DataFlex

Educational game about data centers for children

25 June 2020, Delft University of Technology

Gedeon d' Abreu de Paulo
Bilal el Attar
Ali Al-Kaswan
Nathan Kronstadt
Gianni Wiemers

Supervisors:	Dr. Ir. W.P. Brinkman, TU Delft, Supervisor Drs. Ir. S. de Wit, VHTO, Client
Bachelor Coordinator:	Ir. T.A.W. Overklift, TU Delft

An electronic version of this report is available at http://repository.tudelft.nl/

Summary

Women are largely underrepresented in IT, girls' interest in STEM and IT fields tends to drop throughout secondary education. Educational games are a great tool to change the perception of certain topics, as well as changing the behaviour of the players. Thus, this report describes the development of a game to make the field of IT more appealing to girls between the ages of 10 and 14.

After collecting requirements with the client and doing a literature study a design is proposed. The final product is a two-player 2D Role-Playing-Game with puzzle elements, specifically designed to be played in a classroom environment. The game takes place in a data center and will show the players the societal importance of data centers as well as the diversity of the work in data centers. The gameplay consists of exploring a data center, talking with both male and female employees in various roles, helping them with their work through minigames, and solving a mystery. The game was designed to specifically cater to girls and to break stereotypes regarding women in IT.

Preface

For the final course, TI3806, of the study Computer Science and Engineering at the Delft University of Technology we have created a product. This product was constructed in a period of ten weeks under the guidance of a supervisor.

We chose to do the project to create an educational game for VHTO and Interxion because we wanted to build something meaningful, be creative in our own way. Moreover, this project had the potential to fulfill the needs of every member of the group. Each member has their wishes, namely: to do something with or for kids, to do something for a company, to solve a real problem, and to do something that wasn't just coding.

Before starting this course we already were acquainted with one another. This allowed for the workflow to be optimal. We understood each other and decisions were discussed together. It is an advantage when you work with people you already know the strengths and weaknesses of, it allows for better understanding and an efficient way of working.

We tackled the problem by brainstorming a lot in the beginning, as was advised by our supervisor. This allowed us to have many ideas to pitch to the client. The client gave feedback on these ideas and this gave us an indication of what the client expects. Weekly, we met and discussed the progress with the supervisor and the client. And each week we saw the product evolve into what we have created now.

We are very proud of the work we have delivered. Considering the circumstances of the COVID-19 period, we have tried doing the best we can. In other circumstances, we would have been able to research and evaluate the game we have developed more scientifically, by testing the game in person with a group of kids after parents giving written consent to do so. Moreover, we would have been able to meet in person which greatly improves discussing ideas, since you can draw them, explain through examples, show with body language, and discuss these ideas in parallel.

We would like to thank the anonymous students from the study Computer Science and Engineering at the TU Delft and anonymous experts on education for giving us feedback so that we could improve the game. We would also like to thank Melissa and Alex for providing us with resources, insight, and information needed to develop the game. Special thanks go to Shirley de Wit, for allowing us to do this project, giving us weekly feedback on the progress of the game, and handling meetings with Interxion and experts. Without her help, the product would not be as it is. We are grateful to Prof. Brinkman for guiding us through this project, providing helpful pointers, meeting with us on a weekly basis, and being an insightful supervisor.

Table of Contents

1. Introduction

Technology is a rapidly growing field, with an increasing need for engineers and scientists. Unfortunately, there is a global phenomenon of the underrepresentation of women in science, technology, engineering, and mathematics (STEM) studies and careers (Booy, 2012). This is also called a “leaking STEM pipeline”, where the interest in STEM declines throughout education.

The main client of this project, VHTO, the Dutch national expert organization on girls/women and science/technology, organizes classes on the areas of Science, Technology, Engineering, and Mathematics (STEM) for children between the age of 10 and 14. “The mission of VHTO is to increase the participation of girls in STEM fields. VHTO does this by tackling the problems of gender-stereotypes, vague views on the possibilities in STEM, a lack of female role models, and people being unconfident in their talents for STEM” (VHTO, 2020).

Additionally, Interxion is also a client for this project. Interxion is a company that provides carrier and cloud-neutral colocation data center services. Apart from providing consistent design, operational excellence, aid in market share, and communities of interest where you can interconnect, Interxion also wants to be more familiar to people, because of the shortage in employees. This is why they have partnered up with VHTO to create the project. In order to aid the goals of both Interxion and VHTO, the idea of an educational game got expanded to an educational game about data centers of Interxion.

For this project, VHTO and Interxion has come up with the idea to create an educational game for children between 10 and 14 years old, to aid in their mission. The goal for this game from the perspective of VHTO is to stimulate a positive attitude towards STEM and also teach about an area in STEM. This educational game will be used in a summer camp organized by VHTO and will also be used in educational programs Interxion organizes.

The game DataFlex was created as an educational game, about data centers, for children between the age of 10 and 14. The game is designed to show the diversity of gender in a data center. Moreover, women are positioned in different roles in the data center. With these aspects, the game tackles the problems mentioned above.

1.1. Outline

The report follows the development process of DataFlex. To create a product that satisfies the clients' needs, the actual problem that had to be solved was first defined, followed by a literature study (see **Appendix J: Research Report**) on how this could potentially be solved. Requirements were derived from this study and a development plan, as well as a design for the final game, was created based on these requirements. The stages of the development process can be found in chapters: 2: Problem definition and Analysis, 3: Design and 4: Implementation respectively. The game was implemented based on this initial design over the course of a few weeks, in which testing routines and various evaluations by the Software Improvement Group (SIG) were used to improve the maintainability of the product. Finally, the entire process and product were evaluated and a list of future improvements was compiled, which can be found in chapter 6: Conclusion and discussion.

2. Problem definition and Analysis

In this section, a definition and analysis of the problem are given, followed by the requirements of the final product. The problem definition is divided in two parts, the problem definition of the clients and the problem definition of the developers.

2.1. Problem definition

As stated by the clients, the game has two main goals, namely teaching children about data centers and creating a positive attitude about technology. As mentioned in chapter 1. Introduction, Interxion would like for their data centers to be more familiar to people. According to Dutch Data Center Association (2019), which has Interxion as a member, there is a shortage of technical employees for data centers. To fill up this shortage, Interxion tries to focus on a diverse group. In this project, Interxion focuses on young children to tackle this problem. In general, they invest in educational programs to guide school students to develop an interest in data centers. However, before building an interest in data centers, one would need to know what a data center is, what occupations there are at a data center, and why we need data centers. Educational games have proven to be a great tool in creating interest in, as well as changing attitudes towards certain topics (Prensky, 2001).

As mentioned in chapter 1. Introduction, VHTO's mission is to increase the participation of girls in STEM fields. The societal problem of this mission is that few women are involved in STEM, which leads to stereotypes and fewer female role models, thereby discouraging girls from a young age. The STEM fields seem to be more oriented towards men and lack diversity (Booy, 2012). The solution to the problem is to educate girls on STEM and create a more positive attitude towards those fields. This could lead to women being more interested in STEM, study STEM, and get a job in STEM.

To tackle the problems of the clients, an engaging educational game about data centers has been developed, in such a way that the game is fun and diverse in people. The game's goal is to teach children about data centers and to aid in creating a positive attitude towards IT. The target audience for the game is girls around the age of 12, in pre-vocational secondary education (VMBO).

2.2. Problem analysis

As mentioned in 2.1, the game has two main goals, namely teaching children about data centers and creating a positive attitude about technology. Since technology is too broad, the following refinement has been made: creating a positive attitude about IT.

Due to the fact that the game must be an educational game, the game has to be able to teach the players something. The goal "to educate end-users on data centers", is a broad goal. This is because there are a lot of things one can know about data centers. Data centers are the basis

of the digital economy and form part of the foundation of the internet. They are crucial to many aspects of our lives and applications that we depend on, ranging from Google maps all the way to electric cars. Essentially everything that happens online is in a data center. Therefore, in the requirements, which will be further elaborated on in section 2.3, there will be more achievable learning goals.

It is important to know what game best works for the goals of this project, because a game that focuses solely on memorizing facts in the form of a quiz, would not be engaging. On top of that, such a game would not aid in showing girls the diversity in data centers. So, the game should allow for an interaction with a data center. Research was conducted to find out what best works for this particular project. In **Appendix J**, an in-depth research report on an optimal choice for the game type can be found. In this report, an analysis can be found of what kind of game can help achieve the main goals.

The children will be playing a game that also teaches them about data centers. To improve the player's learning outcomes, the game should foster an appropriate attitude. According to Sentance, Barendsen & Schulte (2018), the learning outcomes centers on the player's attitude to whether they think they can play it. This is called a growth-Mindset. A growth mindset can be achieved by appropriate feedback and a feeling of success (Sentance, Barendsen & Schulte, 2018). When not stimulating a growth mindset, a fixed mindset can be created. Fixed mindset players tend to value the performance, where the player will only focus on finishing the game instead of enjoying it and learning from it.

2.3. Requirements

From the above, it follows that there are two main objectives for the project, namely educating children about data centers and creating a positive attitude towards IT. For each objective, requirements were set up in order to address them. These requirements, including the non-functional requirements, have been compiled into the following table (figure 1) and will be further elaborated on below.

Educate about data centers
The game must be fun
The game must be educational
The game must be challenging
Create a positive attitude towards IT
The game must provide a positive experience for the players
The game must break stereotypes regarding STEM - Women must be displayed in important roles - There must be an equal division of male and female characters - Co-operative rather than competitive aspects should be emphasized
The game must be gender-inclusive - Ensure that the diversity of science is represented to the largest extent possible - Ensure that there is a balanced approach to learning preferences
Non-functional requirements
The game must be playable within an hour
The game must be in Dutch
The game must be deployable to and playable on a low-end machine
The game must be extendable by the client
The game must have at least 70% of meaningful code coverage by automated tests

Figure 1: A table of the requirements

2.3.1. Educate about data centers

The first main objective is that children should be aware of what data centers are, what occurs in data centers, and why they are important for the future. Thus, the main requirement for this objective is that the game should explain the workings and importance of data centers. In order to accomplish this, the following requirements have been set up.

Fun

The feeling of engagement and the experienced fun of the players is one of the most important aspects of an educational game. The players are more likely to be interested if they have a very positive emotional connection to the game (Levine & Pizarro, 2004). Thus, one of the requirements for the game is that it has to invoke feelings of entertainment in the target audience.

Educational

The learning aspect of the game should be clear and measurable. This can be achieved by making use of learning goals. These learning goals can be constructed with the Bloom model (figure 2). Bloom's taxonomy makes it easier to define learning objectives using the verb chart. These learning objectives are measurable, as can be seen in **Appendix D: Learning Goals**.

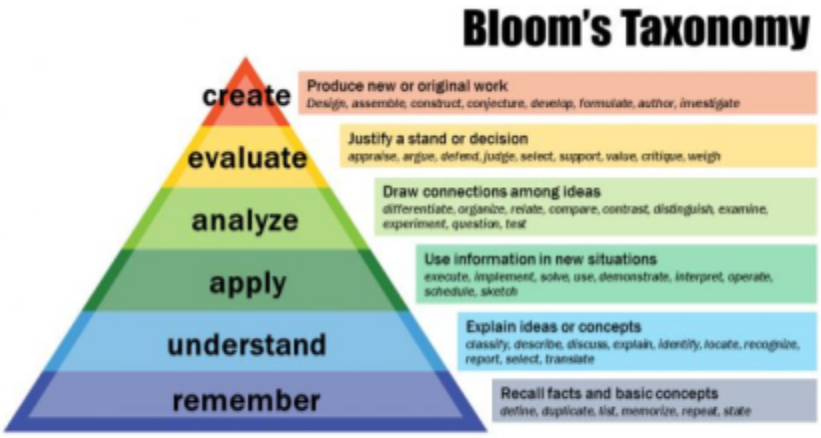


Figure 2: Bloom's Taxonomy
Retrieved from: https://cft.vanderbilt.edu/guides-sub-pages/blooms-taxonomy/

As for the motivation behind the learning goals, these were based on the requirement that the game must be educational. In the research report (see **Appendix J: Research Report**), various concepts of data centers have been studied. Because the target group of the product is relatively young (10-14 years old), subjects were chosen that are not too complex. Additionally, the game is meant to raise interest in IT, rather than teaching as much as possible in a short amount of time. This led to the learning goals being set in the lower parts of the Bloom Taxonomy (figure 2). Higher levels of the taxonomy are dependent on prerequisite knowledge of lower levels, meaning that each level is built on the foundation of the levels below.

