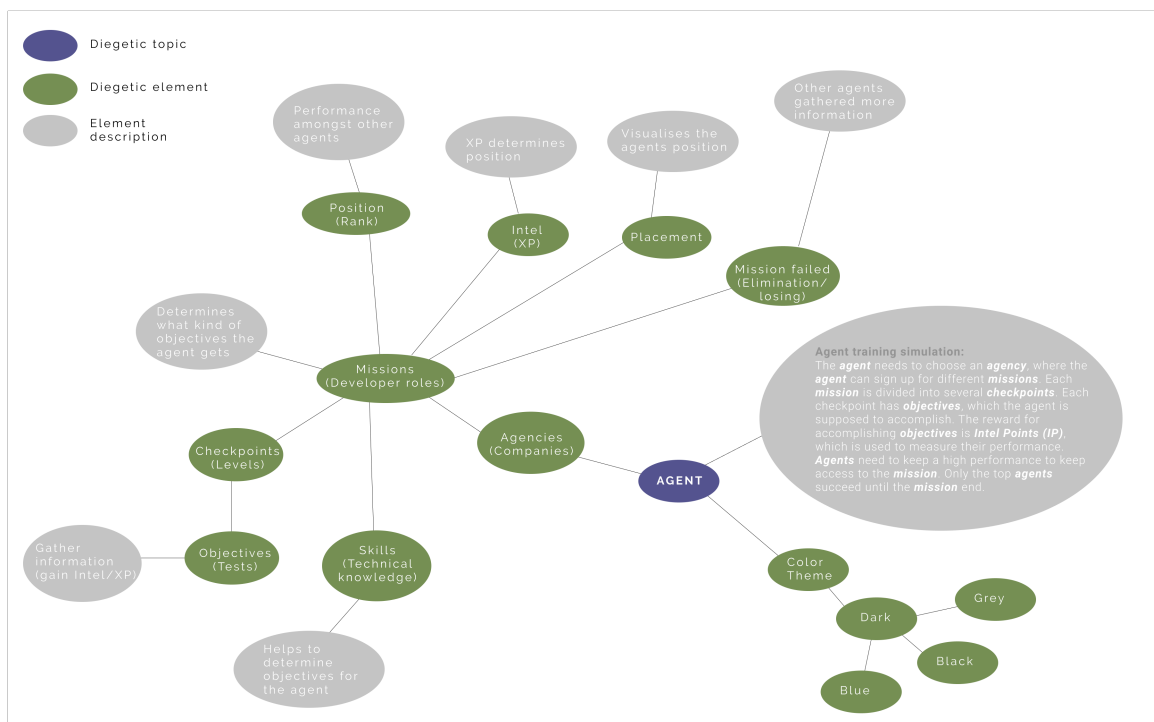


Figure 5.47: A concept map of the new theme.

In association with the new theme, a flow map was created, see Figure 5.48, to further define the diegetic elements and how they are correlated in the game. The flow map also indicates what game states exist. A substantial change was made to what happens if a player manages to complete the goals of the game and win it. In the concept developed in iteration two, the player received a job offer as a result

of winning the game. After discussions and reasoning, this was deemed unrealistic, due to a company most likely not being willing to hire a person they have never met or spoken to. Instead, in this redesign, a player who manages to accomplish the goals of the game will receive a certificate of their accomplishment, which could work as a gateway to a job at the company in question, rather than an actual job offer.

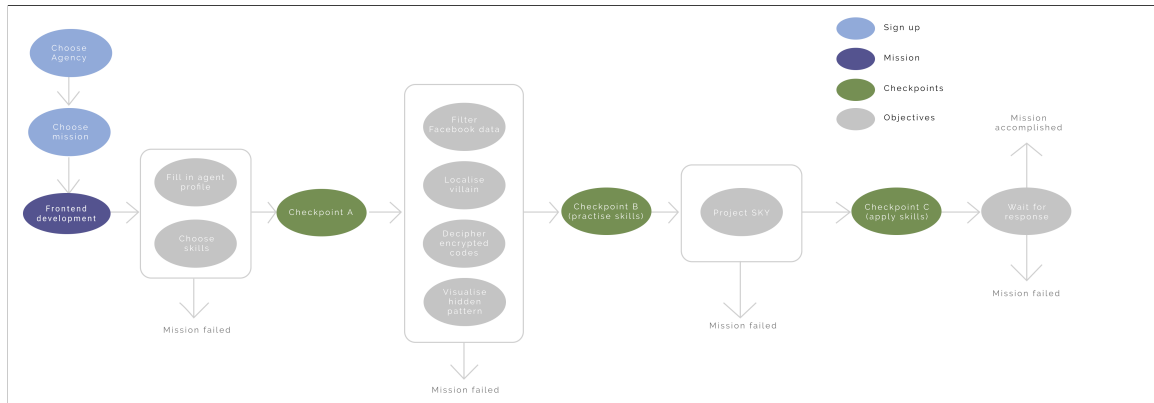


Figure 5.48: A flow map that works as an overview of the content in the new theme

5.4.2 Implementing a theme

Subsequent to creating a concept map and a flow map to act as the foundation of the new design, the implementation phase was initiated.

5.4.2.1 Lo-Fi Sketching

Even if Hi-Fi mockups had been made during the second iteration, it was of importance to fall back to Lo-Fi prototypes, to step away from the polished design and faster bring up new ideas and integrate them together with the new theme aspects.

A new entrance to the game was developed to give the game story more attention. If the game is opened for the first time, the player sees a thematic introduction text, as seen in Figure 5.49. Figure 5.50 shows the screen that would come up if the "start" button is pressed, which contains the creation of an account for the website and the game. The player can create an agent account by entering a codename (unique user name), email and password. Figure 5.51 shows the view that appears if the player already has an account. It was decided that the player needs to have a codename with three digits in the end to further solidify the agent theme, for example "Jeppzone001".

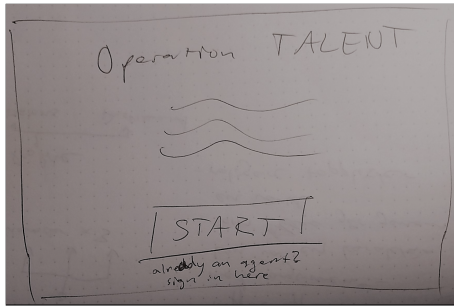


Figure 5.49: Lo-Fi sketch illustrating the game introduction.

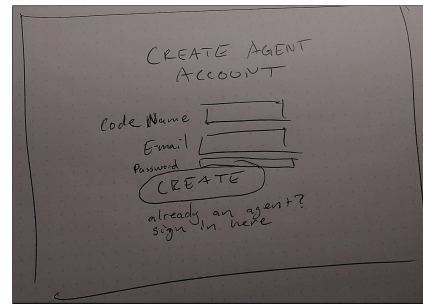


Figure 5.50: Lo-Fi sketch illustrating the creation of an agent account.

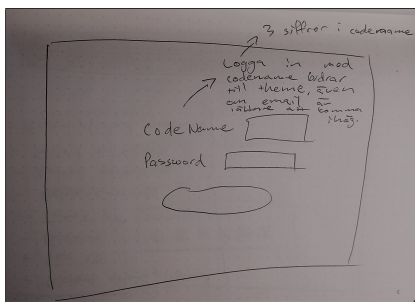


Figure 5.51: Lo-Fi sketch illustrating log in page for agents.

From here on, the player will be referred to as "the agent". Instead of conceptualizing a role in a company as a class, the agent now selects a mission. To add more thematic elements to this choice, missions were visualized as folders. These folders can in turn be opened to bring up a document that contains the details of the mission. Figure 5.52 illustrates this described aspect.

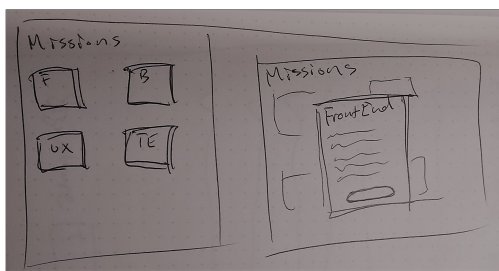


Figure 5.52: Lo-Fi sketch illustrating missions overview and mission document.

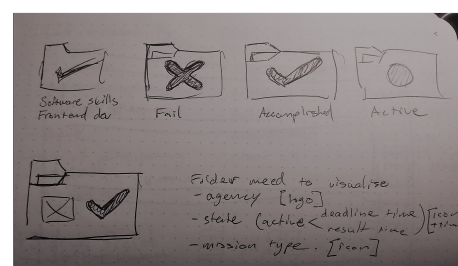


Figure 5.53: Lo-Fi sketch illustrating alternative designs on visualizing mission states.

Further development of the missions lead to the creation of a mission tab on the website. The game needed a place to put started, accomplished and failed missions in. Figure 5.53 shows the work around how to indicate the state of each mission.

Figure 5.54 and Figure 5.55 together demonstrate one idea of an advanced search and filter list, containing a lot of data about the missions.

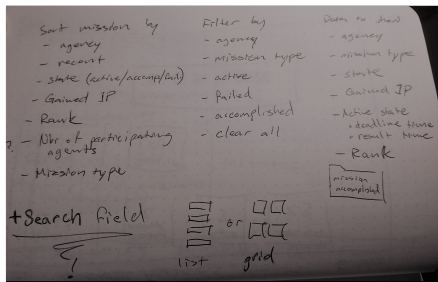


Figure 5.54: Lo-Fi sketch illustrating possible data to show about missions.

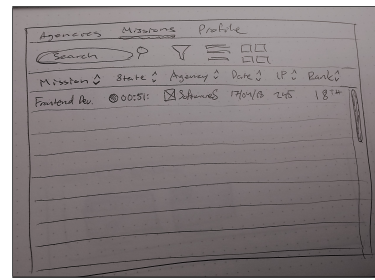


Figure 5.55: Lo-Fi sketch showing an idea of how to list missions.

The advanced list was considered to be too complex, since most agents are not supposed to join that many missions (the goal is to get hired by completing one of them). A simpler alternative was created, as seen in Figure 5.56, where the folder icons from Figure 5.53 would be put into the three different categories: "Active", "Accomplished" and "Failed". An active mission is a mission where the agent has started the mission but have not yet passed or failed it.

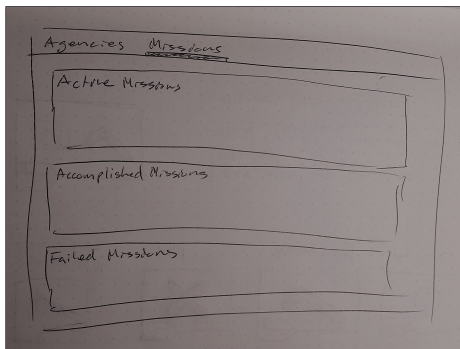


Figure 5.56: Lo-Fi sketch showing a simplified mission list.

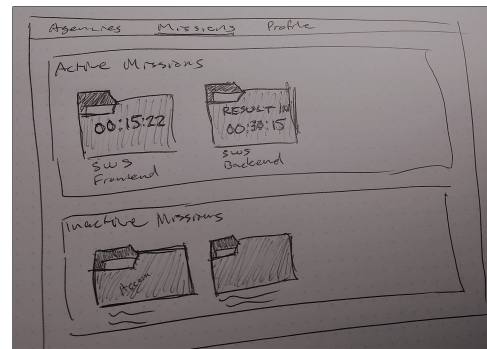


Figure 5.57: Lo-Fi sketch showing the end result of the Lo-Fi missions' tab design iterations.

The Lo-Fi missions' tab design resulted in the view illustrated in Figure 5.57. Accomplished and failed missions were merged into "Inactive missions" to focus more on the "Active missions" above. The active missions are displayed with a time countdown (the same as in the mission) printed on the folder. They also have the agency and mission written below. The rightmost folder in the active missions section shows a special case "Result in", which is when the agent waits on the result after their very last objective (the company needs to evaluate their sent in project and decide if the mission is passed or failed).

A remake of the progress bar was considered a need. Figure 5.58 demonstrates this redesign. The new progress bar realizes the idea of agents having to reach checkpoints. The start of the progress bar does not contain a circle anymore since the agent cannot be on a checkpoint immediately after starting the game. There are three checkpoints: checkpoint A is reached after the profile creation and skills selection, checkpoint B is reached after completing the objectives (if receiving enough

IP) and checkpoint C is reached after completing the project (if once again receiving enough IP). If all checkpoints have been reached, the agents wait for the end result.

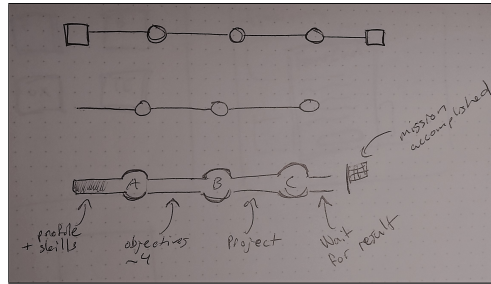


Figure 5.58: Lo-Fi sketch illustrating the remake of the progress bar.