Some learning goals are focused on the concept of a data center, while others may be focused on broader subjects within IT. The learning goal: *I can explain the constraints that influence password strength* for example has been constructed to teach an important security aspect in IT, which can also hold real-life value for the players. Some learning goals try to present the

diversity of science and societal relevance. All learning goals can be found in **Appendix D: Learning Goals**.

Challenging

To continuously engage with the game, players need the game to be fun but also challenging. “A game or activity can quickly lose its draw when a child is no longer challenged” (Hello Motherhood, 2018). Thus, in order to stimulate the student, the game should also be challenging. As the target audience for the game is children between the age of ten and fourteen in pre-vocational education, care should be taken that it is not too complex for the target audience.

2.3.2. Create a positive attitude towards IT

The second main objective is to create a positive attitude about IT (specifically data centers) with the goal of increasing gender diversity in STEM fields including computer science. The focus should be on girls in prevocational education to stimulate them to pursue a career in STEM, and more specifically a career involving data centers. The following requirements have been set up to best address this.

Positive experience

Players tend to identify themselves with the characters, meaning that success and failure are mapped directly to the player's ego and self-image in the experience (Lieberman, 2006). Thus, allowing for heavy punishment from failures, for example, can have a negative effect on the player’s attitude. This could negatively impact the objective of creating a positive attitude about technology. For this reason, it is also required that every player should come out with a positive experience.

Breaking Stereotypes

Harmful stereotypes can lead to women not choosing STEM fields (Meyer et al., 2015) and can lead to a decreased demand for women in STEM (Reuben et al., 2014). They may be impacting the consciousness of women, through the media, starting from a young age (Ter Bogt et al., 2010) Additionally, exposure to negative stereotypic content, such as in games, can reinforce attitudes that might lead to discriminatory and prejudicial attitudes (Brenick et al., 2007). Thus, breaking stereotypes is an important way of stimulating girls to pursue a career in STEM.

In order to do this, research has been done on particularly harmful stereotypes. Based on this, sub requirements were established that could be addressed by the design of the game. Particularly harmful stereotypes as portrayed in the media are their portrayal in more supportive roles in the laboratory rather than as leaders, and the presenting of science as requiring masculine traits (Steinke, 2016). Thus, these stereotypes should be avoided and challenged. This last point can be particularly harmful, as the physical sciences are often associated with

stereotypically masculine traits such as aggressive and competitive behavior (Easlea, 1987). This is why the co-operative rather than competitive aspect should be highlighted. Finally, stereotypes associated with the association of masculinity with science can be further amplified by the visible numerical superiority of men in scientific fields (Nosek, 2009). To combat this, there should be an equal division of male and female characters within the game.

Gender inclusion

The target audience for the game will be both male and female children. As such, the game should target both genders. However, in order to pursue the objective of stimulating girls to pursue a career in STEM, special care should be taken for this audience. When attempting to bring girls into STEM, it should be explicit that the end-users are girls (Sammet & Kekelis, 2016). An important part of this is gender inclusion. In order to assess and design for gender inclusion, the gender inclusion criteria by Hypatia Project, a project which attempts to engage teenagers in STEM fields, was used (Achiam & Holmegaard, 2016). This framework of gender inclusion criteria can be used to assess the gender inclusiveness of science education activities. The gender inclusion criteria used are the following:

- *"Ensure that the diversity of science is represented to the largest extent possible in the activity".*

Since there will be dealt with a diverse audience, the diversity of the subject matter should be presented.

- *"Ensure that the activity has a balanced approach to participants' learning preferences".*

Since everybody has a different way of learning, the solution should take a diverse method of teaching into account.

These criteria functioned as sub requirements and were used during the design of the game in order to ensure gender inclusivity.

2.3.3 Non-functional requirements

In addition to these main objectives, together with the client, expectations and requirements were set up that the product should also adhere to. The game is intended to be deployed at summer schools. Since not every school has access to high-end computers, the game has to be playable on low-end machines as well. Generally, it has to be easily installable or preferably playable in a web browser and its performance should be optimized. The game will be played by students in a classroom for the duration of one class hour. Because of this, it should also provide enough material to be played for one hour. And, as it will be deployed in a Dutch learning environment, the game should be presented in Dutch. Finally, the game will be extended by the client. Thus, it should be designed in such a way that it can easily be extended.

To make sure the game behaves as expected, the team came up with another non-functional requirement: at least 70% of meaningful code coverage should be achieved through automated

unit testing. Meaningful in this case means that 70% of the game logic must be covered. High level testing will be used for the UI and game flow. This will make sure that the game behaves as intended and enables faster debugging in the case that a bug occurs.

3. Design

As was seen in the previous chapter, numerous requirements and objectives were set up that had to be fulfilled. Research was conducted on the most optimal way to accomplish these objectives, after which a game was designed on the basis of this research. The final design of DataFlex is a 2D multiplayer RPG (role-playing game) with puzzle elements. In the game, players can follow a storyline by freely walking around a data center and solving various puzzles that come in the form of minigames. In this chapter, the main objectives and their corresponding requirements will be listed, along with how these requirements were addressed by the design.

3.1. Educate about data centers

As was stated, the first main objective was to educate children about data centers. To this end, the requirements were that the game had to be fun, challenging, and educational.

Fun

The main objective of the game is to increase the interest of girls between the ages of 10 and 14 in IT, therefore extra attention was paid to the fact that the game caters to girls. However, both genders will be playing the game, which is why DataFlex contains various game elements to grab the attention of both boys and girls. DataFlex can be described as an open-world role-playing game that contains various minigames. Research has found that the explorative and adventure elements that are introduced by the open-world part of the game are preferred by both genders (GameTree, 2019; Hartmann & Klimmt, 2006; Kinzie & Joseph, 2008). This is why DataFlex enables the players to walk around freely in the data center. Story and role-playing elements, although more preferred by girls than boys, are also strongly preferred by children of both genders over most other game elements (Kinzie & Joseph, 2008; Tekofsky et al., 2017). These role-playing elements are provided by the fact that the players can choose their character skins and can progress through the story by playing the role as professional engineers. These elements ensure that the main game is enjoyable for both boys and girls. The last element of DataFlex, the problem-solving element that is introduced with these minigames, shows no bias in preference between boys or girls (Kinzie & Joseph, 2008). These elements help to create a game that is fun for both boys and girls between the ages of 10 and 14.

Educational

The task was given to create an educational game to educate users on data centers. In order to efficiently and discretely accomplish this, clear learning objectives were established according to the taxonomy of BLOOM (Bloom, 1956). The usage of minigames enables for an effective grouping of learning goals about the same subject and offers the opportUnity to apply different learning strategies for each of these groups. With educational games, it is important to clearly link back to the concept that should be explained (Sentance, Barendsen & Schulte, 2019). This is why minigames were used; in order to clearly link a specific learning goal to a particular

activity. An overview of how each learning goal is incorporated by a specific minigame is specified in section 3.7.

Challenging

The player progresses through DataFlex by completing minigames. Thus, the challenge of the game is based on the complexity of the minigames. For each minigame, the complexity and how this can be varied will be discussed for every specific minigame in section 3.7.

3.2. Create a positive attitude towards IT

As was established in the requirements section (see 2.3), the second main objective was to create a positive attitude towards IT. In order to accomplish this, it was established that every player should have a positive experience. An important aspect of this objective is to increase the interest of young girls in IT. Thus, it is crucial that the design breaks stereotypes and is gender inclusive.

Positive experience

As was stated, it should be ensured that every player has a positive experience at the end of the game. As such, it should be avoided that players get stuck or frustrated. Because of this, the minigames and story progression were designed such that losing is not possible. The story is progressed by completing minigames and for all these minigames, it is not possible to fail when the end is reached. For minigames which only progress through player activity, hints and guidance are provided in order to assist the players. Finally, each minigame is preceded by a tutorial that informs the player on how to progress. Further discussion on each minigame will follow later in section 3.7.

When the players get stuck in the game positive feedback with suggestive words leads to players understanding the mistake and have the motivation to continue on (van der Wal & de Wilde, 2017).

Breaking stereotypes

As one of the requirements was to break stereotypes, care was taken to design the game in such a way that it would not propagate harmful stereotypes. To this end, a selection of particularly harmful stereotypes was researched, compiled and sub requirements were set up in order to best break these stereotypes (see figure 1). The first sub requirement was that the highlighting of women's femininity over their technical expertise had to be avoided. In light of this, the characters were designed to be as diverse and non-stereotypical as possible.

As one of the requirements was to break stereotypes, care was taken to design the game in such a way that it would not propagate harmful stereotypes. To this end, a selection of particularly harmful stereotypes was researched, compiled and sub requirements were set up in

order to best break these stereotypes (see figure 1). The first sub requirement was that the women had to be displayed in important roles. Thus, the female characters included in the game were designed to have important and crucial roles such as specialist, manager and engineer. The second sub requirement was that there had to be an equal division of male and female characters. To address this, the game was designed in such a way that there would be an equal division of male and female characters. An overview of the gender and profession of each character can be found in figure 4. Finally, the co-operative rather than competitive aspect had to be emphasized. To this end, the minigames were designed such that the players would not have to compete with each other. All the objectives for the mini games are achieved through co-operative gameplay. DataFlex avoids keeping track of individual scores that could encourage competitive behavior.

Gender inclusion

As was previously stated, a framework of gender inclusion criteria by the Hypatia project was used in order to manage gender inclusion. These were then compiled into sub requirements (see figure 1).

The first sub requirement was that the diversity of the subject matter (in this case, data centers) should be represented to the largest extent possible in the activity. This is one reason why DataFlex is composed of multiple minigames: they provide a powerful mechanism to teach a wide range of concepts. Thus, the minigames were designed in such a way that DataFlex could educate not only on the technical aspects of data centers but also in ways that could show the societal relevance. The second sub requirement was that the activity has a balanced approach to participants' learning preferences. Yet again, the minigames allow differing learning strategies to be accommodated, as each minigame can contain different game mechanics. This allows engagement with the subject matter through various mechanisms.

Girl centered design

DataFlex was designed in a way that it could be approachable to girls, by taking into account how girls learn certain subjects and how girls can become more interested in STEM subjects. The interest of girls lies in contexts such as health, the human body, medicine, ethics, and aesthetics (Sjøberg & Schreiner, 2012). Yet again, the minigames allow various aspects of data centers to be highlighted. In order to make it more approachable to girls, the minigames also feature environmental and social aspects. Another aspect which has to be paid attention to for girl-centric design, is competition. Women tend to shy away from competition while men embrace it (Niederle & Vesterlund, 2010). This is yet another reason why DataFlex is multiplayer and encourages co-operative play.

3.3. Non-functional requirements

The requirement that the game should provide enough content to be played for one hour was addressed through the complexity of the minigames. Namely, the complexity of each minigame can be varied in order to fine-tune the completion time of the game. Additionally, the game was designed to be playable in the Dutch language. The rest of the non-functional requirements will be discussed in chapter 4. Implementation.

3.4. Game flow and plot

The user flow is shown in the flowchart in figure 3. At the start of the game, the main menu is shown to the players. From there, players can start the game. When the players click on the button, they will be greeted by a tutorial. This tutorial explains the controls and the context of the game. After finishing this tutorial, the players can choose a character and start playing the game. The two characters walk together throughout the game and engage in dialogues, which requires the interaction of at least one player.

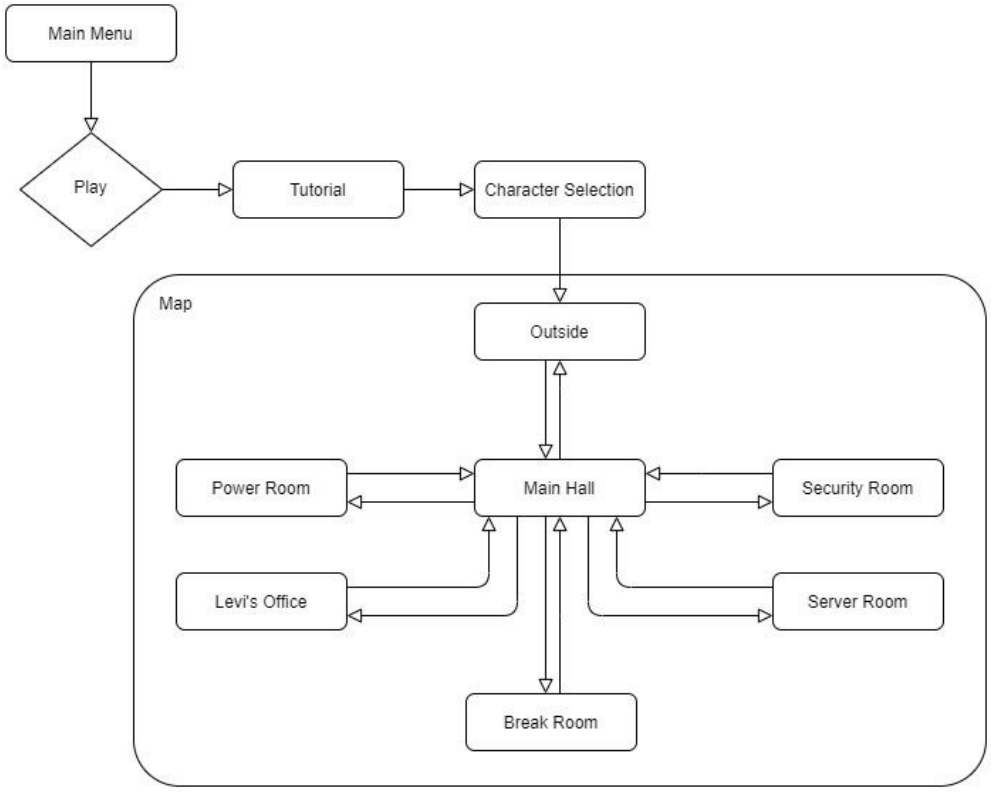


Figure 3: A flow chart of the game

Plot

The players are two engineers in an Interxion data center. When they arrive at the data center, they will be told that an intruder caused chaos in the data center. The goal of the players is to find the intruder by completing various tasks. These tasks are represented by minigames (see

3.7). Every minigame is invoked by a character that is present in the different rooms. The layout of the rooms and the roles of the corresponding characters are shown in figure 4.

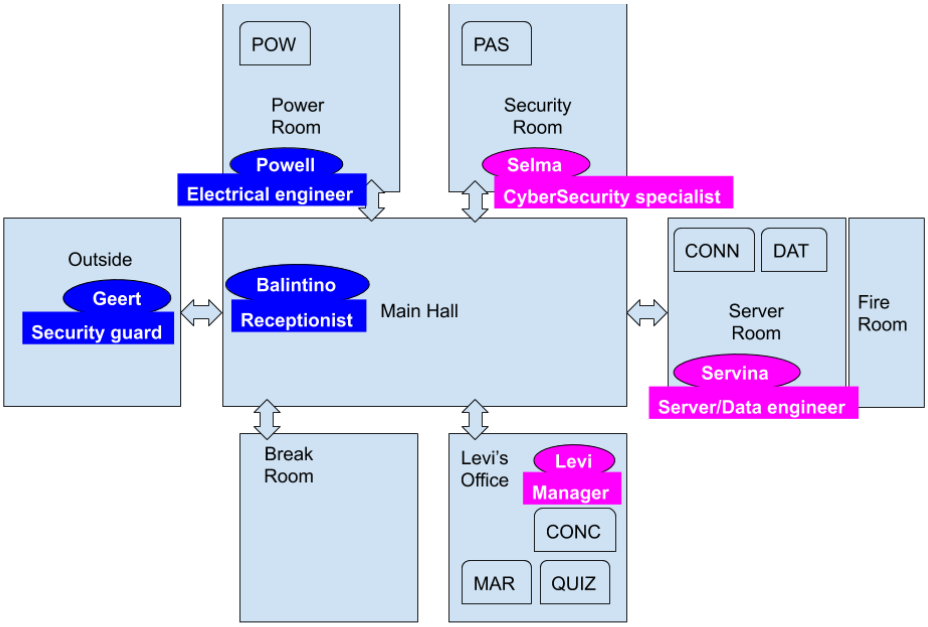


Figure 4: An overview of the map, minigames and in-game characters

At first, the concepts game (see 3.7.1) needs to be played. After completing the concepts game, the players need to turn on the lights in the building. This is done by the Power eco-friendly minigame (3.7.3). After these two games, the players can complete the other minigames in any given order. When the players finish the minigames, they report back and are asked to finish a quiz (3.7.7).

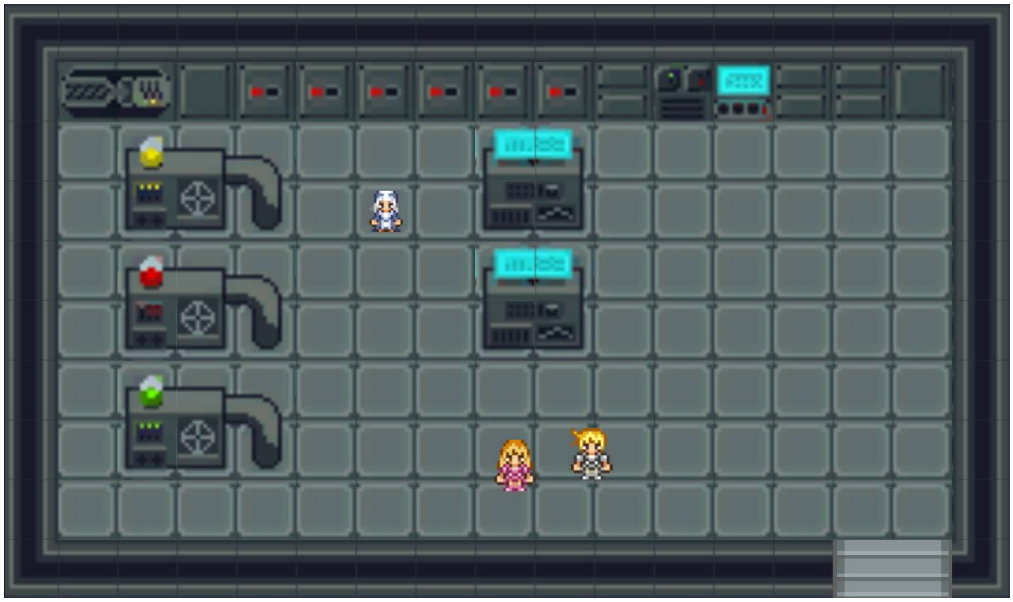


Figure 5: Two characters walking in a room

3.5. Art

The art style of a game can help to create a more positive impression of the game itself (Gerling et al., 2013). The consistency of the art can also help to create an immersive feeling, whereas color can help convey a certain mood (Masuch & Röber, 2004). The use of bright colors is useful since this can help to create a positive attitude towards IT-related subjects (Terwogt & Hoeksma, 1995). All these effects that the art style has on the game experience had to be taken into account while developing DataFlex. Due to the lack of time and expertise regarding graphical design, the actual creation of the art for DataFlex was outsourced.

As was mentioned in section 3.2, it is important that the non-playable characters (NPCs) within the data center, as well as the characters that can be chosen by the players, have an equal distribution in gender, as well as ethnicity. The interior and exterior of the data center should also closely resemble the Interxion data centers for which images and a blueprint were provided by the client.

3.6. Dialogues

DataFlex is mainly a role-playing game, where the players are engineers in a data center. To create the story elements that belong to this genre, a flexible dialogue system has been created (4.2.2). With the help of this dialogue system, players can interact with various non-playable characters (NPCs) or objects. This offers the ability to not only enhance the explorative aspects of the game but also elaborate even further on the importance of certain activities within a data center. Powell (an NPC within the power room) can, for example, explain through a dialogue that it is important for a data center to have backup generators in case of a power outage. In addition, Interxion also has the ability to provide the players with statistical information, specific

to their data centers, through this dialogue system. Information that might be difficult to provide through a minigame can in this case still be provided through the dialogue system, which makes it easier to achieve certain learning objectives.

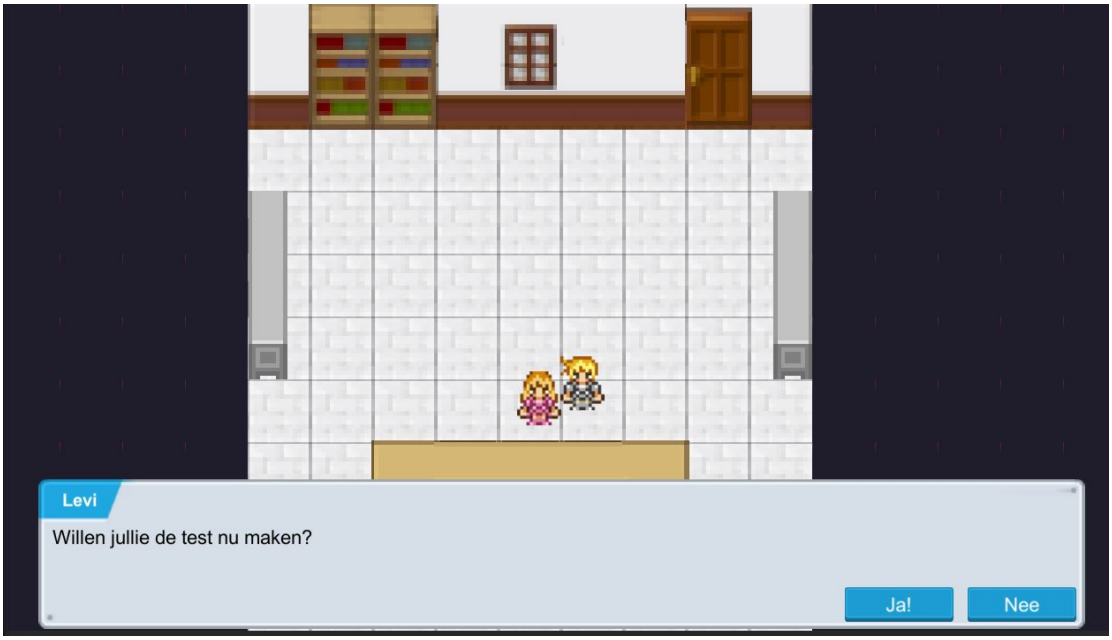


Figure 5: The dialogue window displayed

3.7. Minigames

One of the main purposes of the project was to educate the players on data centers. Because of this, clear learning goals were developed using Bloom's taxonomy (figure 2). Bloom's taxonomy makes it easier to define learning objectives using the verb chart. These learning objectives are measurable, which in this case, is done by the quiz game (3.7.7). These learning goals are the foundation of the various minigames. To this end, before each minigame, a slide is shown which explains the concept to be taught. Also, as was established in the requirements (2.3), it is important that the diversity of the subject matter is represented to the largest extent possible in the activity. The multiple minigames provide a powerful mechanism to teach a wide range of concepts. The length and difficulty of each minigame can be adapted based on user feedback. However, the way this is achieved differs per minigame and will, therefore, be explained in their respective sections.

3.7.1. Concepts game

The concepts game is the first minigame, which the players need to complete before playing the other minigames. This game is meant to introduce the subject matter; the basics of a data center, before playing the story. The goal of the game is to combine pairs of items until the last item is unlocked: the cloud.

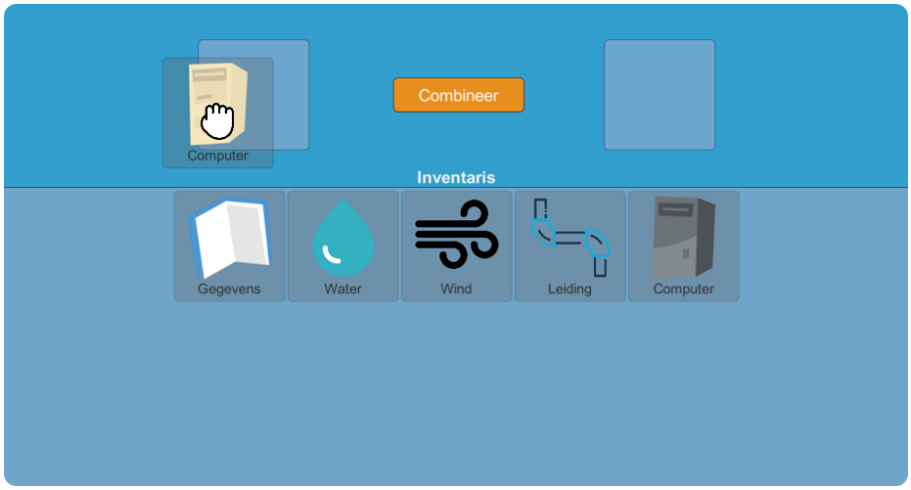


Figure 6: The concepts game; the player is dragging an item

The following elements form the basis of the game: a combine button, two crafting slots, an inventory, and the items (as seen in figure 6). The players start with a few items in the inventory. These items can be put in the two crafting slots and combined using the combine button. The result of the combined items could be either a new item, no item, or a hint.