5.4.2.2 Hi-Fi Mockups

The new and reworked theme was here brought into high-fidelity mockups. Figure 5.59 shows the introduction screen, starting with the new, more thematic game name "Operation T.A.L.E.N.T.". T.A.L.E.N.T is an acronym standing for "Operation Training Agents to Localize Enemies Near Town", and the word talent is also a reference to recruiting in general, where talent often is a keyword. The introduction starts with three questions that try to convey the message that this game can lead to something new and exciting. The questions are followed by a short description about the game.

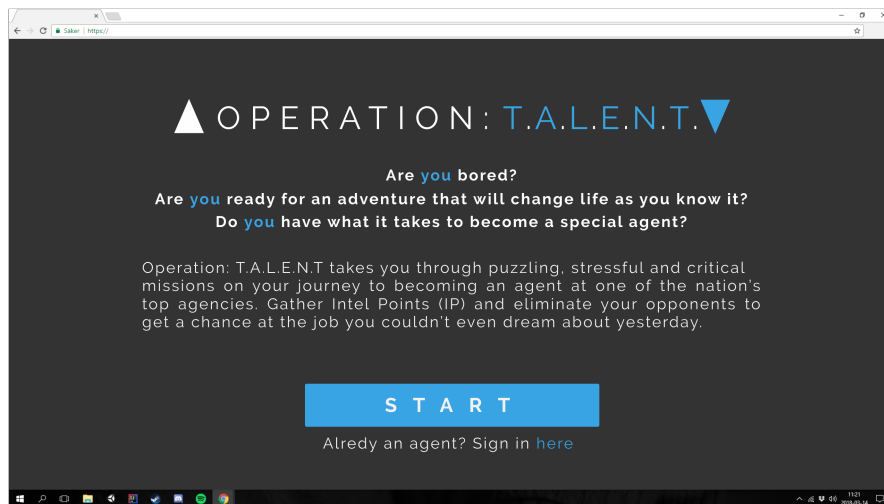


Figure 5.59: Hi-Fi mockup showing the introduction screen.

The creation of an account and sign, seen in Figure 5.60 and Figure 5.61, works as described in the section 5.4.2.1.

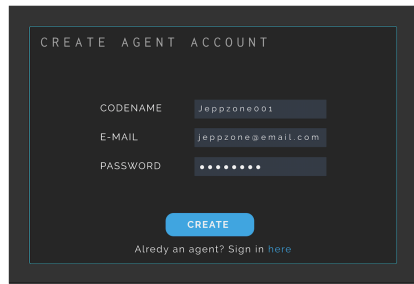


Figure 5.60: Hi-Fi mockup illustrating an example of creating an agent account.

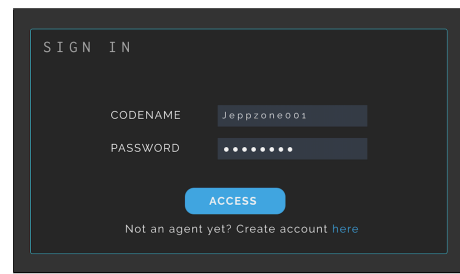


Figure 5.61: Hi-Fi mockup illustrating an example of signing in to an agent account.

The updated "choosing company screen" is seen in Figure 5.62. The "Servers" naming (from Section 5.3.5.4) have been changed to "Agencies" and instead of "Join" there is now "Select". The agent can select an agency to browse through their missions.

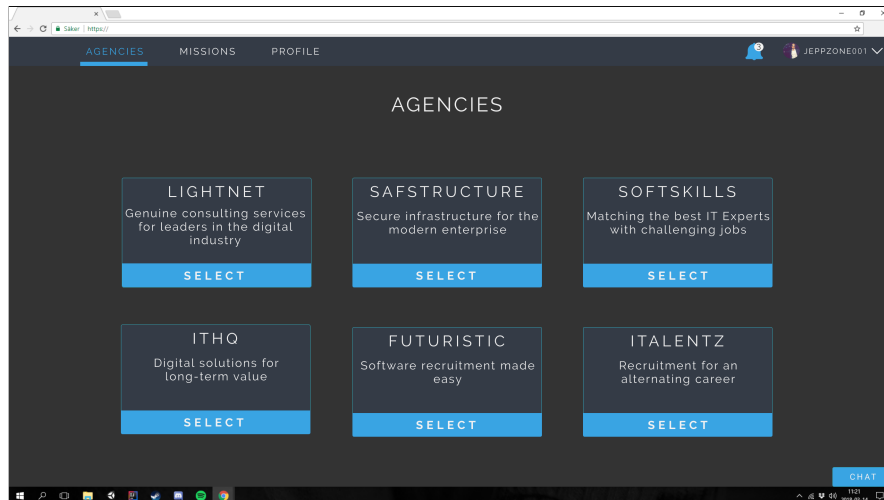


Figure 5.62: Hi-Fi mockup displaying selectable agencies.

The overview of missions is seen in Figure 5.63 and if the agent opens a folder, the view in Figure 5.64 is displayed. Here, the agent gets to read the details about a mission contract and if the agent wants to carry out this mission, a signature (by using keyboard text input) is needed at the bottom of the contract.

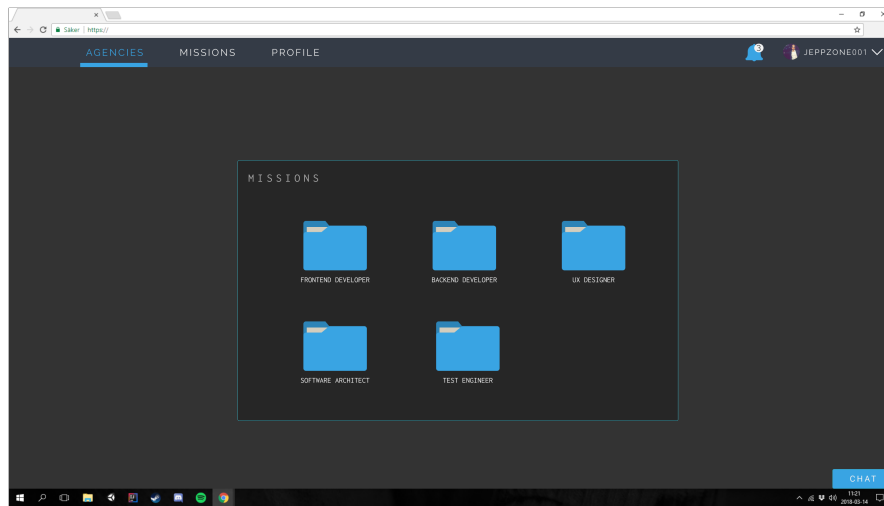


Figure 5.63: Hi-Fi mockup illustrating the new missions' overview.

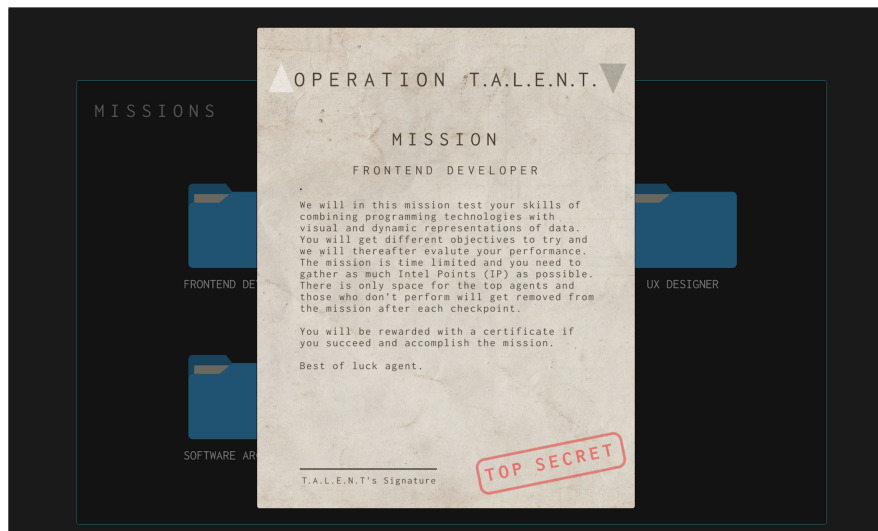


Figure 5.64: Hi-Fi mockup illustrating the mission contract.

Figure 5.65 shows the screen where the agent is supposed to fill in an agent profile. There are changes compared to Section 5.3.5.4, such as the updated progress bar (that works like described in section 5.4.2.1) and "Character details" is changed to "Agent profile". In the agent profile form, "nick" is changed to "codename".

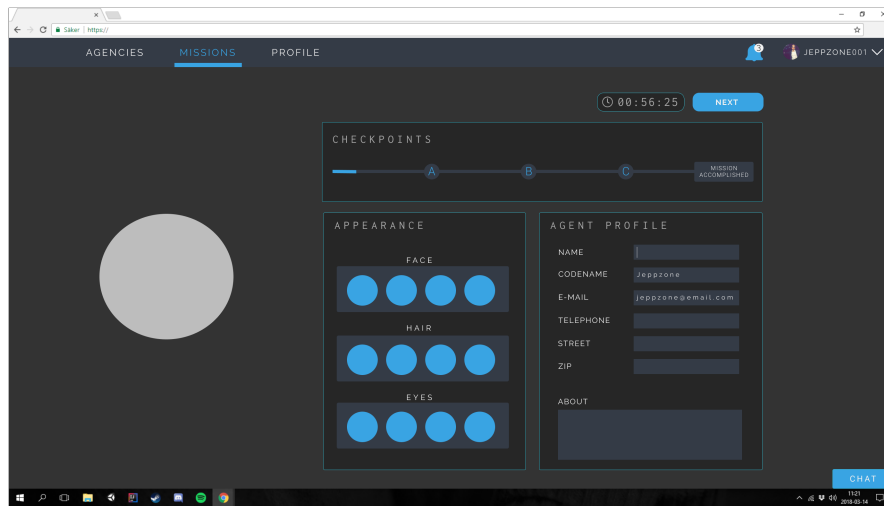


Figure 5.65: Hi-Fi mockup displaying how to fill in the agent profile.

Choosing the "top five abilities" as in Section 5.3.5.4 has been changed to choosing the "top five skills" as seen in Figure 5.66 and Figure 5.67 shows the updated views. Notice that the progress bar also gets filled in the more skills the agent picks into the fields, as it moves the agent further towards the checkpoint.

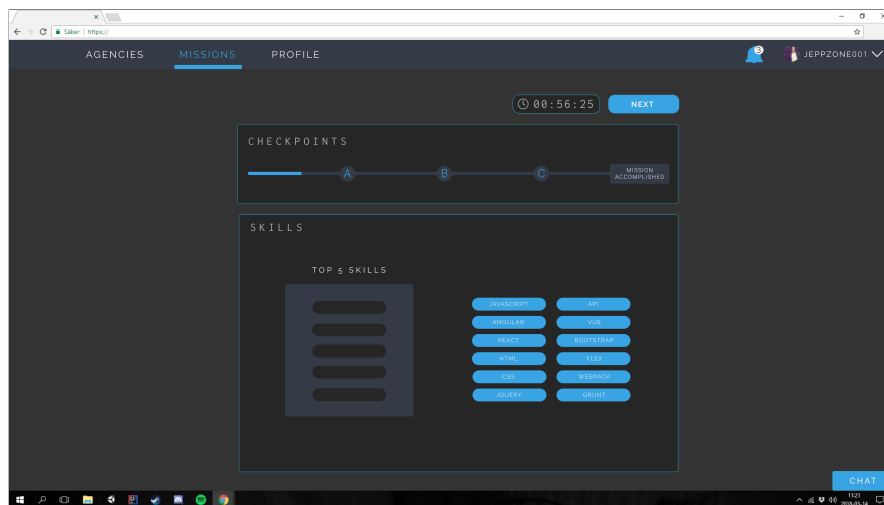


Figure 5.66: Hi-Fi mockup displaying selection of top five skills.

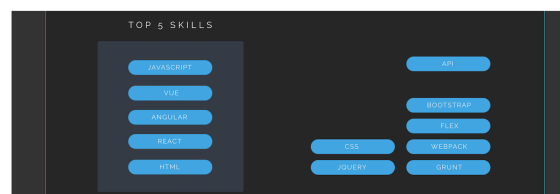


Figure 5.67: Hi-Fi mockup illustrating the top five skills having been selected.

There are both thematic and usability updates with the main view seen in Figure 5.68. The thematic updates are that "Quests" are now named "Objectives" and

"Experience Points (XP)" are named "Intel Points (IP)". The agent is on a mission with several preceding objectives in between checkpoints. The agent will receive an amount of IP depending on their performance on the objectives. One usability update is that the light blue colour is only applied to interactive elements such as buttons. Another usability update is that the placement visualization has been reversed from the one in Section ??, now the small light blue vertical line approaches to right (instead of left) when getting a better position. Also a small text with a percentage is indicating how many other agents that have been eliminated. Figure 5.69 and Figure 5.70 shows the updated hover feedback.

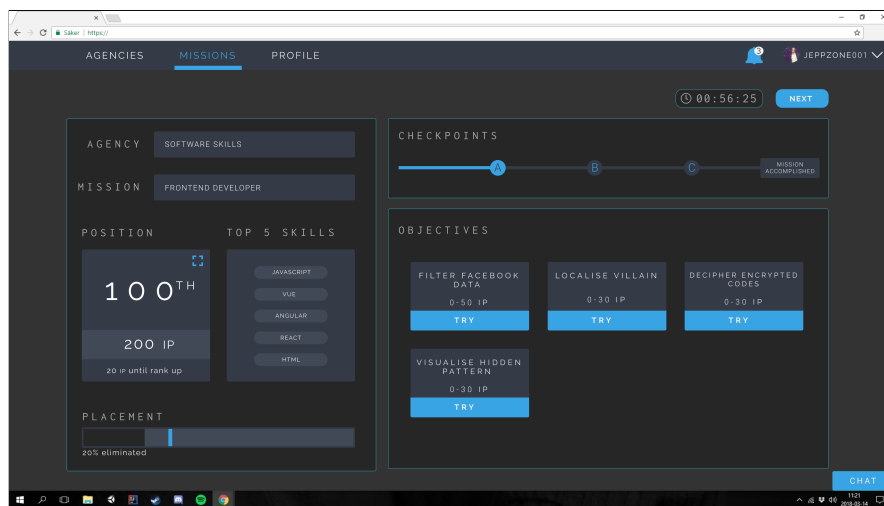


Figure 5.68: Hi-Fi mockup displaying the main view after reaching checkpoint A.

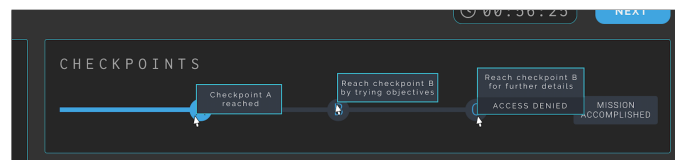


Figure 5.69: Hi-Fi mockup illustrating hover feedback on the progress bar's checkpoints.

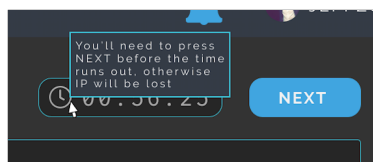


Figure 5.70: Hi-Fi mockup illustrating hover time feedback.

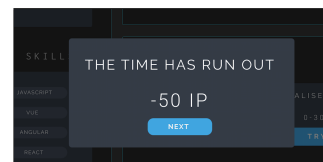


Figure 5.71: Hi-Fi mockup showing an example of a time out penalty.

Figure 5.71 shows the thematic update on the time penalty.

The look and feel of a mission objective in the code editor environment now looks as in Figure 5.72. The title has been changed to the thematic title brought up in

the flow map, earlier seen in Figure 5.48. The functionality is the same as in Section 5.3.5.4.

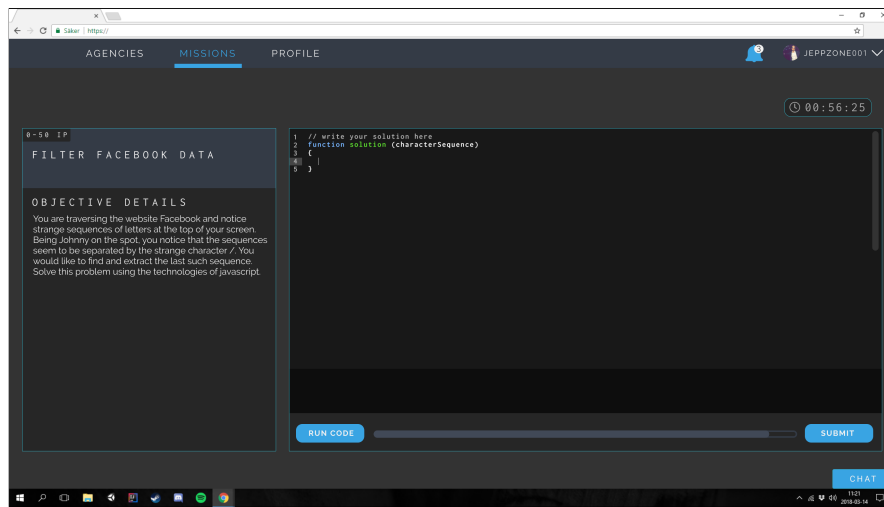


Figure 5.72: Hi-Fi mockup showing an example of an objective to be solved in the code editor.

Objectives with the quiz format have been thematically updated in form of naming, objective details and colours as shown in Figure 5.73 and 5.74.

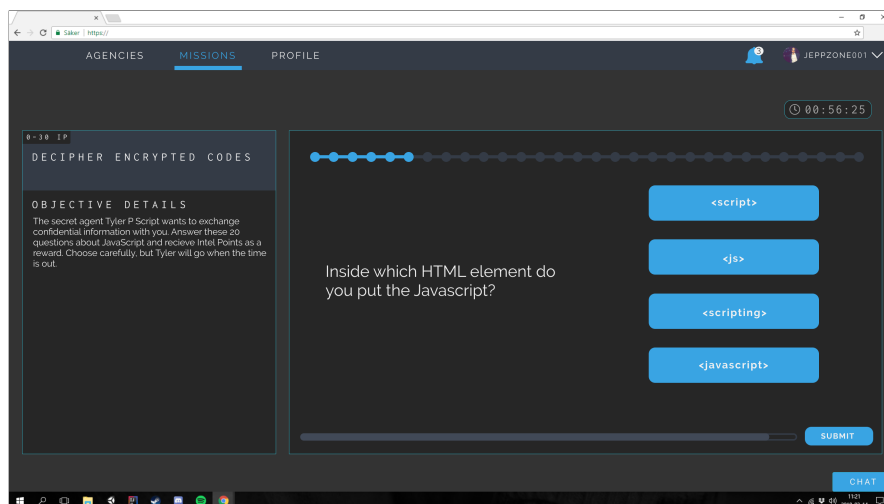


Figure 5.73: Hi-Fi mockup, agent themed quiz

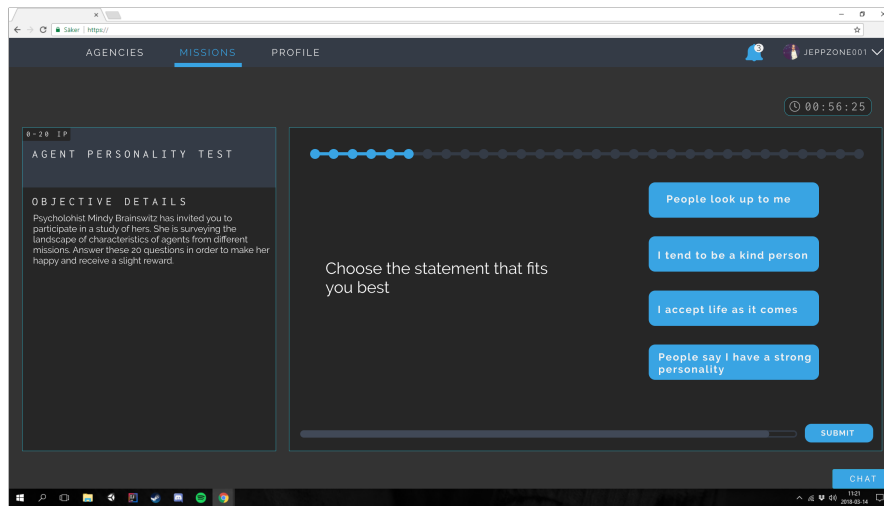


Figure 5.74: Hi-Fi mockup, agent themed personality test

Completed objectives are now shown with an amount of received IP and a text saying "completed", as seen in Figure 5.75. The more IP the agent gathers, the more the progression bar gets filled up by the light blue colour, approaching the checkpoint.

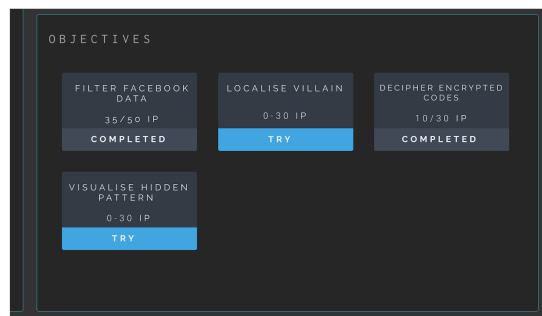
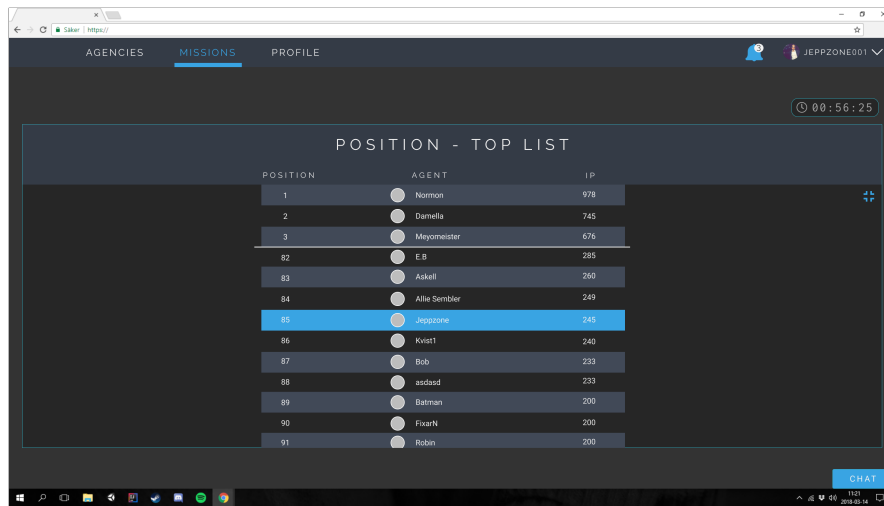


Figure 5.75: Hi-Fi mockup, showing completed objectives and received IP.

Figure 5.76 shows the updated top list view. The only changes except from colouring are the naming of the titles, from "player" to "agent" and "XP" to "IP".



POSITION	AGENT	IP
1	Norman	978
2	Daniella	745
3	Maymeister	676
82	E.B	285
83	Askel	260
84	Allie Sembler	249
85	Jeppone	243
86	Kvist1	240
87	Bob	233
88	esdard	233
89	Batman	200
90	FixarH	200
91	Robin	200

Figure 5.76: Hi-Fi mockup showing the updated top list.

One big change from the first Hi-Fi concept in Section 5.3.5.4 is that the agent-themed concept has fewer stages (checkpoints) in the progress bar. The amount of stages has been reduced from five to three. It was considered unnecessary to have more than three checkpoints because the testing of a candidate should not be too long and time consuming in a recruitment process. The agents are faced with their last objective, a bigger project called "Project Sky", already after reaching checkpoint B, as seen in Figure 5.77.

The project is still of the same characteristics as in the previous Hi-Fi concept in Section 5.3.5.4. Figure 5.78 gives an example of how a bigger project could look in the agent-themed code editor.

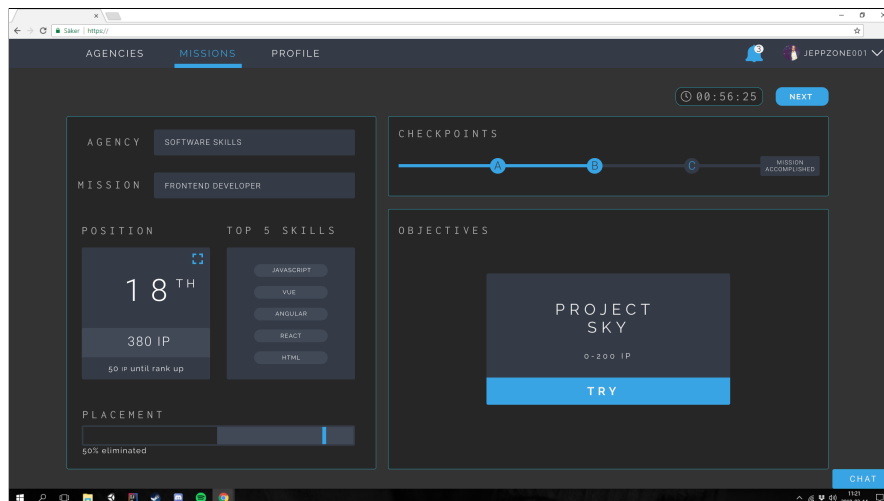


Figure 5.77: Hi-Fi mockup with the last objective called Project Sky.