Whenever the result of a combination is a new item, the players are presented with an unlock screen. This unlock screen explains the unlocked item and the players can not return to the game for two seconds (in order to read the explanation). After this, the item appears in the inventory, to combine with it the other items.

When the players combine two elements which form a hint, the players will be shown a textual hint. This is created in order to prevent players from getting stuck and ensuring that every player can finish the game. The hierarchy of combinations is shown in figure 7.

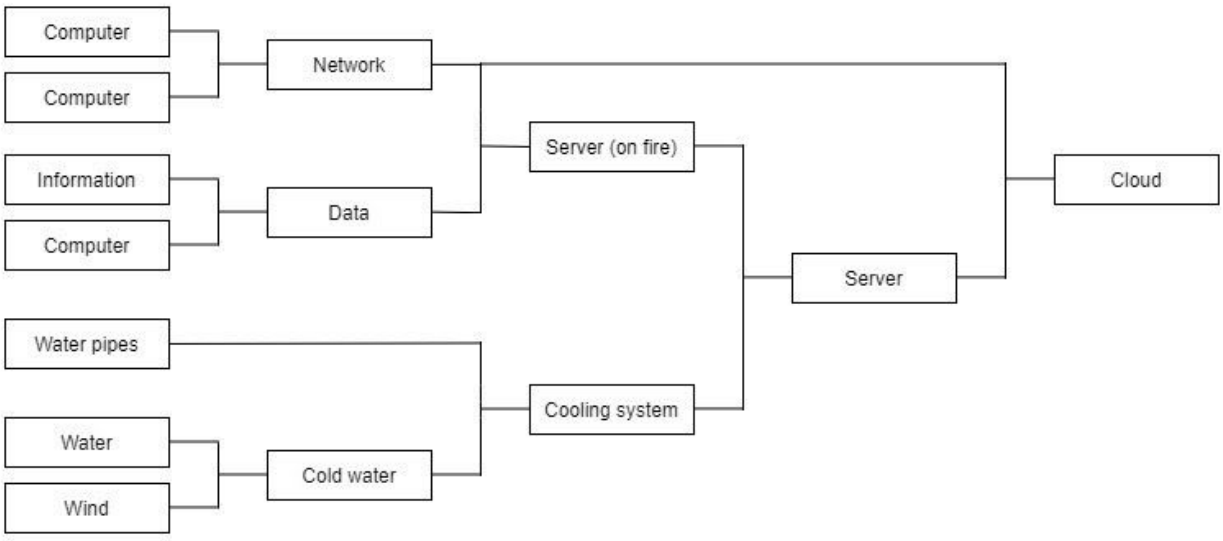


Figure 7: Hierarchy of the concepts game

The learning goals that are associated with this game are the following:

I can recognize elements a data center is composed of
The game approaches this goal by introducing different elements that form the basis of a data center. The game explains important aspects of a data center, such as a server, a cooling system, data, and the cloud.

I can differentiate different elements of a data center
The information in the game is broken into different parts, which the players are expected to organize or to find a connection between. This makes it clearer to the players that the parts are connected to each other.

I can define a data center as the home of the cloud
The main goal of the game is to create the cloud. The players understand what components make the cloud possible and remember the definition of the cloud because this will be explained at the end of the game.

3.7.2. Connection game

The connection game is a puzzle game with a goal of connecting all servers to each other and the power outlet. The players start with a board with random generated edges and vertices. These edges can hold the shapes as shown in figure 8.

Figure 8: Shape edges in the connection game

In addition to these shapes, there are also vertices, which are either a server or a power outlet. Every edge and vertex can be rotated. The players succeed whenever there is a spanning tree that includes all vertices and edges, as shown in figure 9. Here, the red squares represent the servers and the yellow square represents the power outlet. In order to determine which difficulty will fit the game, the board dimensions could be changed.

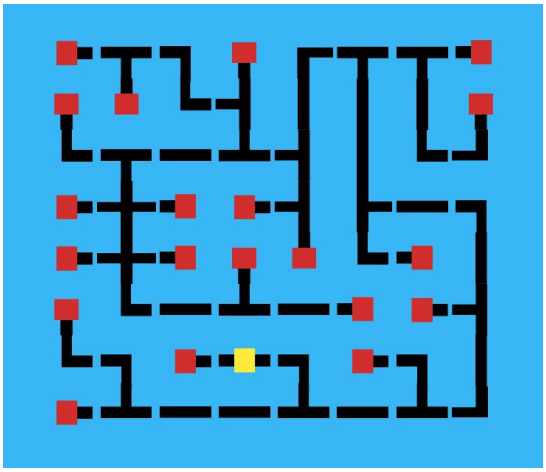


Figure 9: A completed connection game

The following learning goals are associated with this game:

I observe that individual servers are part of a whole system
The connection game introduces the servers as fundamental parts of the whole system. It implies that the system does not work whenever all the individual servers are not interconnected.

I understand that servers are an integral part of data centers
The connection game implies that the data center can not function, without working/connected servers. The tutorial also explains that these servers are essential for the data center.

I can solve a mathematical puzzle problem
The game is mathematical because its goal is to find a spanning tree, which is a mathematical problem in the field of graph theory.

3.7.3. Power eco-friendly game

The power eco-friendly game is similar to the connection game in terms of mechanics. The players also start with a board of random generated edges and vertices. However, the vertices can be one of the three options: a sustainable energy source, a nonrenewable resource, or a data center. In addition to these vertices, there is an environmental impact constraint.

Whenever an energy source (either non-renewable or renewable) is connected to the data center, the environmental impact rises (see figure 10). This environmental impact becomes too high when the nonrenewable resources are connected. Thus, the players need to find a spanning tree, which only includes all the sustainable energy sources and the data center. To leave the players with a positive experience, it is important that this game is not too difficult for the target audience. Like the connection game (3.7.2), the dimensions of the board could be changed to make the game easier or more difficult.

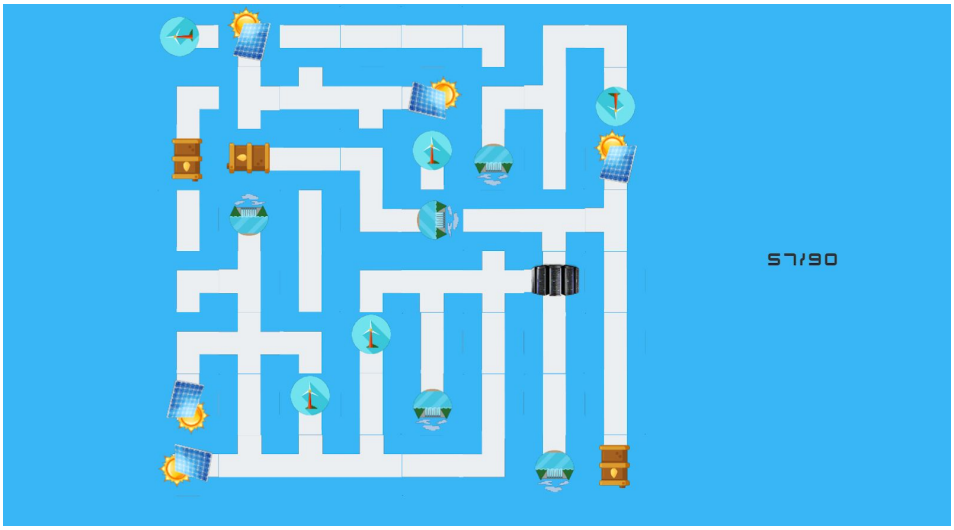


Figure 10: The power eco-friendly game

The following learning goals form the basis of this game:

I understand that power comes from different sources
The power eco-friendly game tries to teach this by introducing different icons of energy sources. This holds for non-renewable, as well as renewable energy sources.

I understand that sustainable energy resources have less environmental impact than nonrenewable resources
The game has an environmental impact score, which increases whenever nonrenewable energy sources are connected to the data center. This implies that non-sustainable energy causes more environmental impact than energy from renewable sources.

I remember that data centers (partly) run on sustainable energy
The game forces the players to only connect renewable energy sources. The fictional data center that is presented in the game, is therefore only connected to renewable energy sources.

3.7.4. Data priority game

The data priority game shows the players what kind of data can depend on data centers. Different types of data fall out of the sky, and the task of the players is to catch this data. However, there are also ‘bad’ types of data (hackers and viruses for example). More valuable data is also worth more points, this includes hospital and police data. The game ends whenever the timer runs out. There is no minimum score, instead, a message appears at the end of the game that is dependent on the number of points gained throughout the minigame. This is to ensure that players finish the game. The learning goals that are associated with this game are the following:

I remember that data centers are fundamental to the Internet
The game shows the types of data as icons. These icons include concepts such as social media, the hospital, and the police. Players learn that this data is handled by data centers.

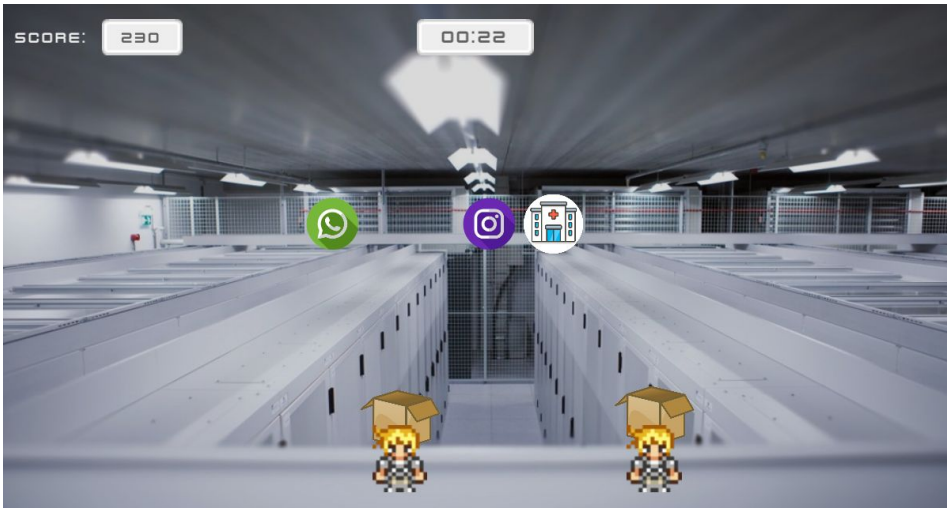


Figure 11: The data priority game: types of data that fall to the floor.

3.7.5. Password game

The goal of the password game is to teach players about the importance of security, which in this case relates to password strength; the effectiveness of passwords against brute-force attacks. The game consists of a list of passwords that the players need to make. Every password has its own constraints (figure 12). When the players submit a password that satisfies every constraint, they are presented with a message which informs them how long it takes to brute-force the password[1]. The players need to remember these passwords because they are required to repeat them at the end of the minigame in order to progress. Whenever the players forget the password, they will be given three attempts. If they fail to enter the right password, the created password appears and they will pass the stage.

[1] **Brute Force Calculator**, from: https://tmedweb.tulane.edu/content_open/bfcalc.php

Figure 12: The password game

The following learning goals were established for the password game:

I can explain the importance of security
As is explained around the game, security is an important element of a data center. This includes both virtual, as well as physical security (e.g. guards).

I can explain the constraints that influence password strength
The game introduces constraints at every stage that influence the password strength (capital letters, numbers, symbols, and length). This password strength will be visible whenever the players submit a password, by showing a message that explains the time needed to brute-force the password.

3.7.6. Marketing game

The marketing game is meant to show players the different sides of operating a data center. The goal of this game is to allow players to create slogans for multiple divisions of the data center.
The players are presented with a context and a codeword (figure 13), which should be included in the slogan (the text on the sticky note). After slogans have been created for every division, the game ends.

Figure 13: The marketing game, with the codeword on the sticky note

I remember that maintaining a data center requires expertise from diverse fields
The game introduces a different side of working in a data center, namely the marketing division. The players have to create slogans for this division, but the game itself also introduces other divisions, such as security and energy.

I can motivate why different aspects of a data center are important
The game explains that different aspects of the data center are also important to maintain the company. This is explained using the tutorial before the game.

3.7.7. Quiz game

In the end, to test the knowledge of all the players, a quiz game was created. The questions are all related to the learning goals of the minigames, which have been described before (for an overview, check **Appendix D: Learning Goals**). The quiz attempts to ensure that this measure is quantifiable.