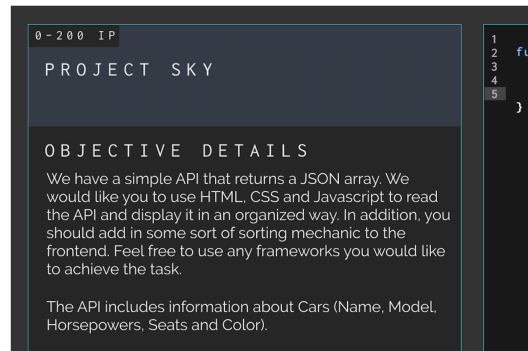


Figure 5.78: Hi-Fi mockup with an example of a bigger project.

If the agent tries the objective project sky and submits it, the system will show a view as in Figure 5.79. Since this is the last objective of the mission, the agent is presented with a countdown to when the results will arrive. In the meantime, the agent can choose to click the button "mission overview", which will display the home screen of the missions' tab (same as clicking at the tab "missions" in the tab bar).

Figure 5.80 demonstrates a scenario of using the missions' tab. If the agent opens one of these missions, the system will the agent back to the last accessed view on that mission. The inactive missions have a green or red stamp with "mission failed" or "mission accomplished" on them.

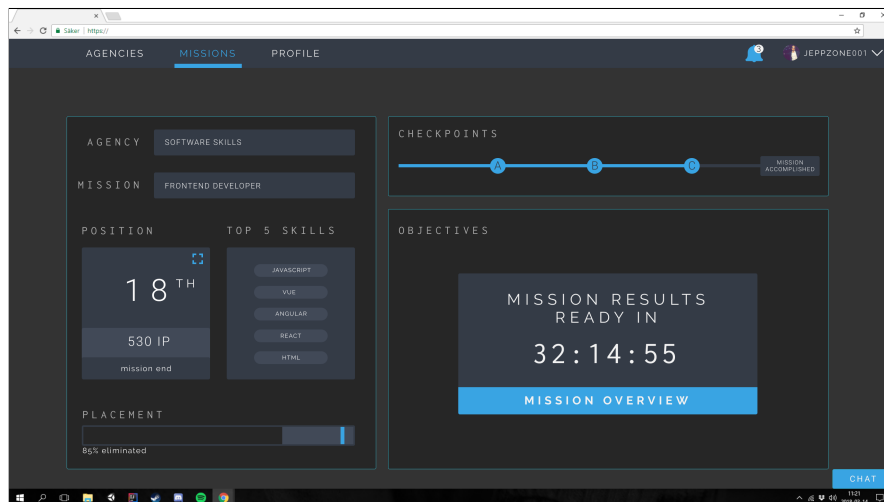


Figure 5.79: Hi-Fi mockup where the mission has ended and the agent is awaiting results.

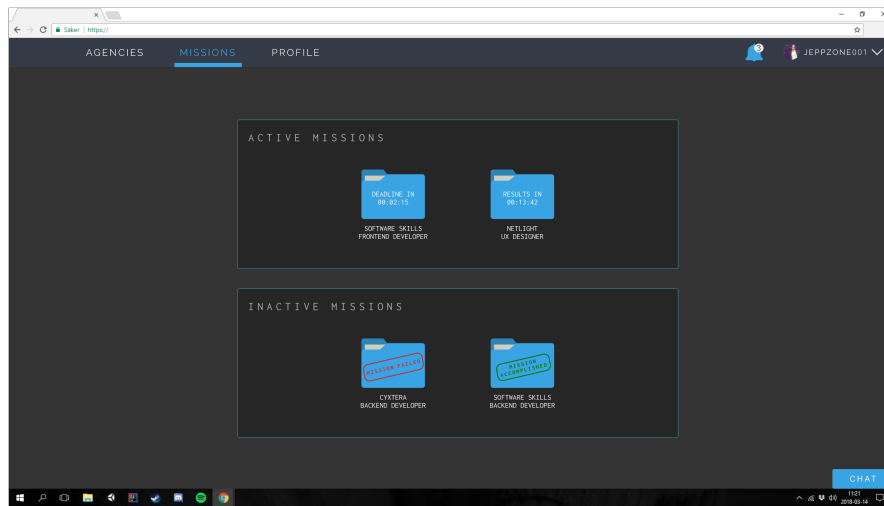


Figure 5.80: Hi-Fi mockup with an example of missions in the missions' tab.

There are two scenarios when the results for the mission arrive. Scenario one is shown in Figure 5.81 and that is that the agent accomplished the mission. The agent can choose to press the "receive certificate" button to open the view in Figure 5.82. The certificate can be used as a proof of knowledge on upcoming, real-life interviews.

The second scenario is that the agent failed the mission, as displayed in Figure 5.83. The agent has failed and can not choose to try the same mission again, but trying another mission is recommended by the button "try another mission".

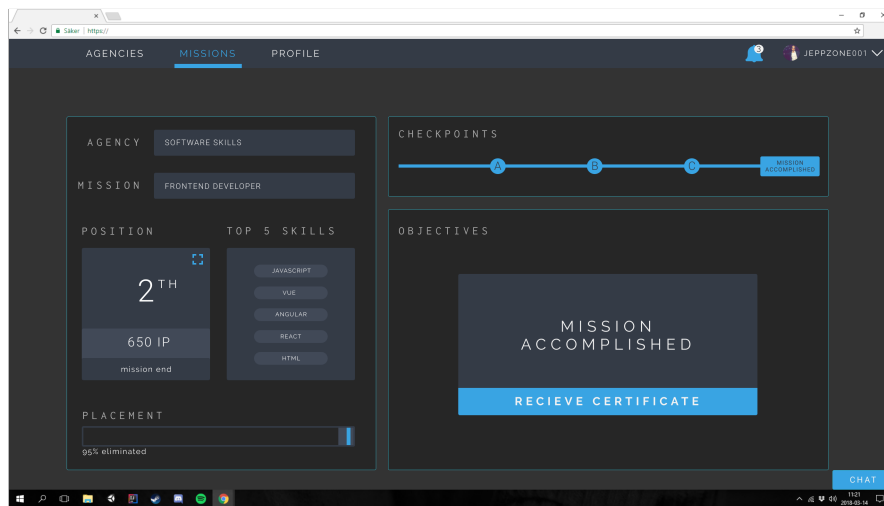


Figure 5.81: Hi-Fi mockup showing the scenario when a mission is accomplished.

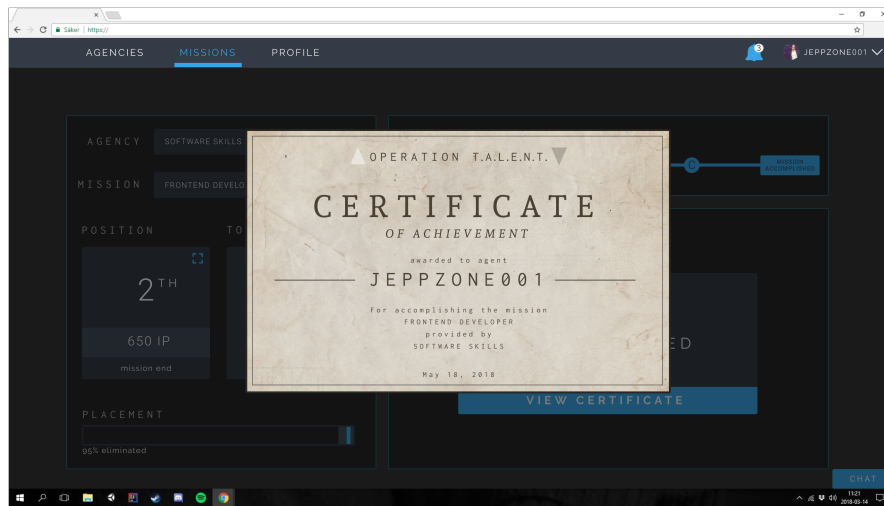


Figure 5.82: Hi-Fi mockup showing when the agent receives a certificate.

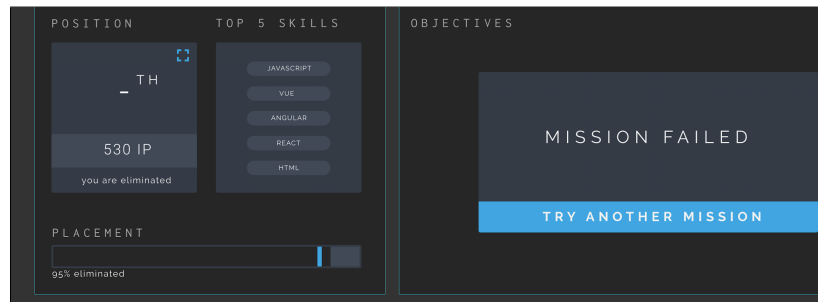


Figure 5.83: Hi-Fi mockup showing an example of a failed mission.

Figure 5.84 shows how the profile tab could look like after the agent has signed up for a couple of missions. It works in the same way as described in Section 5.3.5.4, but naming and style has been updated in accordance with the agent theme.

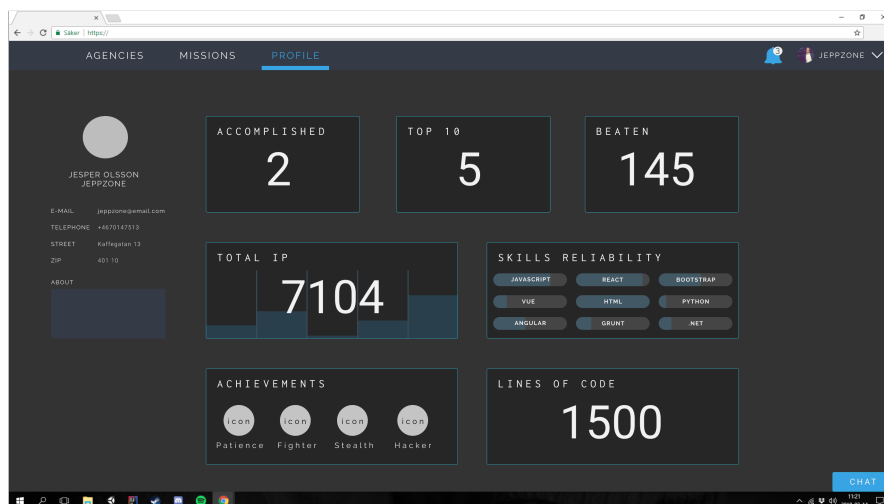


Figure 5.84: Hi-Fi mockup illustrating an example of the agent-themed profile page.

This image concludes the presentation of the recruitment tool, which in future sections is referred to as "Operation Talent". The version of Operation Talent displayed in this section is the final version of the tool. This tool was developed to serve as a discussion point when talking about the application and utilization of gameplay design patterns on the recruiting process.

6

Results & Findings

In this section, the results and general findings of the thesis are presented. As most of the work done in this thesis is presented in Chapter 5 *Design Process*, a lot of partial results resides there. To avoid redundant information in this section, in cases where the partial results are already presented earlier in the thesis, references to the specific sections containing the results are given.

As stated in Section 1.1, the goal of this thesis was to investigate the following research question

What impact do the application of gameplay design patterns have in the development of a recruiting process?

by addressing the following subquestions:

- What gameplay design patterns exist in a general recruiting process?
- What additional gameplay design patterns would have an impact on users' engagement?

Furthermore, as stated in Section 1.4, the intention was that the result of this thesis should be a three-part result. The first part of the result, being a list of patterns identified in a general recruiting process, covers the first subquestion. This question is answered partly in Section 5.3.2, which contains a list of the most relevant identified gameplay design patterns. For the full list of identified gameplay design patterns, see Appendix C.

Regarding the second subquestion, a hypothesis of which gameplay design patterns could impact user engagement is given in Section 5.3.3. In order to get a concrete example of what combining and implementing all these patterns together could look like, a concept design was developed as a part of the design process. For the final concept design, see Section 5.4. The concept design is intended to work as a discussion catalyst and as a more concrete way of talking about gameplay design patterns and their impact on different aspects. It is difficult to pinpoint what affect specific gameplay design patterns have on aspects such as user engagement, but taking a more holistic perspective and looking at gameplay design patterns as a framework, it is possible to make arguments around it's impact on the recruiting process.

6.1 Impact on the Recruiting Process

Section 1.2 outlines a hypothesis trying to answer the research question. This hypothesis briefly outlined the ways in which it was believed that gameplay design patterns would impact the recruiting process. It was believed that utilizing gameplay design patterns would attract talented candidates with an interest in games, and that they would be able to show qualities such as persistent efforts in this tool.

Looking from a holistic perspective and the recruiting process as a whole, one could say that Operation Talent has impacted the process of recruiting by reversing the order of some of the stages in it. Generally, as explained in Section 5.3.1, the recruiting process begins with a candidate going to an interview and if successful, that candidate moves on to a competence test. Operation Talent reverses this order by having candidates go through competence tests before ever reaching the interview. This should lessen the workload of recruiters, since Operation Talent will attract candidates broadly, but only let through a few of them, with little effort needed from recruiters. It works both as a marketing tool for attracting candidates with its gamified and thematic nature, as well as a funnel for sieving candidates based on competence.

Looking from the candidates' perspective, the introduction of this tool would result in a significantly higher effort required from them to get an interview, with the benefit of having a higher chance of actually getting the job after the interview. Even if candidates do not get to the interview stage after using this tool, they would still have gained valuable experience about what it would take for them to get a job and what companies are looking for in a candidate, competence-wise. This holds true for a normal recruiting process as well, where even if candidates do not receive a job offer, they have still gotten valuable experience. The difference is that through Operation Talent, candidates can get this valuable experience without ever leaving their computer, meaning less effort and no interaction with the company in question. Overall, the increased effort required from candidates puts increased pressure on user engagement being high enough in Operation Talent to sustain applicants fulfilling the process.

For applicants, the recruitment process might not start until they actually apply for a job, but for recruiters, that process starts with reaching out to potential candidates. Operation Talent should work as a marketing tool for the companies recruiting, which would implicate companies spending less money on marketing and promotion. Furthermore, circling back to the concept of viral sharing introduced in Section 3.5, Operation Talent has the potential to become viral due to its gamified nature, thematic influence and elicitation of both positive and negative emotions. This should imply that companies would receive more applications through Operation Talent than their normal recruiting process. Furthermore, the quality of candidates should be greater, both due to the presumed increased amount of applicants but also due to the competence the candidates have to pass in order to reach the interview stage.

By utilizing gameplay design patterns such as *High Score Lists*, Operation Talent impacts the recruiting process by turning it in to a more competitive experience. In a normal recruiting process, candidates seldom interact with other candidates, and

information about a candidate's progress is generally kept secret from people outside the company. Operation Talent gives candidates information about the performance of other candidates.

Gameplay design patterns can facilitate creating *rich experiences* (a concept mentioned in Section 3.4). Operation Talent has an overall focus on the rich experience called "The Challenging", as explained in Fokkinga's and Desmet's paper on designing for emotions, [18], which stands for a focus on eliciting the negative emotion "Frustration", which in turn is transformed into the positive emotion "Satisfaction". Operation Talent naturally approaches this rich experience due to the multitude of programming challenges that it provides, but the rich experience has been reinforced by patterns such as *Last Man Standing*, *Player Elimination*, *Penalties*, *Tension* and *Time Pressure*.

Evocation of the rich experience The Challenging is believed to improve user experience and engagement. On the other hand, working with the negative emotion also comes with a risk, which in this case would mean that the candidate gets stuck in the negative stimulus and never gets to experience the transformation to the positive stimulus.

Furthermore, in Section 3.4, Csikszentmihalyi's three factors [10] for achieving flow were stated:

- Clear goals
- Balance between challenges and skills
- Instant feedback

The use of gameplay design patterns in Operation Talent does, to some extent, support the first factor of having clear goals. For example, the use and visualization of the GDPs *Checkpoints* and *Main Quests* gives the candidate knowledge about what objectives must be completed in order to reach each checkpoint and to beat the game.

A "balance between challenges and skills" is the second presented factor of flow. Even though there are GDPs that contribute to this specific factor, they were not implemented in Operation Talent. All candidates (choosing the same mission) are faced with the same challenges (objectives) in the game and since the skill level varies from candidate to candidate, it results in some candidates facing problems that are too challenging or too easy to solve. This is a disadvantage in the perspective of flow, and hence also in the aspect of user engagement.

The third factor, "instant feedback", is in Operation Talent provided through the use of GDPs. By using patterns such as *Near Miss Indicators*, *Progress Indicators*, *Scores* and *Player Elimination*, the candidate is provided with continuous and fast feedback on the current state of the game. These different types of instant feedback help the candidates in their decision making, hence also work as a factor to achieve flow.

Even if the use of GDPs in Operation Talent theoretically supports flow and user engagement, one should be aware that these supportive elements can be reached outside the scope of Operation Talent. A recruiter could by other means (e.g. via meetings, phone or e-mail) provide the information that for example the patterns

Last Man Standing, Player Elimination, Penalties, Tension and *Time Pressure* provide to the candidates. Albeit recruiters having this possibility, this feedback is seldom provided due to, for example, requiring high administration workload to manually reach out and update all candidates.

6.2 Role of the Tool in the Recruiting Process

Operation Talent's role in the recruiting process is multifaceted. As mentioned in Section 6.1, the tool should act as a marketing tool for a company aiming to reach out to potential applicants. Furthermore, Operation Talent should work as a means for collecting and dealing with applications for job openings, as opposed to the usual application forms or e-mails. Operation Talent also encapsulates competence tests and provides, to some extent, automatic evaluations of these competence tests. The tool creates a competition between applicants, referred to as missions, where the winners receive a certificate of completion for each mission where they placed among the top contestants. The exact number of candidates receiving certificates on every mission is not yet established, but one could imagine the companies deciding on this number depending several factors, such as how many openings they have. Receiving a certificate is not equivalent with receiving a job offer. Rather, the certificate provides a significant advantage in the continued recruiting process. This means that the recruiting process starts with Operation Talent, but does not end with it.

7

Discussion

In this section, a discussion on all relevant parts of the thesis follows. This includes discussions on the results of the thesis, namely the collection of gameplay design patterns identified in the recruiting process and the concept design that followed a selection of these patterns, as well as a discussion of the process followed throughout the thesis and the tools involved in that process. As a wrap-up, opportunities of future work on this subject and ethical considerations regarding the work done in this thesis are presented.

7.1 Reflection on Results

One interesting aspect of the Operation Talent tool is what weight points and placement carry in the actual recruiting process. Initially, as seen in the concept model in Figure 5.6 and as described in Section 5.3.5.1, the end goal of the recruitment tool was that the candidate performing best in the game would receive a job offer. This was later revised, as described in Section 5.4.1, to candidates receiving certificates of completion rather than an actual job offer. This revision brings up questions regarding to what use this tool is for candidates placing high on the top list and how recruiters look at these prospects differently depending on their success in the game. The intention with the certificates is that they should provide a significant advantage for candidates wanting to get an interview with the company in question, but actual, quantifiable numbers of that advantage is yet not determined.

As mentioned in Section 6.1, the introduction of Operation Talent would imply candidates having to put significantly more effort into getting an interview. Whether this is positive for candidates and recruiters is up for discussion. One could argue that this hurdle will scare away a significant amount of candidates from applying for a job and thus not all leading to a broadened spectrum of applicants. On the other hand, one could argue that being able to apply and go through competence tests without ever interacting with the company directly and with no requirement of interest in the job will increase the number of applicants. But is this the type of applicants that companies are looking for? There will most likely be applicants with the only intent of exploring the tool and see if they have what it takes to get the certificate, or beat their friends applying for the same job. This doesn't necessarily have to be negative for the company in question. Even though they might get applicants with little to no interest in the job, those applicants might share the tool with their friends, and those friends could be the applicants the company is looking for.

It was mentioned in Section 6.1 that the concept Operation Talent did miss out on one of Csikszentmihalyi's flow factors [10], the factor of "balance between challenges and skills". The concept could be altered to embrace this balancing factor, enabling a change of difficulty level. The programming challenges in the tool could be adaptive toward the candidates performance and change the difficulty level, hence potentially allowing more candidates to achieve flow while using Operation Talent. Albeit the fact that this modification could increase user engagement through increased flow, questions regarding the fairness of such an approach are raised. If candidates receive challenges of varied difficulty level, how can the candidates be compared to each other? How is the score weighted? How does elimination work?

Many popular competitive games (e.g. Overwatch [15] and Counter-Strike Global Offensive [9]) solve the comparison and score questions by dividing players into different ranks and rank groups. The players' performance are evaluated during gameplay by the system and the players are then placed into different "difficulty level groups", enabling them to play against equally skilled opponents. What would happen if this method of sorting into ranking groups was applied on Operation Talent?

One way of using rank groups in Operation Talent could be to use them as a metaphor for different junior and senior developer roles. The missions available in the agencies could be grouped into different difficulty levels, allowing the candidates to either choose a difficulty level on their own or somehow be placed into one automatically by earlier placement tests.

The latter, to have placement tests would increase the time the candidate needs to invest in the game and also prevents candidates from trying out other ranking groups than the one they got placed in. A senior developer may want to apply for a more junior role to lower the workload, even if that also implies lower salary. Junior developers may want to try out for the senior roles to see what type of requirements such a role has, even if they most likely will get eliminated early on in the game. Therefore, the former, to let the candidates to choose a ranking group on their own, could be a more appropriate approach.

Discussing senior developers brings up the question of whether they would want to use a tool like Operation Talent at all. One can argue both ways. Why would a senior developer with a lot of experience and contacts (references on their competence) in the industry want to do extra tests in order to get another job? First of all, senior developers can use a game like Operation Talent with no strings attached. Having a concrete interaction with a contact could lead to false expectations, while Operation Talent is there for the curious developer to discover different companies and roles. Secondly, even senior developers may want to reach out to new fields beyond their current contact network.

Overall, having ranking groups would not only increase the experience in the form of engagement amongst the candidates, but also amongst recruiters, since they could create more targeted challenges for each ranking group and potentially reach out to more suitable candidates.

Even if Operation Talent is seen as a game, it has some differences from "ordinary games". Huizinga's [23] game definition, presented in Section 3.1.1, states that a

game is an activity decoupled from material interest and profit. This is one of the largest factors that makes Operation Talent different from ordinary games, due to the strong coupling with recruitment and jobs. As an example, playing ordinary games does not considerably affect players' lives, but in Operation Talent, the game could give players a better chance of getting a job, which could be considered as an impact on the players' lives.

It is possible that players do not find the same freedom and playfulness in Operation Talent, as opposed to other games, due to the serious part of recruitment and jobs. This could also result in players experience higher stress and pressure levels. However, Operation Talent can still be played like an ordinary game, if a player plays the game for the sake of playing rather than finding a job.

7.2 Reflection on Process

The plan for this thesis was to work iteratively according to the IDEO design process [25] described in Section 4.1.4. To a certain extent, this was achieved. But it is very easy to state beforehand that a project should be carried out in iterations, and much more difficult to actually do it. Working iteratively demands structure, and that structure needs to come mainly from the people executing the project. A framework can provide guidelines, but it is not the be-all end-all. In this thesis, there was a lack of structure as to what the different iterations should imply, what the goals of them should be and what length they should have. This lead to lengthy iterations in which too much work was done. So, a learning to be drawn from this is to structure iterations beforehand, and set up goals for the first one before any other work as done. After the time for an iteration is up, make sure to evaluate the work done in it, and set up a structure for the next one before initiating further work.

Section 4.1.3 introduces the concept of playcentric design, in which the importance of continuous play testing and validation of player experience goals are stated. As mentioned in Section 5.2, this thesis had a more user-centered focus in its early stages, which would have involved play testing sessions had it continued to have that focus. As the thesis evolved and went towards a more experimental focus, the play tests were a casualty of that change. Ideally, play tests would have been incorporated in the process, as they are an important part of game development, but other parts of the process were prioritized higher.

This thesis was conducted with a mindset of research through design (a concept introduced in Section 4.1.1), which comes with both advantages and drawbacks. First of all, this mindset fits well with interaction design in general, as it recognizes design activities as a part in generating knowledge. Second of all, it correlates well with the purpose of this thesis - exploring the gameplay design patterns framework for developing a recruiting process - as much of the generated knowledge actually lies in the process and not the end result or final design. However, conducting research through design, one should not expect to create validatable theories, which could be seen as an oxymoron when mentioned in a research context.

7.2.1 Gameplay Design Patterns as an Analysis Tool

Throughout the process of this project, gameplay design patterns have been utilized in one way or another. They have been used as both a tool to analyze and describe real-world processes, as well as a design foundation for a concept design. Regarding the analysis part, the gameplay design pattern framework works well. At first glance, it might sound peculiar to analyze the real world from a gameplay standpoint, but the work done in this thesis, specifically in Section 5.3.2, shows that it is easy to identify gameplay design patterns in real-world processes. However, this could be a product of games generally being based on some aspects of the real world, which in turn would lead to gameplay design patterns being based on real-world patterns. Nonetheless, using gameplay design patterns as a way of describing behaviour in real-world processes is, based on experiences gained in this project, efficient and worthwhile. The patterns in the collection (see [1]) are well worked through, and offer little to no ambiguity as to what they refer to, which makes working with them a breeze. However, one should be sensible to the fact that the one-sentence description provided for every pattern might not be enough to determine whether that pattern describes the intended behaviour.