The quiz is one of the two minigames (along with the concepts game) that can not be completed in an arbitrary order, as this game can only be started after having completed all the other games. One or more multiple-choice questions related to each learning goal are asked to the players, presenting them with four options (figure 14). For every question, there is one valid

answer. After going through all the questions, the player completes the game. There is no minimum score in order to pass the game.

Figure 14: The quiz with a score and four possible answers

4. Implementation

The product consists of different components. This chapter gives a global overview of the implementations of these components, why certain implementation choices were made and how these components can be used. However, due to the documentation of the scripts, no in-depth explanation of the scripts and their respective methods will be given.

4.1. Key concepts

Before the details of the implementation, there are a few key concepts that are part of the Unity game engine that will be explained. As was established in the requirements (see 2.3), the extensibility was a crucial aspect for the product. The prefabs and scriptable objects form the basis for the extendibility of the product.

4.1.1. GameObject and MonoBehaviour

In Unity, every game is run within one or multiple scenes, each containing their own game environment, menus, and variables within the game. In a scene, there is a hierarchy of GameObjects. These GameObjects represent the objects that are present within a scene. Thus, a script can only run if it is part of a GameObject entity. Every GameObject is a collection of components. A few examples of components are: transform (used to change the position and scale of GameObjects), button, image, and a script. This script component is also called a MonoBehaviour component.

4.1.2. Scriptable objects

Scriptable objects are data containers that can be saved as assets in the project folder. This type of object is used for entities that have the same properties which are widely used within the project. Creating scriptable objects makes it easier to manage changes for a large group of entities that all have the same behavior. For instance, the quiz game needs questions in order to work. These questions could be made with the use of a scriptable object. It makes it easier to adjust and create multiple questions.

4.1.3. Prefab

A prefab is a GameObject that is converted into a reusable asset. Prefabs are used to maintain and create multiple instances of GameObjects. For example, if one wants to create a button that appears multiple times within the game, one could make a prefab to prevent creating a new button for every instance.

4.1.4. Singleton

A singleton is a design pattern that is used to ensure that a class has only one instance and provides a global access point. This is useful in game design, since the logic of a class is often handled, independent of the GUI. A singleton allows the different classes to communicate via a single access point.

4.1.5. Game hierarchy

DataFlex is built with different scenes, each scene contains its own game objects and can be seen as a separate level. These scenes keep the game neat and tidy and enable the team to work on separate parts of the game at the same time. Every minigame contained within DataFlex has its own scene, as well as every room and menu. Every time a scene is loaded, the existing game objects are destroyed and new ones are created as specified in the specified scene. To change these scenes according to game progression, the singleton design pattern (4.1.4) is used. In addition to this, every minigame, except the quiz, will be introduced with a tutorial that exists of multiple slides.

4.2. Main game

The main game contains various elements, these elements were created in a way to optimize the extensibility and maintainability of the final product, since this was also a requirement specified in section 2.3. These elements and their specific design choices will be explained in the sections below.

4.2.1. Global game state

DataFlex follows a certain storyline. The progression through this story has to be stored, as well as some other information such as the chosen player skins or the minigames that have successfully been completed. To do so, a GameProgression script was created. This is a singleton that is created at the start of the game, which stores the following variables:

- The skins chosen by the players
- The index of the spawn gate where the player should spawn after the next scene transition (see section 4.2.3)
- The current game stage
- The number of times all NPCs have been updated
- A list of the current stages of the NPCs
- A list of the current stages of the minigames (completion)

This script only manages changes to the game, NPC, or minigame stages. To keep the GameProgression script neat and clear, the changes based on the NPC and minigame stages (such as visual scene changes) were moved to separate managers (for example, the dialogue

script looks up the NPC stage and only parses the necessary lines based on this stage). However, changes to the game stage are handled within the GameProgression script since these changes will often affect NPC stages as well. How the different managers make use of specific variables stored within the GameProgression script will be explained within their dedicated sections.

4.2.2. Dialogue

In order to introduce a story, a system had to be created to allow the player to advance through the stages. This is mainly done by allowing the player to interact with characters through the use of dialogue. Through this interaction, the player can understand the plot and receive tasks, which they can accept. While the game dialogue system could have been hardcoded and implemented straightforwardly, this would not have been very maintainable or flexible. One of the goals of the project is that it should be easily extendable by a future party that wishes to continue development. For this reason, a system was designed to allow the creation of new dialogue in a simple and flexible manner.

To keep the project organized, it was decided to store dialogues in .txt files. These files could then be added to an NPC or object. If a room contains an NPC or object with a dialogue, the dialogue prefab should be added to the scene. This contains the dialogue manager, as well as the canvas containing the dialogue box. NPCs and objects with a dialogue have a stage that is stored in the GameProgression class (4.2.1). The dialogue system consists of three scripts, DialogueTrigger, DialogueManager, and Dialogue. Their usage and the design choices will be explained below, together with the layout of the .txt files containing the dialogues.

Dialogue .txt files

Before creating the customized dialogue system, the requirements and limitations first had to be specified. The functionalities that were specified in this stage, were then translated to commands that could be used in the .txt files. The first necessity for the dialogue system was the ability to use pointers. Text for NPCs or objects might change, depending on the stage of the game. Dialogues should also be able to start minigames, which is done through a scene transition. Players should, however, have the freedom to decide which minigame they want to play at what point of the game. Therefore, choices had to be added to the dialogue system. The number of choices was limited to two in order to prevent the parsing of the commands from becoming unnecessarily complicated and keep the syntax simple. This choice functionality enables different paths to exist within a dialogue stage, therefore separate pointers within specific stages were added. Some learning goals might be achieved through dialogue, or some dialogue might be important to understand the story. Therefore, a command that disables the functionality to close the dialogue at any given moment was added. Lastly, the closing of the dialogues and updating of the NPC or object stage were added as dialogue commands for extra flexibility. An overview of the different commands and their specific syntax can be seen in **Appendix I: Dialogue .txt commands**.

DialogueTrigger

The DialogueTrigger component (figure 15) is added to an NPC or object to handle the interaction. This component contains a Dialogue object, which consists of three fields:

- The .txt file, which contains the desired dialogue
- The index of the spawn location indicated where the players should spawn after a minigame for example. (For more information on SpawnGates, see section 4.2.3)

Figure 15: Dialogue trigger component

The NPC or object with the DialogueTrigger component should also contain a collider that functions as a trigger (figure 16). As soon as one of the players enters this collider, the DialogueTrigger will freeze the players and open the DialogueBox. The DialogueTrigger will also store the name of its parent Unity object as the name field in the Dialogue script. This means that the name of the Unity object will be the name displayed within the dialogue box, as well as the entry for which the NPC or object stage is stored.

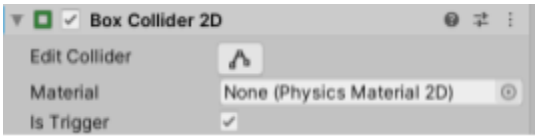


Figure 16: Box collider as trigger

Dialogue

The Dialogue script stores the necessary information for the current dialogue. It parses the specified .txt file and automatically stores only the sentences of the current stage of the NPC. This saves time as well as memory, since not everything is stored and not the whole file is parsed. It also enables easier access of pointers within a specific stage, compared to when all pointers for all different stages have to be stored. The first time a player interacts with an NPC, the Dialogue script will initialize this NPCs' stage based on the number of times every NPC has been updated (which is a variable within the GameProgression script).

DialogueManager

Once the Dialogue script stores only the lines necessary for the current NPC or object stage, the DialogueManager will parse these lines. The DialogueManager translates the text and commands specified in the .txt file to the desired UI elements of the dialogue box. This makes sure that the right button functionality is added if a command is parsed and that the dialogue box switches between either a choice or a regular dialogue box if necessary.

4.2.3. Scene transitions

Where the players spawn in a room depends on where they come from or what activity they have done before they spawn. For example, when the player walks from the power room to the

main hall, the players should spawn in front of the door to the power room. To achieve this, a SpawnGate script was created. This SpawnGate can be added as a component to a game object and contains an index, which should be determined by the developer, as well as two spawn locations, one for each player.

There are two use cases of these spawn gates. In the first case, it is used when the player exits or completes a minigame. In this case, a spawn gate prefab is added in the desired location in the room, as well as a reference to the index of this gate in the DialogueTrigger component of the NPC or object (see section 4.2.2. Dialogue).

The second case is when a scene transition should be triggered if both players walk in a specific area, for example, a door. In this case, a scene transition prefab can be added to the room. This prefab contains the same SpawnGate component as the spawn gate prefab does, as well as an additional SceneTransition component. This SceneTransition component contains the name of the scene that this transition should lead to, as well as the index of the spawn gate in that scene where the players should spawn. For example, when the players trigger the scene transition with the component (see figure 17), the players will spawn at the spawn gate with index 4 in the MainHall scene.

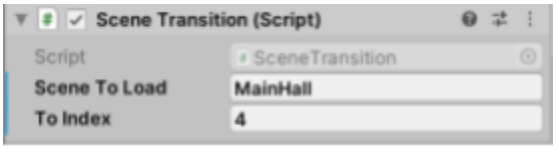


Figure 17: Scene transition component

4.2.4. Player movement

DataFlex is an open-world game in which the players are able to walk around freely. To enable this movement within the rooms, the PlayerMovement script was created. This script contains the animations that should be used when a player walks into a certain direction, as well as which input managers should be used for which character. To make sure undesired events would not be triggered while playing the game, player movement can be locked through this class (e.g. when the dialogue box is open). Two prefabs, PlayerOne and PlayerTwo, have been created and should be added to newly added room scenes.

4.2.5. Map design

The design of the game map was narrative-based. It was first determined what locations were needed to progress the narrative, these locations were then each assigned their own room. The design of the game map was also derived from an actual blueprint provided by the client. By using this blueprint, the rooms could be designed with a certain level of realism in mind. Every room was implemented in its own separate scene, this ensured that only a small portion of the game needed to be loaded to lower the memory footprint. The rooms were connected with scene transitions.

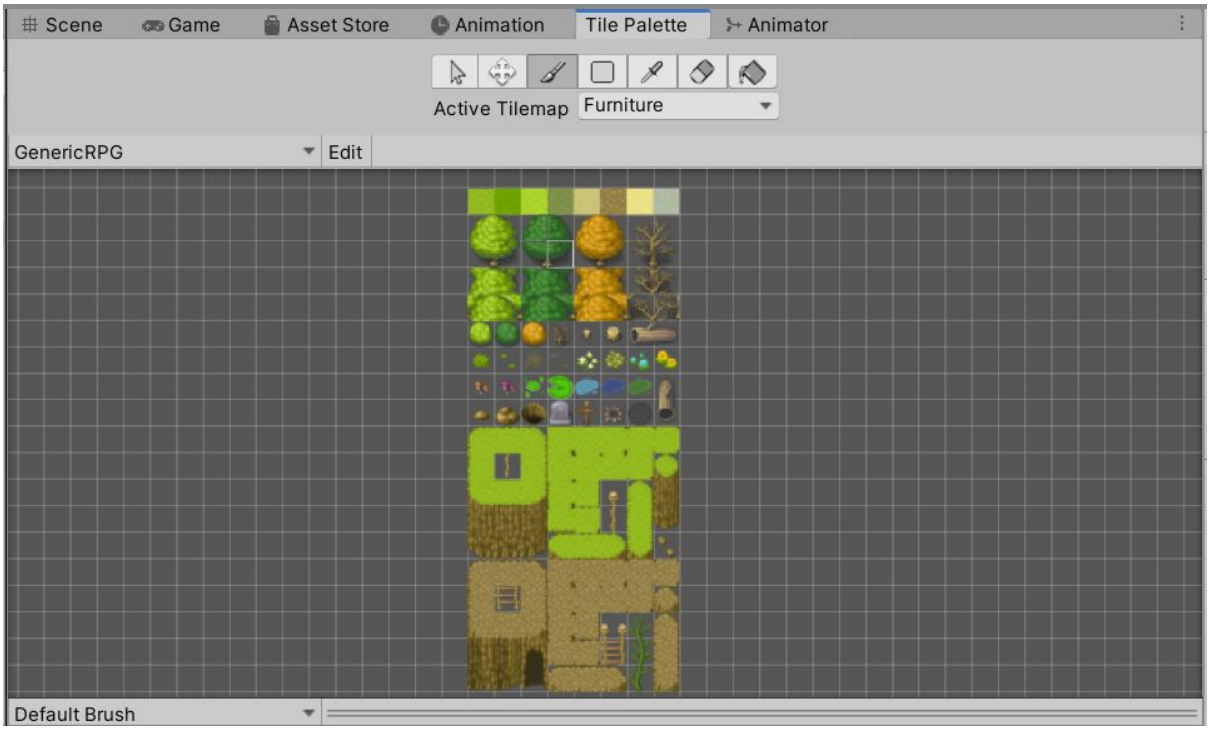


Figure 18: Tile Palette

The individual rooms were designed using TileMaps. TileMaps allows the designer to easily place Tile Assets on a Grid. These Tile Assets are always the same size, usually squares. This means that the sprites that are used to create these Tile Assets must be square too. By using the TilePalette (see figure 18), the designer can paint the required Tile Assets on the Grid to create a room. The Tile Assets can also be grouped into layers. This allows the designer to place some Tile Assets over others, like fences over grass (see figure 19), by changing the ordering of the layer. It also allows the designer to give some layers Colliders, like the walls and furniture, so that players can not move through them. These colliders can be identified in figure 19 by the green outlines.