The resource at [1] contains a list of all gameplay design patterns in alphabetic order with regards to their names. This list was used as a basis when analyzing the recruiting process. Patterns were, as a first step, gauged by their names. Approximately half of the time, the name was sufficient to determine whether the pattern was a match or not. If not, then a second step was initiated, where the one-sentence definition was assessed. Most of the time, this step was sufficient to determine whether the pattern should be included in the analysis or not. In rare cases, the whole description of the pattern was needed in order to make this determination. This way of working puts a lot of pressure on the names and the one-sentence definitions of patterns being descriptive enough...

Furthermore, using gameplay design patterns to analyze real-world processes can result in redundant and trivial information. For example, the pattern *Real World Knowledge Advantages* rings true for a recruiting process and was hence included in the analysis performed in this thesis, the result of which resides in Appendix C. But is this information really necessary in an analysis of a real-world process? It should be trivial that candidates make use of their real-world knowledge in a recruiting situation. What other information would they make use of?

7.2.2 Gameplay Design Patterns as a Design Foundation

When working with a design space, especially a relatively unexplored one, it is important to share some of the knowledge that has been gained along the way. In the case of this thesis, that design space is gameplay design patterns outside the context of games, and this section aims to express some experiences from working with gameplay design patterns.

Starting off, gameplay design patterns are in some aspects abstract descriptions of behaviour; concrete implementations of them may vary. This might seem like a trivial observation, but having knowledge of this will ease the design process.

Furthermore, gameplay design patterns are exceptional for idea generation. Brainstorming from scratch with no constraints at all is a hard skill to master and is not suitable for everyone. Using gameplay design patterns early in the design process to generate ideas is an exceptional way of working. Not only do the patterns act as a constraint on the design space, but they also provide ideas of artifact behaviour in and of themselves.

Continuing, gameplay design patterns provide a structured way of expressing design goals and intended behaviour. Together with the MDA framework (introduced in Section 3.1.4), gameplay design patterns provide even more structure. The MDA framework was shown to be a powerful tool to use when working with GDPs. It facilitates abstracting and visualizing the used patterns during the design work by giving a deeper understanding of the relations between patterns in the core gameplay (see 5.3.4). The MDA framework made it possible to see in what direction the user experience was heading. Knowing whether a pattern is a mechanic, dynamic or aesthetic is difficult, as it sometimes depends on how the pattern is utilized in the design. However, putting these labels on patterns is generally worth the extra effort, as it provides knowledge of the distribution of patterns within this framework. One might, for example, notice that there are a lack of aesthetics in the current design, and can thus focus the work around adding more aesthetics.

Furthermore, gameplay design patterns are not easily isolated. Trying to prove how a certain gameplay design pattern impacts aspects of a game or user experience is difficult. The difficulty comes partly from the fact that the impact of introducing a pattern can vary depending on what patterns already exist in a game, due to the different relations that patterns can have with each other. In addition to this, it is difficult, if not impossible, to design a game with only one gameplay design pattern present. This means that in order to enable the introduction of patterns one at a time, some patterns must already exist to form the core game.

Wrapping up, it is safe to say that not all gameplay design patterns have been discovered. The collection of gameplay design patterns found at [1] is still under construction and gameplay design patterns are continuously being discovered and added. There is a chance that some behaviour that should be considered a pattern is not on there yet.

7.3 Further Work

Operation Talent utilizes several gameplay design patterns, but there are still many left to explore together with this concept. The tool stands to gain significant improvement from yet another iteration by taking a step back to the brainstorm stage of applying new GDPs. One may discover patterns that contribute to achieving more flow or maybe patterns that reinforce the rich experiences.

Operation Talent has in this project been designed with the candidate in focus, hence the recruiter's side has been left out in the design of the tool. Hence, designing and developing the recruiter's side of Operation Talent could be an interesting way of building on the work done in this thesis. The recruiters are after all the ones that are supposed to use this application as a tool to find talented candidates. What are the recruiters needs and how will their interface look? Can gameplay design

patterns be used to make it engaging on the recruiter's side as well? Should the recruiter play a game with a goal of hiring the best candidates? All of the above questions are of interest to investigate further.

The design work performed in this thesis involved no user testing or user evaluation of the final prototype. This entails consequences, as this concept has not been exposed in its true setting. Hence, implications of what impact gameplay design patterns have on the recruiting process are merely speculations and educated guesses, as the concept of Operation Talent has yet not been utilized in a recruiting process. One way to look at this thesis is as a proof of concept, or a pilot study of how gameplay design patterns can impact the development of a recruiting process. Hence, an interesting way of continuing on the work done in this thesis would be to actually develop and test Operation Talent in a real environment and collect data on user engagement, number of applicants, number of applicants moving on to the interview stage and so on.

Furthermore, in this thesis, a website design of Operation Talent was made, implementing different GDPs. A natural next step would be to design it for mobile devices such as smartphones and tablets. People are today always on the go and thus, a mobile version would open up for other possibilities regarding the application of GDPs. With mobile devices come interesting design questions thanks to the spatial information, but also the focus on interaction in the form of touch, sound, movement and vibration. How could GDPs be integrated with the data that mobile devices provide and how would it impact the users?

Gameplay design patterns have in this thesis been applied in the contexts of recruitment within the programming field, but how generalizable is the use of GDPs in other contexts? Further research of applying gameplay design patterns can be done in branches of other industries, such as finance, construction, health and social care, both in the context of recruiting as well as other company processes.

Finally, another aspect to consider if building on the research performed in this thesis is the medium of the design. Operation Talent resides in the digital space, as it felt most natural to apply gameplay design patterns in a digital medium. But does it have to be computer-based? Would it not make more sense to develop a recruiting game based in the physical world? Designing a live action role-playing game to be used in a recruiting process would be an interesting and different angle to approach the problem from.

7.4 Ethical Considerations

Designers play an important role in society, as they are contributing to shaping the future of technology. It is therefore of importance to look at projects from a holistic perspective, and be able to predict what affects the design could have on society as a whole. This project may contribute to further digitizing and gamifying the recruitment process of programmers.

The digitization part might lead to a decreased demand for professional recruiters, leading to fewer employment possibilities within the field of recruitment. On the other hand, this project has the potential to improve the talent sourcing of programmers, leading to increased employment within the programming field.

The gamification part of the project is part of a general trend in interaction design, and incorporating game elements into all routine processes could potentially have a negative effect in that people disinterested in games feel left out, or that the positive effects that gamification can have lessens due to an inflation of gamified processes.

Turning a recruitment process in to a competitive game, and publicly displaying the performance of other candidates could imply a greater amount of pressure amongst candidates. This could in turn lead to increased levels of stress and anxiety.

A recruitment platform that is heavily focused on showing competence before employment or even the first interview could decrease the opportunity for candidates to get a foothold in the labour market. Candidates that currently lack knowledge (which is required in Operation Talent) but are good learners could be missed out on because they don't get the chance to show their skills due to being eliminated before ever reaching the interview.

If a recruitment tool succeeds to become deeply engaging it will most likely also attract many candidates. Even if there is a lack of candidates in the field of programming today, a deeply engaging recruitment tool could on a larger scale lead to a saturated market for programmers.

Another potential issue with digitization is that it can lessen the need for face-to-face interaction in the recruitment process, which can be argued to be counter-productive.

8

Conclusion

This master's thesis focused on investigating what impact the application of gameplay design patterns has on the process of recruiting programmers in general, as well as how that impact correlates to user engagement. As a step in this investigation, a general recruiting process was analyzed, using gameplay design patterns to describe different elements and behaviours of it. Furthermore, a selection of these identified patterns were used as a design foundation to create a recruitment tool called *Operation Talent*. In addition to the patterns identified in a general recruitment process, a number of other gameplay patterns were added to the design foundation, as an attempt at diversifying the structure and behaviour of the recruiting process.

Operation Talent encapsulates and visualizes the application of gameplay design patterns on a recruiting process and was developed to facilitate the investigation of what impact gameplay design patterns have on the recruiting process. It was concluded that Operation Talent, through the use of gameplay design patterns, impacted the recruiting process by changing the order of the different stages in a general recruiting process. It also demands more effort from applicants in order to get an interview at a certain company. Furthermore, Operation Talent should attract candidates broadly, and narrow them down to the most fitting without much effort required from the recruiters, which could implicate a lower workload for recruiters.

The impact is generated through a combination of implementing specific gameplay design patterns in the design and the use of a digital medium as a basis for the design.

The research performed in this thesis is only the tip of the iceberg of a reformed recruiting process with a larger focus on the needs of the candidates rather than the companies recruiting. Future work could implicate further development of Operation Talent, introducing and testing the tool with real users and evaluating how well it actually works.

More generally, it could be interesting to investigate the application of gameplay design patterns on other routine processes. Gameplay design patterns is a framework for describing games, and games are a proven commodity for user engagement. Applying this framework on routine processes should therefore, by the property of transitivity mean increased user engagement for completing these processes. Whether this is actually true or not is a case for the future.

Bibliography

- [1] Staffan Björk. *Collection of gameplay design patterns*. Accessed 2018-01-31. 2018. URL: <http://www.gameplaydesignpatterns.org/>.
- [2] Staffan Björk, Sus Lundgren, and Jussi Holopainen. *Game Design Patterns*. 2003.
- [3] Ian Bogost. *Gamification is Bullshit*. Accessed 2018-06-09. 2011. URL: http://bogost.com/writing/blog/gamification_is_bullshit/.
- [4] Codility. *Codility*. Accessed 2018-02-15. URL: <https://www.codility.com/>.
- [5] Elizabeth L. Cohen. “What makes good games go viral? The role of technology use, efficacy, emotion and enjoyment in players’ decision to share a prosocial digital game”. In: *Computers in Human Behavior Vol. 33 No. 6* (2014), pp. 321–322.
- [6] Elizabeth L. Cohen. “What makes good games go viral? The role of technology use, efficacy, emotion and enjoyment in players’ decision to share a prosocial digital game”. In: *Computers in Human Behavior Vol. 33 No. 6* (2014), pp. 321–329.
- [7] Elizabeth L. Cohen. “What makes good games go viral? The role of technology use, efficacy, emotion and enjoyment in players’ decision to share a prosocial digital game”. In: *Computers in Human Behavior Vol. 33 No. 6* (2014), p. 324.
- [8] Alan Cooper. *About Face 3: The Essentials of Interaction Design*. 2007, pp. 77–83.
- [9] Valve Corporation. *Counter-Strike Global Offensive*. 2012. URL: https://sv.wikipedia.org/wiki/Counter-Strike:_Global_Offensive.
- [10] Mihaly Csikszentmihalyi. *Beyond Boredom and Anxiety: Experiencing Flow in Work and Play*. Jossey-Bass, 1975.
- [11] Mihaly Csikszentmihalyi. *Flow : psychology, creativity, optimal experience with Mihaly Csikszentmihalyi*. Accessed 2018-02-08. 2014. URL: <https://chalmers.kanopystreaming.com/video/flow-psychology-creativity-optimal-experience>.
- [12] Mihaly Csikszentmihalyi. *Flow and the Foundations of Positive Psychology: The collected Works of Mihaly Csikszentmihalyi*. Springer, 2014.
- [13] Sebastian Deterding et al. “From Game Design Elements to Gamefulness: Defining “Gamification””. In: *MindTrek ’11 Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments* (2011).
- [14] Daniel Engelberg and Ahmed Seffah. *A Framework for Rapid Mid-fidelity Prototyping of Web Sites*. 2002.