From a performance point of view, TileMaps also has many advantages compared to using sprites. TileMaps drastically reduce the memory footprint (4.7.1) of the application. The overall file sizes are also decreased, and the loading and rendering times of the game improve dramatically leading to a higher FPS[2].

[2] **TileMaps**, a tool for creating 2D environments from: https://Unity.com/how-to/optimize-performance-2d-games-Unity-tilemap

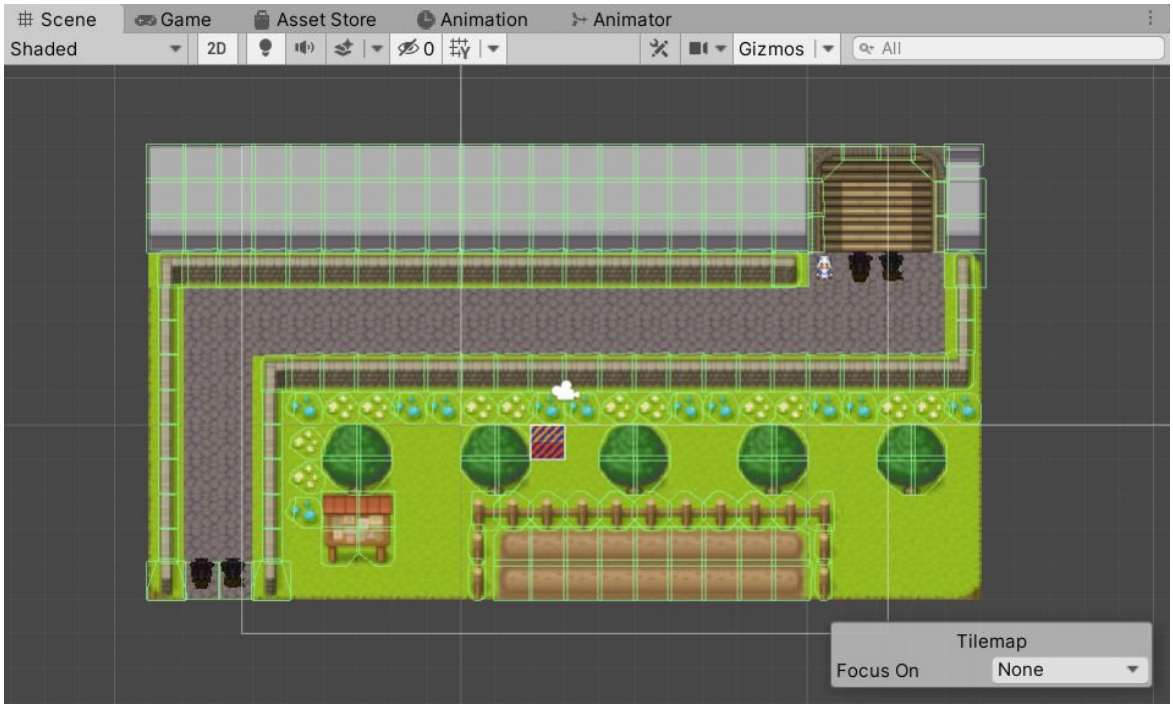


Figure 19: TileMap of the outside area

4.3. Minigames

DataFlex contains a variety of minigames. Due to the flexibility in scene transitions through the dialogue system and every minigame being contained within its own scene, new minigames can easily be added. How the currently implemented minigames work will be explained in their respective sections below.

4.3.1. Concepts game

The concepts game contains a lot of classes, but only the classes that handle the game logic are discussed in this section. A UML diagram of the game can be seen in figure 20. Two classes form the basis of the game:

Crafter

The crafter class handles all the game logic. All UI elements that are present in the game communicate through the crafter class. This is made possible by the singleton design pattern (see 4.1.4). The crafter as a singleton makes the UI interaction easier because UI elements can invoke certain methods.

Inventory

The inventory is a class that handles the items in the players' inventory. At first, a list of start items is loaded into the inventory. Whenever a new item is discovered by successfully following a recipe, the item is instantiated by the inventory class.

Scriptable object; items, recipes and hints

To make the game more extensible, scriptable objects (4.1.2) have been used to add new items, recipes and hints. The main goal of the game is to combine certain items, in order to unlock new items, until the player unlocks the cloud (see 3.7.1). These items are represented by an Item class which contains the following:

- Image: the sprite of the item
- Name: the name of the item
- UnlockText: text that explains the unlocked item to the player.

This item object could be made through the Unity editor by adding a new item object without coding. In order to create recipes for unlocking different items, a recipe object has been created. This recipe object is made of the following elements: two inputs (Items) and a result (Item). A recipe can also be made within the Unity editor.

The last scriptable object is the hint object. A hint is composed of:

- An input (Item)
- A list of second inputs (List of items)
- An end item (Item)
- A hint string (String)

The hint is supposed to have the option for multiple combinations, that is the reason why a list of second inputs was chosen as a second field. The end item is meant to be the item at which the hint is directed at. When this item is discovered, the hint is not supposed to be displayed. The hint string represents the text that will appear on screen.

By using these three scriptable objects, the game can easily be extended or reduced. The UML diagram in figure 20 shows all classes that handle the behaviour of the game. The UI entities are not shown completely, because this is dependent on the environment.

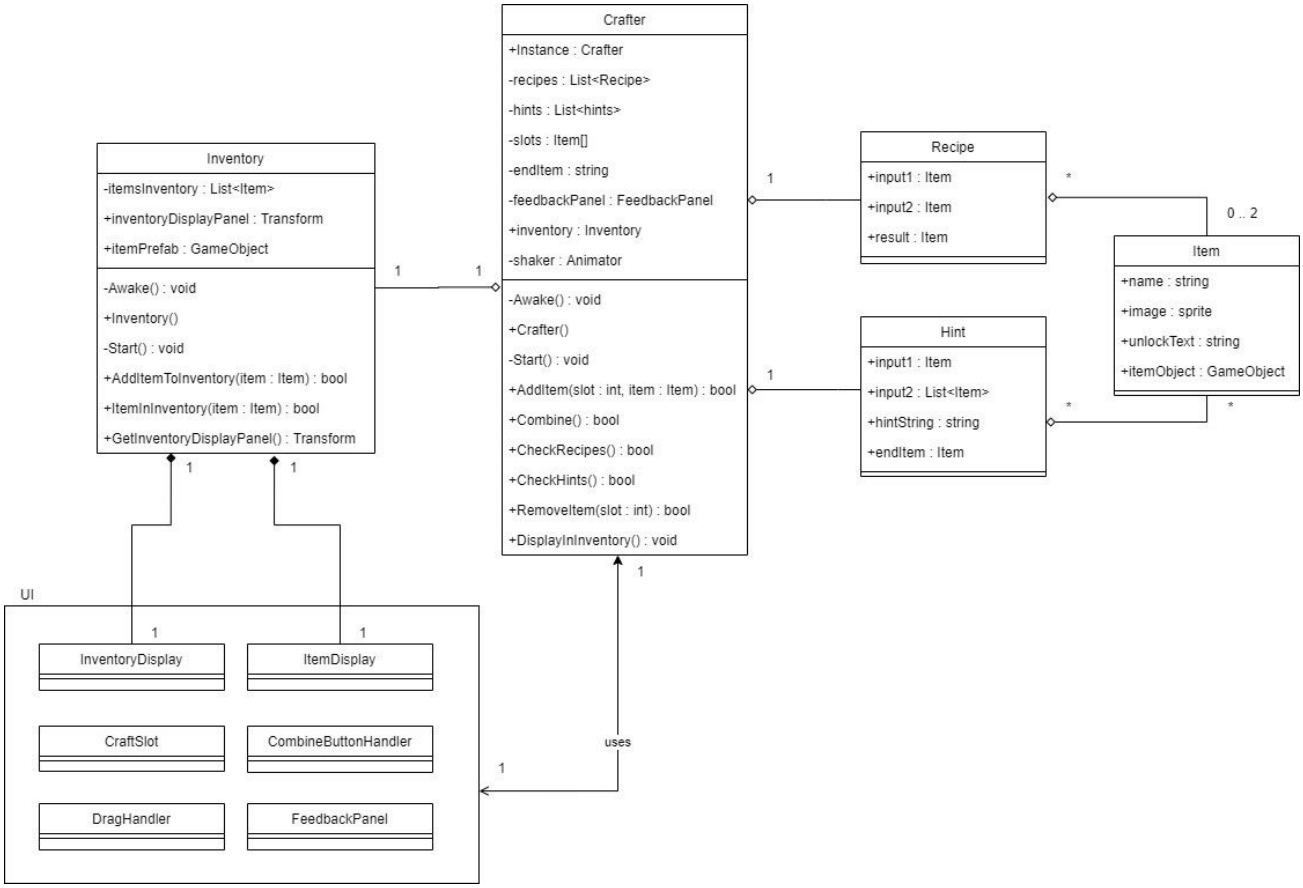


Figure 20: A UML class diagram of the Concepts game

4.3.2. Connection game and power-eco friendly game

The connection games are static games with pieces that can be rotated when a player clicks on them, the games are controlled by a manager called the ConnectionGameManager. This is a class with all functionalities to handle the games. The UML for this minigame can be seen in figure 21. It was chosen to be done this way because only two connection games are developed. If there are more types of connection games to be added to the game, it would be best to split the ConnectionGameManager class into an interface and a class for each type of game. This would allow for better expandability.

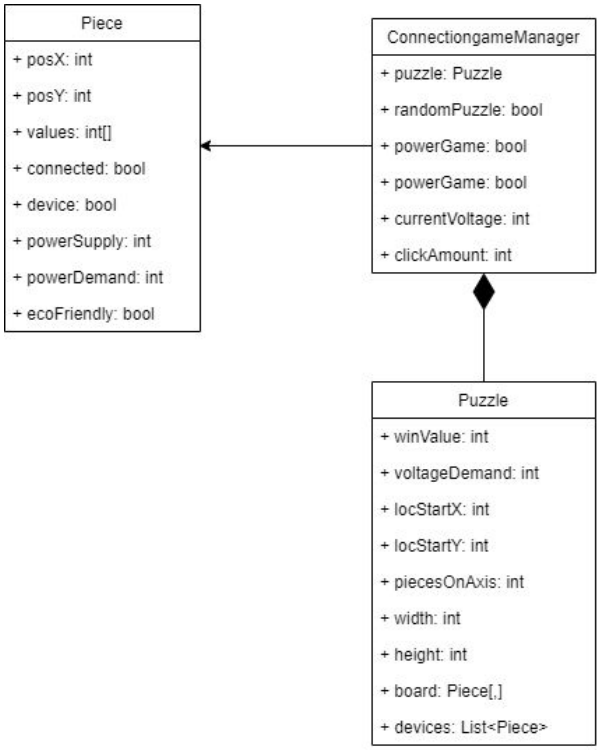


Figure 21: UML class diagram of the Connection game

Piece

A piece is a game object. Each piece can be constructed to be a piece for the server game or for the power game. To avoid creating individual pieces for the map, prefabs have been made to make the game easier to adjust, e.g. adjusting the sprite of a piece. Pieces can be rotated in the game by clicking on it. The rotation has been implemented to rotate +90 degrees. Once 360 has been reached the game resets this value to 0. It doesn't really matter for the functionality, but to be neat and avoid complications with future implementations (e.g. a different color for the piece at each rotation), it was implemented this way.

For the server game, a piece is normal. For the power game, the pieces get a value for *powerSupply*, *powerDemand,* and *ecoFriendly*.