-
- [15] Blizzard Entertainment. *Overwatch*. 2016. URL: <https://playoverwatch.com/en-us/game/overview/>.
 - [16] Carlo Fabricatore. *Gameplay and game mechanics: A key to quality in videogames*. 2007.
 - [17] Haakon Faste et al. *Brainstorm, Chainstorm, Cheatstorm, Tweetstorm: New ideation strategies for distributed HCI design*. 2013.
 - [18] Steven F. Fokkinga and Pieter M.A. Desmet. “Ten ways to design for disgust, sadness, and other enjoyments: A design approach to enrich product experiences with negative emotions”. In: *International Journal of Design* 7(1) (2013).
 - [19] Nico H. Frijda. “The Laws of Emotion”. In: *American Psychologist Vol. 43 No. 5* (1988), pp. 349–358.
 - [20] Tracy Fullerton. *Game Design Workshop - A playcentric approach to creating innovate games - Second edition*. Elsevier Inc., 2008.
 - [21] William Gaver. *What Should We Expect From Research Through Design?* 2012.
 - [22] HackerRank. *About HackerRank*. Accessed 2018-02-15. URL: <https://www.hackerrank.com/aboutus>.
 - [23] Johan Huizinga. *Homo Ludens*. 1938.
 - [24] Robin Hunicke, Marc LeBlanc, and Robert Zubek. *MDA: A Formal approach to game design and game research*. 2004.
 - [25] IDEO.org. *The Field Guide to Human-Centered Design*. 2015.
 - [26] Catherine Courage Kathy Baxter and Kelly Caine. *Understanding your users: A practical guide to user research methods*. 2nd ed. Morgan Kaufmann, 2015.
 - [27] Kattis. *Kattis homepage*. Accessed 2018-02-15. URL: <http://kattis.com>.
 - [28] Jiro Kawakita. *The Original KJ Method*. 1982.
 - [29] Gabriela Kiryakova, Nadezhda Angelova, and Lina Yordanova. *Gamification in Education*. 2014.
 - [30] Raph Koster. *A Theory of Fun for Game Design*. 2nd ed. O’Reilly Associates, 2013.
 - [31] Sari Kujala. “UX Curve: A method for evaluating long-term user experience”. In: *Interacting with computers Vol. 23 Issue 5* (2011), pp. 473–483.
 - [32] Petri Lankoski and Staffan Björk. *Game research methods*. 2015.
 - [33] Yefeng Liu, Todorka Alexandrova, and Tatsuo Nakajima. *Gamifying Intelligent Environments*. 2011.
 - [34] Rossella Lorenzi. *Oldest Gaming Tokens Found in Turkey*. Accessed 2016-02-05. 2013. URL: <http://news.discovery.com/history/archaeology/oldest-gaming-tokens-found-130814.htm>.
 - [35] Microsoft. *A national talent strategy*. Accessed 2018-02-07. 2012. URL: <https://news.microsoft.com/download/presskits/citizenship/MSNTS.pdf>.
 - [36] Dong Nguyen. *Flappy Bird*. dotGEARS, 2013.
 - [37] Lene Nielsen. *Personas - User Focused Design*. 2012, pp. 1–6.
 - [38] Lene Nielsen. *Personas - User Focused Design*. 2012, p. 7.
 - [39] Lene Nielsen. *Personas - User Focused Design*. 2012, p. 122.
 - [40] Robert Plutchik. “Emotion: Theory, research, and experience”. In: *Theories of emotion Vol. 1* (1980).

- [41] Horst W. J. Rittel and Melvin M. Webber. “Dilemmas in a general theory of planning”. In: *Policy Sciences* (1973), pp. 155–169.
- [42] James A. Russel. “A circumplex of affect”. In: *Journal of Personality and Social Psychology Vol. 39 No. 6* (1980).
- [43] Katie Salen and Eric Zimmerman. *Rules of play: Game design fundamentals*. 2003.
- [44] Future Skill. *About Future Skill*. 2018. URL: <https://futureskill.com/about>.
- [45] Jared M. Spool. *The KJ-Technique: A Group Process for Establishing Priorities*. 2004. URL: https://articles.uie.com/kj_technique/.
- [46] Bernard Suits. *The grasshopper: Games, life and utopia*. 1990.
- [47] Alistair Sutcliffe. *Designing for User Engagement: Aesthetic and Attractive User Interfaces*. 2009, pp. 3–16.
- [48] Alistair Sutcliffe. *Designing for User Engagement: Aesthetic and Attractive User Interfaces*. 2009, pp. 7–8.
- [49] John Zimmerman, Jodi Forlizzi, and Shelley Evenson. *Research through design as a method for interaction design research in HCI*. 2007.

A

Results of Brainstorming Features For Recruiting Platform

The following list represents a collection of ideas for features that were generated during a brainstorm session focused on four broader categories of features: Community Social Tools, Acknowledgement of Success, Competitive Measures and finally The Big Picture.

Community Social Tools

- Rate Tag challenges
- Separate chat for each challenge as well as a global chat
- Avatars to put a face on other users
- A possibility to save moments (that can be shared)
- Forum section
- Community-created and community-approved challenges
- Social share (Facebook, Twitter etc...)

Acknowledgement of Success

- Positive feedback after every submit: “Great Job! Your tank is owning”
- In the code editor: Visualize how fast your algorithm is (Big O notation)
- Give feedback about how close a user is to beating the next person on the high score list
- If a user plays against others’ AI - make the final visualization of victory large
- Show a count of how many achievements a user has (5/272)
- Badges showing “Won season 1” for example
- Social share (Facebook, Twitter etc...)

Competitive Measures

- Matchmaking lobby (team vs team)
- Names on the ranks to make the user want to climb, for example “bronze, silver, gold”
- Seasons/ladders
- Being able to compare stats in detail
- Trash talking with emotes
- Spectators
- Elo-rating
- Prizes

Competitive Measures

- Line diagram of rank progress
- Contribution visualisation, much like github
- Visualisation of when each achievement or badge was earned
- Visualise how the user climbed the highscore list with each submit on every challenge

B

Programming Challenges Questionnaire

Programming Challenges

We are conducting a master thesis about online programming challenges. We are currently analysing the user experience in programming challenges and their gamification possibilities. We would be happy if you could help us out! There are a total of 7-12 questions, where 1-4 of them are optional (depends on your earlier experience). The survey should only take about 5 minutes.

***Required**

Definition

Programming challenge definition: an algorithmic problem (challenge) that you are supposed to solve with code. The website containing the problem can possibly provide an online editor for you to use. The challenge may be of the competitive type, hence give you a score and rank you against other competitor's algorithms. Programming problems that appear in work or in school courses aren't included in this definition. The programming problems that this definition aims at are meant to be solved to entertain/learn/challenge yourself.

Questions

1. Which programming languages do you prefer? *

Tick all that apply.

- ☐ JavaScript
- ☐ Python
- ☐ Java
- ☐ C
- ☐ C++
- ☐ C#
- ☐ PHP
- ☐ Golang
- ☐ Other: _____

2. Have you participated in any online programming challenges? *

Mark only one oval.

- ☐ Yes *Skip to question 3.*
- ☐ No *Skip to question 12.*

Programming Challenges - Earlier experience

3. Why have you participated in these types of challenges? **Tick all that apply.*

- ☐ To learn new skills
- ☐ To improve existing skills
- ☐ To compare my skills to others
- ☐ To compete against others
- ☐ To find a job
- ☐ For entertainment purposes
- ☐ Other: _____

4. Where have you done these programming challenges? *

add your own alternative if it doesn't exist

Tick all that apply.

- ☐ CoderByte <https://coderbyte.com/>
- ☐ Kattis <http://www.kattis.com/>
- ☐ Project Euler <https://projecteuler.net/>
- ☐ HackerRank <https://www.hackerrank.com/>
- ☐ CodinGame <https://www.codingame.com/>
- ☐ Software Skills <http://honeypot.softwareskills.se/>
- ☐ TopCoder <https://www.topcoder.com/>
- ☐ Other: _____

5. Name one of your favourite challenges and where you found it *

6. Why is that challenge your favourite?

7. What type of programming challenge do you enjoy the most? **Mark only one oval.*

- ☐ Text input and text output
- ☐ Text input and visualised output
- ☐ Other: _____

8. What factors do you think are most important for you to give continuous efforts in a programming challenge?*Tick all that apply.*

- ☐ I can be experimental
- ☐ Easy to get started
- ☐ Challenging problem
- ☐ I receive a highscore
- ☐ I get achievements that indicate success
- ☐ I can see detailed statistics of my performance
- ☐ I can beat others highscore
- ☐ The difficulty adjusts as I become better
- ☐ Other: _____

9. What factors do you think would increase the odds of you sharing the challenge to others?*Tick all that apply.*

- ☐ The challenge has an interesting background story and theme
- ☐ The visualisation of the challenge is engaging to look at
- ☐ I can do the challenge with a team (where each team member contributes to different parts of the problem)
- ☐ Challenging problem
- ☐ I can challenge a friend and compare our results
- ☐ Other: _____

10. How much time do you spend on a challenge (on average)? **Mark only one oval.*

- ☐ < 1 h
- ☐ 1-2 h
- ☐ 2-5 h
- ☐ 5-10 h
- ☐ > 10 h

11. If you got to create your own programming challenge, what would it be like?

*Stop filling out this form.***Programming Challenges - No experience**

12. What sounds most exciting, rank #1-5, where rank #1 is the best **Mark only one oval per row.*

	#1	#2	#3	#4	#5
Program the best art auction bidder	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Program a battle tank that moves and shoots enemies	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Program a bluff AI for the game bluff	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Program the most efficient elevator	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Program a space ship that moves and shoots	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

13. What would be most exciting - challenging a lot of random players or challenging a few of your friends? **Mark only one oval.*

	1	2	3	4	5	6	
Random players	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Friends

14. How much time would you be ready to spend on a challenge? **Mark only one oval.*

- ☐ < 1 h
- ☐ 1-2 h
- ☐ 2-5 h
- ☐ 5-10 h
- ☐ > 10 h

15. Where would you most likely read about, and join a challenge? **Tick all that apply.*

- ☐ Facebook
- ☐ Email from a company
- ☐ Message from a friend
- ☐ LinkedIn
- ☐ In the group "Kodapor" on Facebook
- ☐ Other: _____

16. Do you have any ideas that you would want to see in a programming challenge?

C

Gameplay Design Patterns in a Recruiting Process

This appendix contains the complete list of gameplay design patterns that were identified in a general recruiting process for programmers. The structure of the list is as follows: The bold title represents the name of the gameplay design pattern in question, the first bullet point represents a quote of a one-line description of the pattern taken from [1] and the second bullet point represents a description of how the gameplay design pattern in question correlates to the recruiting process.

Abilities

- *“Actions that agents can do which allow players to affect game states”*
- Every candidate has a set of abilities, and using / showing them will affect how they perform in the recruiting game.

Achilles’ Heels

- *“A special weakness that can be exploited to cause defeats easier than can be done by using the other means available”*
- The correlation here is that every candidate has weaknesses, and if the recruiter happens to identify them, asking questions about them can be used as a technique to eliminate a candidate from the process.

Actions Have Diegetically Social Consequences

- *“Actions by a person in a game world influences how other people perceive and interact with that person”*
- A recruiting process is all about building your own perception of the candidate. Every action the candidate takes will have an impact on this perception, for example how they answer questions, perform in tests etc.

Agents

- *“Diegetic game elements that can be interpreted as having goals in game worlds, and working towards those goals”*
- Both candidates and recruiters can be seen as agents, as they both have their own goals, and continuously work towards achieving them.

Alarms

- *“Alarms are abstract game elements that provide information about particular game state changes”*
- A candidate can be given information about how they are progressing in the process by for example reading the body language of the recruiter during an

interview. Their body language can signal problem. In this case, the body language of the recruiter is the abstract game element.

Always Vulnerable

- *“The situation where players’ resources in game instances can be threatened even when they are not playing”*
- If the recruiter’s perception of a candidate is correlated to the candidate’s resources (since that is what the candidate is trying to build up), that perception can change even when the candidate is not actively interacting with the recruiter, for example if the recruiter checks the candidate’s Facebook page and sees something inappropriate.

Analysis Paralysis

- *“Halts to gameplay due to one or more players spending considerable amounts of time planning or understanding the consequences of different possible actions”*
- This can occur in many ways during a recruiting process. For example, if a candidate spends a considerable amount of time pondering a question that the recruiter just asks, this contemplation will halt gameplay. The different actions in this case would be the different answers the candidate can give, or the different ways in which the candidate can present the answer.

Arithmetic Progression

- *“A linear relationship between the effort put into an action and its potential reward or risk”*
- Generally, the more effort a candidate puts into the recruiting process, the better the performance / reward. While maybe not always linear, as recruiting is subjective, one could still see a correlation between effort and success.

Asymmetric Starting Conditions

- *“That players do not begin their game sessions with the same possibilities”*
- The correlation here is quite obvious. Every candidate is unique, and have different skills, abilities and possibilities.

Asynchronous Gameplay

- *“Gameplay that does not require - or demands - that not all players are playing at the same time”*
- A recruitment process does generally not require all candidates to be actively participating simultaneously. Rather to the contrary. A recruiting company can not handle all candidates at once. People can drop in and out, people are at different stages in the process etc.

Attention Demanding Gameplay

- *“Gameplay where players can easily suffer bad consequences for being inattentive at any given point”*
- Generally, not paying attention when interacting with a recruiter is a bad habit. Take an interview for example. The candidate not paying attention to the questions the recruiter is asking will most likely have bad consequences.