Connection game manager

The connection games consist of the Server game and the Power game. The connection games are controlled by a single game manager. This manager is a very big class with all functionalities a connection game should have. The manager first checks which of the two game types is going to be played, by checking the booleans that are set up in Unity. Next, the game constructs a puzzle by creating a 2D array of the pieces. The coordinates of the pieces are set by a method in this class. For each click on a piece, the manager checks whether the game is won, by going over the whole puzzle. The difference between the two connection games is that

the server game requires all devices to be connected and the power game requires an exact amount of power supply to the source.

Random generation

Within the ConnectiongameManager class, there is a part of the code that handles the random generator of a puzzle. This is only possible for the server connection game, as the power connection game would require a different setting for the pieces. On top of that, it also has different requirements for winning the game. A puzzle is valid, if the puzzle has a source piece, at least one device piece, and the source piece can reach all the device pieces.

To generate a random puzzle a developer first needs to specify how many pieces there are on the x- and y-axis. Let this be n with $2 \leq n \leq 10$ and the puzzle is $n \times n$. The number 2 is the lowest value n can have because a 2×2 puzzle is the smallest size for a valid puzzle. The number 10 is the biggest value for n because an 11×11 puzzle would not fit the screen.

To avoid the possibility that a puzzle has multiple clusters, meaning the puzzle is not strongly connected, there is a method that goes over the pieces and verifies the puzzle. When the puzzle is not valid a new puzzle is made and the method is called again to check.

4.3.3. Data priority game

The data priority game is a dynamic game that is managed by two scripts namely, RandomSpawner and ScoreManager. RandomSpawner manages the spawning of the icons and ScoreManager manages the score and the gameplay. The UML for this minigame can be seen in figure 22.

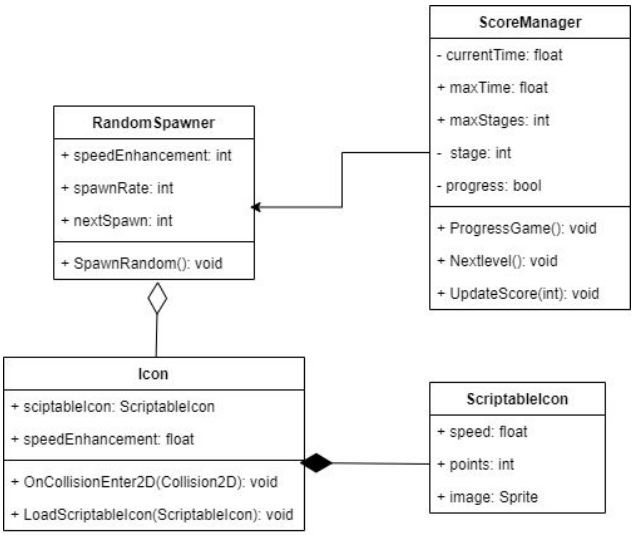


Figure 22: The UML of the data priority game

RandomSpawner

RandomSpawner takes a list of scriptable icons and ensures that these icons are spawned with the appropriate speed and position. The developer can set a number $0 < n < amount\ of\ time$ to assign the number of intervals. With each interval, the amount of spawn icons is increased.

ScoreManager

ScoreManager ensures there is correct gameplay and checks the progression of the game by constantly checking if the current time has reached the maximum time. The manager starts the game in stage 1 and goes to the next stage on specific moments calculated with the variables *maxStages*, *currentTime* and *maxTime*. During the game, the score is updated for each collision.

Icons

The icons are randomly created in the game, by using Scriptable Icons. An icon has a speed and a value for the score. When an icon collides with the basket of a player, the points of the icon are added to the score by the ScoreManager. If there is no collision with the player, but with the floor, the icon disappears and is deleted from the program.

4.3.4 Marketing game

This entire game is handled in a single class, namely the MarketingDisplay class. The objective of the game is to create a marketing slogan for a certain aspect of the data center. This task contains a sentence which describes the task as well as a codeword, which must be contained in the slogan. This is checked by the script and any slogan that doesn't contain the codeword will be rejected. The codeword is not case sensitive and a word that contains the codeword is also accepted. The task also contains an example slogan to give the participants some inspiration. The different marketing tasks are represented by a MarketingTask scriptable object (see figure 23).

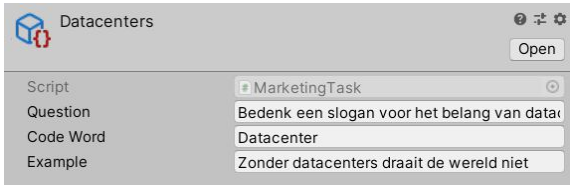


Figure 23: An example of a MarketingTask

To add or change a task for the game, a MarketingTask object must be created and added to the MarketingTasks array in the MarketingDisplay class. To change the order in which the MarketingTasks are displayed in-game, the MarketingTasks can be reordered in the array. This can all be done in the Unity editor and requires no use of any scripts.

4.3.5 Password game

The Password game is mostly contained in the PasswordDisplay class. This class handles the UI of the game, while the logic is contained in the PasswordTask class. The different tasks for the players are represented by a PasswordTask scriptable object. This object contains the name of the password, the length, the required elements, and a string containing the estimated time to brute-force the password. The required elements are uppercase characters, lowercase characters, digits, and special characters. Since there is no concrete definition of a special character, any character that is not a letter or number is accepted. Any configuration of these required elements is possible for a password task, including all and none of the elements. The password given by the player must contain all of the required elements and be of sufficient length to be accepted by the game

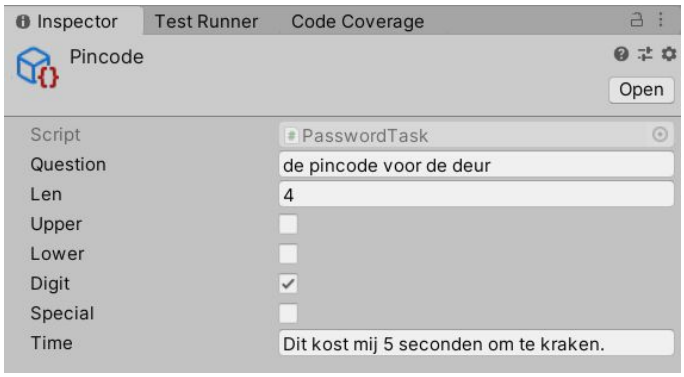


Figure 24: An example of a PasswordTask

To add or change a task for the game, a PasswordTask object must be created and added to the PasswordTasks array in the PasswordDisplay class. To change the order in which the PasswordTasks are displayed in-game, the PasswordTasks can be reordered in the array. This can all be done in the Unity editor and requires no use of any scripts.

4.3.6. Quiz

The quiz is controlled by a QuizManager entity. This entity is a singleton and is accessed throughout the UI. Every question that is loaded in the game, is made using scriptable objects. This allows easy addition, removal and modification of the questions.

At the beginning of the game, the questions are shuffled and for every separate question, the answers are also shuffled using the same method, shuffleList (see figure 25). The composition of the UI elements is not shown in the UML diagram, because these classes contain Unity-specific implementations. As for the separation of UI and logic, this is chosen to achieve low coupling. The logic can be maintained more easily and the UI is not dependent on the gamelogic, making the UI also more adjustable.

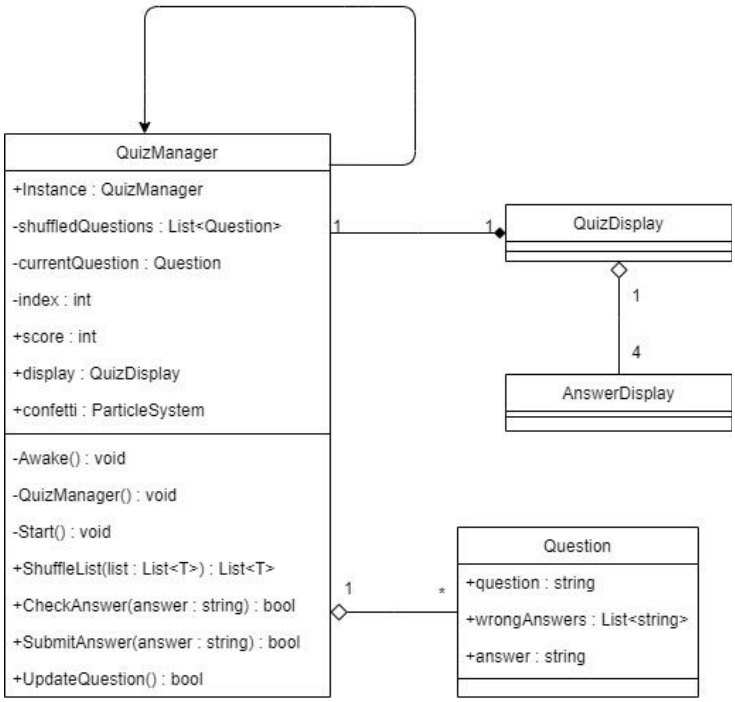


Figure 25: UML diagram of the Quiz game

4.6. SIG evaluation

Within the ten weeks of the project, there were two deadlines to send in all the code to the Software Improvement Group (SIG). The first deadline was in week 8, see **Appendix E: Project Plan**. After sending the code, the feedback was given. This section discusses the feedback and the team's approach with respect to processing the feedback.

4.6.1. First SIG delivery

SIG rated the source code a 3.7 out of 5 on maintainability. This is based on the following criteria (see **Appendix H: SIG Feedback**):

- Volume; size of the codebase (5 out of 5).
- Duplication; duplication of code throughout the project (5 out of 5).
- Unit size; size of methods/functions (2.3 out of 5).
- Unit complexity; complexity of methods/functions (2.5 out of 5).
- Unit interfacing; size of method/function interfaces. The number of input parameters for a single method/function (5 out of 5).
- Module coupling; degree of interdependence between modules (5 out of 5).

In general, the feedback was positive. The score on module coupling and unit interfacing was most likely caused by splitting our application into different scenes and games. Every game handles its own logic, in such a way that no two games can be intertwined.

On the other hand, two points needed attention: unit size and unit complexity. The reason for the low rating for these two criteria is that some methods did not have a single responsibility; a lot of behavior was handled in a single method.

A lot of methods were refactored after the feedback, reducing the overall complexity and unit size. The unit size was reduced by splitting the behavior into more methods. This also reduced the unit complexity, since a lot of methods that were too large also had a high complexity.

4.6.2. Second SIG delivery

SIG rated the source code a 3.9 out of 5 on maintainability out of the following criteria (see **Appendix H: SIG Feedback**):

- Volume; size of the codebase (5 out of 5).
- Duplication; duplication of code throughout the project (5 out of 5).
- Unit size; size of methods/functions (4.1 out of 5).
- Unit complexity; complexity of methods/functions (4.4 out of 5).
- Unit interfacing; size of method/function interfaces. The number of input parameters for a single method/function (2.5 out of 5).
- Module coupling; degree of interdependence between modules (4.7 out of 5).

Compared to the first SIG delivery, the maintainability of the source code improved. This was caused by two positive changes and one negative change to certain aspects of the source code. After the first SIG delivery, it became clear that the unit size and complexity needed improvement. By refactoring the long methods with high complexity, the team was able to improve unit size and complexity by 1.8 and 1.9 points respectively. However, due to this refactoring, new methods with a relatively high amount of input parameters were introduced, causing the unit interfacing to drop 2.5 points.

This is, however, a relatively simple problem to solve in the future. By introducing global variables to store often passed parameters or creating new objects containing these different parameters, the number of parameters passed to these specific methods will decrease.

4.7. Software quality

As was seen in **Appendix J: Research Report**, numerous goals and targets for the product were established. In order to achieve these goals in a sustainable and maintainable manner, it is crucial that the product is properly tested.

4.7.1. Testing

To test the product, various high- and low-level tests have been constructed throughout the development.

Unit tests

For low-level testing, Unity offers the Unity Test Framework (UTF). This framework is based on the NUnit library, an open-source unit testing library for .NET languages. With this framework, numerous constructs used during the development of a Unity game can be tested. The tests can then be run from within Unity itself.

During the implementation of a feature, regression testing was used in order to ensure that no pre-existing functionality is broken. Additionally, new tests were also created and ran in order to validate the implemented functionality. A Continuous Integration environment was set up in order to ensure that each developed build could easily be evaluated and tested for errors.

As set in the requirements, the team aimed for a coverage of 70% of the game logic, excluding UI and display classes. In the end, a coverage of 74% was reached. As seen in figure 26, some classes were entirely covered, while some classes were also left out. These classes encapsulate small functionalities and their CRAP score (see 4.7.2) is already low, making these classes not a priority in terms of testing.