Campaigns

- *“A series of designed gameplay experiences for a game that are intended to be played in sequence”*
- The different stages of a recruiting process can be correlated to the designed gameplay experiences. A recruiting process is generally progressed through in sequence.

Challenging Gameplay

- *“That players experience the gameplay as difficult or challenging their abilities and skills”*
- A recruiting process is generally intellectually challenging for the candidate, as the whole point is proving their abilities, both technically and socially.

Characters

- *“The abstract characteristics of diegetic persons”*
- The diegetic persons in this case is the candidates and recruiters, which all have different characteristics.

Character Defining Actions

- *“That the choices of actions players make during gameplay define their characters”*
- The character in this case is the candidate, and every action the candidate makes will define the way recruiters perceive the character.

Character Development

- *“Changes in characters’ abilities, skills, or powers as part of gameplay”*
- In the eyes of the recruiter, the candidate is building its character throughout the recruiting process, as the recruiter gets more information about the candidate.

Checkpoints

- *“Locations in game worlds which signify game state changes directly related to the progress of some goal”*
- In the case of a recruiting process, the checkpoints are not necessarily locations by definition, but rather metaphorically. Progressing from one stage of the recruiting process to the next can be viewed as a checkpoint. The state changes significantly, as new tasks become available when reaching the next stage.

Context-dependent Reactions

- *“Agents in a game reacts to events and objects (including other agents) depending on their context”*
- One could see candidates taking different actions when meeting their recruiters in different contexts. For example, the candidate might act differently during the first interview as opposed to the second.

Contextualized Conversational Responses

- *“The ability of agent[sic] to gives responses in a conversation depending on all relevant game states”*
- This one speaks for itself. Humans are generally able to give different responses depending on multiple factors.

Coupled Games

- “Games that are designed so that the events in one game affects events in another game”
- The correlation between this pattern and the recruiting process is a little bit harder to grasp than a lot of others. Let’s say that a candidate is applying for multiple jobs. If all those recruiting processes are seen as different games, then the events in one game will affect the other games, if the candidate for example gets a job offer in one process.

Death Consequences

- “Gameplay consequences of avatars or characters dying”
- The death of a character in the case of recruitment is of course metaphorical. A candidate being denied further progress in a recruiting process can be viewed as a character dying in a game. And the consequences of this death is that the candidate generally can no longer apply for the same job, at least within the foreseeable future.

Delayed Effects

- “The effects of actions and events in games do not occur directly after the actions or events have started”
- This rings true for a recruiting process as well. The effects of taking an interview or doing a competence test are not revealed immediately, but will rather take some time to get feedback on.

Dialogues

- “Diegetic conversations controlled by a game system”
- If the recruiter is viewed as the game system, then an interview is a perfect example of a dialogue in a recruiting process.

Difficulty Levels

- “Controls in a game for letting player choose how difficult the gameplay should be”
- The difficulty in a recruiting process can be altered by applying for different types of jobs. For example, applying for a senior role generally means a more difficult recruiting process.

Early elimination

- “The ending of players’ game sessions well in advance of the end of game instances”
- This depends on how one defines a game instance, but the game instance from a recruiter’s point of view might be the process until a role is fulfilled by a candidate. This means that for most of the candidates, the game session will end in advance due to them being eliminated at an earlier stage of the recruiting process.

Emotional Attachment

- “The ability of agents to have noticeable emotional relations inside the game world to the diegetic phenomena in that world”
- This one speaks for itself. Candidates generally have an emotional relation to the job and/or company they are applying for.

Enemies

- *“Game elements that are actively hindering players to complete game goals”*
- One could correlate the recruiter to an enemy in the game. They are constantly performing checks to see if a candidate is a good fit for the role. This could be seen as them actively hindering candidates from completing the goal of getting the job.

Exaggerated Perception of Influence

- *“Players perceive that they can influence the outcome of the game, regardless of whether this is correct or not”*
- Generally, candidates might have a feeling that if their performance in the recruiting process is good enough, they will get the job. While in reality, the outcome of the game lies solely in the hands of the recruiters’ subjective assessment.

Extended Actions

- *“Actions that take so long to complete that they require players to miss opportunities to perform other actions in order to complete them”*
- If an action is looked at as going through a recruiting process at one company, that action might hinder candidates from performing other actions such as going through the recruiting process at another company.

Finale Levels

- *“Levels or other enclosed gameplay areas that are constructed so they are experienced as the final part of a narration arc and resolves that arc”*
- The final interview, a final competence test or such can be looked at as the final level of the recruiting process.

First Player Advantages

- *“The advantageous effect of being the first player to do actions”*
- Being the first candidate to apply for a job can generally have benefits, as the candidate gets a chance to set a first impression on the recruiter without them having something to compare to.

Fog of War

- *“Hiding information about parts of game worlds that are not being observed or have not yet been explored by players”*
- The candidates do generally not have too much information about what will happen later on in the process.

Freedom of Choice

- *“The freedom to choose between several different actions or choices which all seem meaningful”*
- Candidates can choose freely between which jobs and roles to apply for.

Functional Roles

- *“Gameplay where responsibility for different types of game actions can be divided between participants”*
- Candidates and recruiters are example of roles which have different responsibilities in the game.

Further Player Improvement Potential

- *“That players have the possibility to increase their skills in handling the game-play”*
- As candidates gather experiences in different recruiting processes, they will most likely become better at performing them next time.

Gain Competence

- *“The goal of being able to perform a specific ability to a certain level of competence within a game”*
- This goes hand in hand with *Further Player Improvement Potential*, because the candidates would want to train the abilities that are required for completing a recruiting process.

Game Element Insertion

- *“The insertion of game elements into game instances”*
- Tasks are gradually introduced during a recruiting process, ranging from interviews to competence tests.

Game Masters

- *“Facilitators of game worlds, and of players’ interactions with these worlds”*
- The recruiters could be correlated to game masters, as they, to a certain extent, facilitate candidates interactions with the recruiting process.

Game Over

- *“The event which makes a player no long[sic] able to participate in the game-play of a game instance”*
- Failing to proceed in a recruiting process generally means that the candidate can not apply again during the same instance of the recruiting process.

Health

- *“A measure of how much damage or other negative consequences avatars, characters, or units can take before they suffer serious penalties”*
- While not measured and displayed the same way as in a game, a candidate’s certainly has a fixed amount of ‘health’, where certain actions, such as answering a question the wrong way, can lead to the candidate losing ‘health’. After all health is lost the candidate is dismissed from the recruiting process, resulting in the candidate suffering serious penalties.

Higher-Level Closures as Gameplay Progresses

- *“Closures that occur progressively become more important as the game is played”*
- It becomes increasingly more difficult to advance from a stage in a recruiting process the later the stage is in the process.

Invites

- *“Game actions the result in new players being invited to join the game”*
- Invites can be correlated to a person receiving a referral from someone, which leads to that person becoming a candidate in the recruiting process.

Irreversible Events

- *“Events whose effect on the game state cannot be undone”*
- Certainly, it is difficult, if not impossible to reverse an action taken in a recruiting process.

Last Man Standing

- *“Gameplay where players or teams are actively trying to eliminate each other to be the last survivor”*
- While candidates might not feel that they are actively trying to eliminate other candidates, in fact, that is what they are trying to achieve. They want to be the last man standing in the recruiting process and get the job.

Levels

- *“A level is a part of the game in which all player action takes place until a certain goal has been reached or an end condition has been fulfilled”*
- A recruiting process generally consists of different stages or levels, in which candidates can fulfill certain goals and either advance or get eliminated.

Main Quests

- *“Quests whose completion provides the main winning condition of games”*
- Each *level* in a recruiting process generally consists of one or more main quests that the candidate must complete in order to advance towards the next level.

Player Characters

- *“Characters in games that are under players’ direct control or represent the players role in the gameplay”*
- An argument can be made that people choose to represent themselves as different characters depending on which job they are applying for. One could say that when people are creating their resumes and writing their cover letter, they are in fact creating a character, which is under their direct control throughout the recruiting process.

Player Elimination

- *“The forced endings of players’ game sessions before game instances are finished”*
- The elimination can be correlated to a recruiter telling candidates that they will not advance to the next stage of the process, because they have not fulfilled the required goals.

Player-Created Characters

- *“Characters whose creation players’ have had some influence over”*
- The same reasoning as for *Player Characters* works here. It is the candidates that are creating the characters, which explicitly means that they have had influence over them.

Possibility of Graceful Surrender

- *“The ability for players to leave gameplay or surrender to other players without negatively affecting the gameplay for all other players of a game instance”*
- A character electing to leave a recruiting process will not affect the other candidates’ experiences negatively.

Predefined Goals

- *“Goals of the game that have been predefined by designers before gameplay begins”*
- If the recruiters are viewed as the game designers, one could say that they have defined the goals of the recruiting process.

Private Game Spaces

- *“Parts of the game space that only a single player can manipulate directly”*
- Candidates will most likely have their own space in the recruiting process, as they are rarely interacting with other candidates. This means that they have a space that only they can manipulate directly.

PvE (Player versus Environment)

- *“Gameplay where game systems, rather than other players, provide challenges to players”*
- Since the candidates are never really interacting with their opponents, one could say that the recruiting process is a PvE game, as the system (the recruiting company) provides the challenges for the candidates.

Quizzes

- *“Collections of questions asked to players as part of gameplay.”*
- Questions are asked to the candidates during for example interviews, and often as part of the competence test.

Races

- *“Gameplay goals that need to be worked against under pressure, often but not necessarily in competition against others”*
- Oftentimes, the competence test has some sort of time-limit and could be seen as a competition with the other candidates.

Real World Knowledge Advantages

- *“Games where players can make use of specific real world knowledge to their advantage”*
- This one is almost trivial. It is evident that the candidates are using their real world knowledge to their advantage in the recruiting process.

Real-Time Games

- *“The progression of game time during play is tied to the progress of real time”*
- This one is also trivial. The time during the recruiting process progresses in the same way as real time.

Rewards

- *“Things received in games which are perceived as positive”*
- The most obvious reward in a recruiting process is getting a job, but other, more subtle rewards can be received as well. For example, getting positive feedback on an interview or competence test is a form of reward as well, even though they are not ‘things’ by definition.

Role Selection

- *“Selecting what gameplay abilities one will have by choosing from a limit[sic] number of roles”*
- If one looks at applying for a specific role at a company as limiting the abilities a candidate will have during the recruiting process, this pattern makes sense. The candidate will only be able to utilize the abilities required for the role during the process.

Secret Scoring Mechanisms

- *“Ways of gaining points in a game which is not revealed until used or until the final scoring phase of the game”*
- In a recruiting process, it is difficult for the candidate to know how they are being assessed, and what ‘points’ they are getting. While the recruiter might not reveal the exact scoring mechanism used to select the candidate at the end of the process, the mechanism is still secret.

Skills

- *“Representation of how likely diegetic agents are to succeed with a type of activity that can be improved through experience”*
- As with everything, candidates can have different skill levels which will determine how well they will perform in the process.

Social Rewards

- *“Rewards provided by a game that can help players’ social regard rather than giving direct in-game benefits”*
- It speaks for itself that having good social skills will give a candidate a higher chance of succeeding in a recruiting process.

Social Skills

- *“Gameplay where players social abilities affect how well they succeed with their actions”*
- If the game ends when a candidate gets hired, then that reward will provide the candidate with better social regard, as having a job often leads to higher social status.

Solution Uncertainty

- *“Uncertainty in a game which stems from not being aware of an existent solution to a challenge”*
- The challenges provided to the candidate in a recruiting process oftentimes don’t have an obvious solution. For example, knowing what the ‘correct’ answer is to an interview question is unintuitive.

Tension

- *“The feeling of caring about the outcome of actions or events in a game without having full control over them”*
- This is true for most candidates. They care about getting the job, but really has no control over that outcome, since it is the recruiter’s subjective assessment of the candidate’s performance that determined whether the candidate advances in the process or not.

Time Pressure

- *“That gameplay imposes a sense of need for complete[sic] actions or goals quickly”*
- For example, in an interview, the candidate oftentimes feels a sense of time pressure, as they don’t have unlimited time to answer a question. Also, the competence tests generally comes with a time limit.

Turn Taking

- *“Letting one player do some action or actions before letting other players act”*
- Generally, candidates take turns in doing actions such as interviews.

Unsynchronized Game Sessions

- *“Gameplay where the beginning and end of different players’ game sessions are not aligned”*
- Candidates’ game sessions (when they are performing actions related to the recruiting process) are almost never aligned.

Winner Determined After Gameplay Ends

- *“Games where the winner(s) are determined after gameplay ends”*
- The game, for a candidate, ends when all stages of the recruiting process have been completed (or when the candidate has been eliminated). The recruiting company decides, after a candidate has completed all stages, whether or not that candidate gets a job. This means that the winners are determined after candidates’ gameplay are over.