MarketingDisplay	48	0	48	95	100%	
MarketingTask	3	0	3	20	100%	
PasswordDisplay	106	0	106	167	100%	
PasswordTask	23	0	23	54	100%	
Question	1	0	1	13	100%	
ConnectionGameManager	348	21	369	600	94.3%	
DialogueManager	112	15	127	201	88.1%	
QuizManager	48	11	59	104	81.3%	
GameProgression	54	14	68	117	79.4%	
Crafter	63	22	85	147	74.1%	
ScoreManager	52	28	80	155	65%	
Dialogue	41	23	64	105	64%	
Piece	23	16	39	96	58.9%	
SceneTransition	9	12	21	37	42.8%	
RandomSpawn	18	30	48	102	37.5%	
Icon	7	30	37	57	18.9%	
PlayerMovement	1	34	35	65	2.8%	
DialogueTrigger	0	9	9	21	0%	
Inventory	0	34	34	71	0%	
LightManager	0	3	3	11	0%	
OverlayManager	0	18	18	35	0%	
SpawnGate	0	9	9	22	0%	

Figure 26: Test coverage of the logic classes

High-level tests

High-level tests, which are mostly related to the presentation of the game, are not easily automated. While frameworks exist, they are not very workable or maintainable. For this reason, the trade-offs were not considered worth it and it was decided to manually do high-level tests. The aspects of the UI that had to be tested are the responsiveness, the functionality, and the positioning. As the game is targeted towards early adolescents, it should have a responsive, functional, and clear UI.

The presentation of the game was manually tested as each feature was developed, and during the examination of its implementation in a merge request. After the introduction of each UI element, its layout and positioning relative to the screen canvas was examined to make sure that the intended relative positioning among the UI elements was maintained, and that the UI stayed clear and easily viewable. As functionality was added to the UI, this was usually controlled from the UI through the use of event listeners. These listeners captured specific user input, after which a certain action was carried out. As each UI element was coupled with a certain action, the proper execution of this action, given a certain UI listening event, was observed.

For the implementation of features, clear issues were developed that detailed the functionality of the feature to be implemented. Thus, before the approval of a merge request, this functionality could be iteratively tested to ensure that it functioned correctly. Besides functionality, in the above-mentioned ways, the responsiveness was observed to ensure that interaction with the UI is workable and smooth.

One of the requirements set up for the game is that it is playable on a low range device that could be found in a typical classroom environment. To this end, a testing procedure was established that ensures that the product is playable on low-end machines. The two main things that were tested against were the responsiveness of the game as well as the memory footprint of the application.

For a player to feel well-connected with the application, the response time should fall within a certain limit. There are 3 thresholds for response times: 100ms is the limit for a user to perceive the action as immediate. 1 second is the limit for a user to feel the delay but not feel interrupted. Finally, 10 seconds is the limit for a user to get distracted and to get the urge to start doing something else while the action is being executed (Card et al., 1991).

Ideally, the response time of any action in the game should be below 100ms, but anything less than 1 second is still acceptable since the player does not lose the feeling of directly controlling the application. In practice, this meant that any action, be it a button press, or moving between rooms, should take less than 100ms.

This was tested by looking at the requestAnimationFrame call that is defined in the HTML-standard[3] (figure 27). In these tests, the game was hosted locally on a server and the game was tested in the Chrome browser, with the CPU throttled 6x to simulate a low-end machine.

[3] **requestAnimationFrame**, a call standardized in the HTML 5.1 conventions from: https://www.w3.org/TR/html51/webappapis.html#animation-frames

Figure 27: The requestAnimationFrame of a transition to a different room

Similarly, the memory footprint of the application was also tested. To validate that the application is able to run on low-end machines, different locations of the game were loaded in, and a snapshot of the memory footprint was taken. The memory footprint in each location should be less than 250MB (see figure 28).

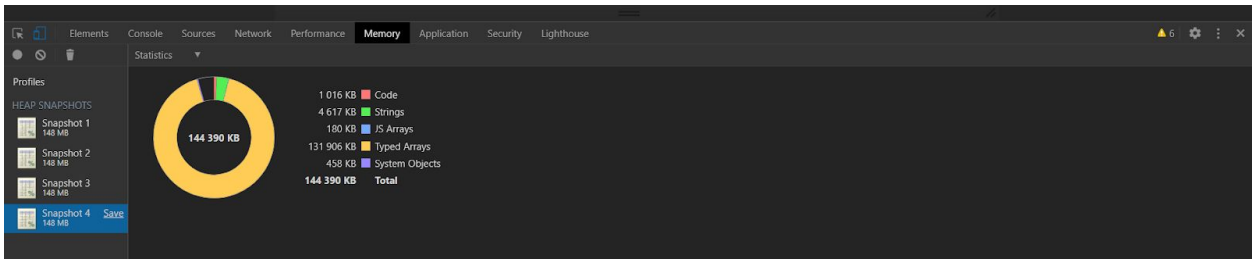


Figure 28: The memory footprint during an interaction with an NPC

4.7.2. Code analysis

To determine the quality of the written code, the CRAP1[4] metric was used. This metric determines the quality of the code by looking at the cyclomatic complexity (*comp*), as well as the unit test coverage of methods or functions (*cov*). A high CRAP score indicates that the method is difficult to extend and maintain. Higher cyclomatic complexity means a higher CRAP score, while higher unit-test coverage levels lower the CRAP score. The CRAP metric of a method *m* is calculated according to the following function[2]:

$$CRAP(m) = comp(m)^2 * (1 - cov(m)/100)^3 + comp(m)$$

The threshold for the CRAP metric is designed to be a score of 30. Any method with a higher score needs to either be split up into smaller pieces or tested more thoroughly. Though there is a limit to the complexity of the method, if the cyclomatic complexity exceeds 30, no amount of testing will compensate for this and the CRAP score will always exceed the threshold.

During the project, the CRAP and coverage metrics were tracked by creating reports (see figure 29). Through refactoring large methods and extensive testing, it was ensured that no methods exceeded the set threshold.

[4] **Change Risk Anti-Patterns**, a metric to measure the quality and health of a code-base. https://testing.googleblog.com/2011/02/this-code-is-crap.html

Metrics

Method	Cyclomatic complexity	NPath complexity	Sequence coverage	Branch coverage	Crap Score
StartDialogue(...)	1	0	100%	0%	1
DisplayNextSentence()	4	0	84.2%	0%	4.06
TypeSentence()	4	0	90.9%	0%	4.01
DisplayChoicesUI(...)	2	0	100%	0%	2
SetButtonAction(...)	6	0	100%	0%	6
StartFromIndex(...)	1	0	100%	0%	1
LoadScene()	1	0	0%	0%	2
DisplayChoice()	1	0	100%	0%	1
UpdateStage()	1	0	100%	0%	1
EndDialogue()	1	0	100%	0%	1
ParseGoToSubstage(...)	1	0	100%	0%	1
ParseScene(...)	2	0	63.6%	0%	2.19

Figure 29: The coverage and complexity metrics of the DialogueManager class

5. Process

In this section, the process of developing the project is discussed. The purpose of this section is to give an overview of how the project was planned, how the issues were prioritized, what methodology for communication was used, how and what feedback was given, and the challenges encountered while developing DataFlex**.**

5.1. Planning

The product deadline which was on the 3rd of July, 2020. This was the moment the product was presented for the end evaluation by VHTO and the supervisor. To realize the deadline, a planning was created with several phases and internal deadlines for the game (see **Appendix E: Project Plan**):

1. *May 3, research phase.* Identify all key problems, contexts, and requirements. Discuss and give an in-depth explanation and justification of the identified aspects. Create the project plan, (**Appendix E: Project Plan**).
2. *May 10, Design phase:* Setting up the tools and settings for the development phase. Finalize the game idea with the client.
3. *May 31, Development phase:* Create a bèta version of the game.
4. *June 21, Development phase:* Receive feedback from experts and students on the béta version 2.0
5. *June 26, Presentation phase:* Finalize the game and the documents
6. *July 3, Presentation phase*: Present the final product

In the development phase, SCRUM[5] was used to develop, test, and evaluate functionalities in the game in an iterative way. Each sprint lasted one week, starting on Monday and ending on Sunday. Every two days the team members would discuss the work. At the beginning of the sprint, a sprint backlog with tasks was created. At the end of the sprint, a retrospective was created, with both positive points, as well as points for improvement. This allowed for looking back on the work of the week and this feedback was used to progress into a new cycle. These sprint documents were sent to the client to receive feedback.

5.2. MoSCoW

To keep track of the functionalities that needed to be implemented, a MoSCoW analysis was created (see **Appendix C: MoSCoW**). The MoSCoW analysis provides an overview of the prioritization of the elements in the game, while at the same time dividing functionalities into smaller, more manageable parts. This MoSCoW document provides the additional benefit of being able to specify the functionalities that the final product will not contain. With the help of

[5] **SCRUM**, an agile framework for development. https://agilemanifesto.org/

this analysis, the team could focus on the more important functionalities. First, the 'must-haves' were added to the product, followed by the 'should-haves' and 'could-haves'.

5.3. Communication

The representative of VHTO, Drs. Ir S. de Wit, and the supervisor, Dr. Ir WP. Brinkman, were primarily involved with the project. S. de Wit communicated with VHTO and with Interxion. Each week, there was a meeting on Jitsi[6] with S, de Wit and WP. Brinkman. The developers had separate meetings on Discord. In these meetings, the current state of the work of every member was discussed, issues were handled and decisions were made with regards to the product.

Communication with the client and supervisor outside the meetings was done through mail. The internal communication was done through WhatsApp[7] and Discord[8]. Gitlab[9] was used to keep track of issues for the development of the project.

5.4. Feedback process

In the first phase of the project, the main feedback given by the supervisor was to create as many ideas, sketches, and prototypes as possible, so that the client would have the opportUnity to choose between different ideas. Because of this, many ideas were pitched and created in the second phase of the project, leading to a selection of seven minigames.

In order to obtain qualitative feedback on the game, the expertise of researchers at Leiden University was enlisted. Most importantly, this was done to obtain feedback on the entertainment and educational value of the minigames. A questionnaire (**Appendix F: Questionnaire 1**) was prepared in which the educational and entertainment value for each minigame could be ranked, and general comments could be given. Due to time constraints, a limited amount of feedback was received. In the end, only three people were able to fill in the feedback form, (see **Appendix F: Questionnaire 1**). Overall, the feedback was good.

Next, the client mentioned that the competitive aspect implemented in the game was not optimal for children. Some children would benefit more if they would work together co-operatively (Nosek et al., 2009). The game was changed to have co-operative aspects, such as answering the question for the quiz game together, instead of individually.

Apart from the experts, Interxion was asked to give feedback on the product and a special meeting was scheduled to show the beta version to Interxion. They were very happy with the result. This showed the team that the game was going in the right direction. There was one remark, which was that there is no score system. VHTO gave feedback that the game was a bit

[6] **Jitsi**, a video call application https://jitsi.ewi.tudelft.nl/
[7] **WhatApp**, mobile messaging application https://www.whatsapp.com/
[8] **Discord**, text-, video-, and audio-communication application https://discord.com/
[9] **GitLab**, web-based DevOps lifecycle tool https://about.gitlab.com/

too dark colored. On top of that, there were inconsistencies and non-gender neutral aspects in the game, e.g. the marketing game had a pink background while the others had blue. This was changed to avoid these stereotypes.

In the last weeks of the game, many students and other individuals connected to education were asked to play the game and review it. The form that was used to conduct this evaluation can be found in **Appendix G: Questionnaire 2**, together with the results. The feedback from this is discussed in section 6.1 as this is seen as the last evaluation point of the product.

5.5. Challenges

Some of the challenges faced while developing the game are stated in this section.

5.5.1. Unity and testing

Developing in Unity turned out to be a challenge in the first weeks of the project. Unity was a new engine for the majority of the team. This required more time at the beginning of the project to learn certain concepts, e.g. prefabs and scriptable objects. Even though this took a bit of time, it was worth it considering the improved workflow. The use of Unity did not only pose a challenge to the development process, but also the testing process.

In the beginning, the team was inexperienced with the unit testing of games. Unit testing in Unity turned out to be more of a challenge than initially expected. On top of this, Moq, a framework that allows the mocking of dependencies, had to be learned. This resulted in some shifting within the schedule. The team prospered through this challenge by putting extra effort into testing and managed to reach at least 70% meaningful test coverage.

5.5.2. Graphical design

One of the things that were decided at the beginning of the project was the outsourcing of graphical design. In order to implement this, an overview of all the graphical elements in the game had to be sent to the client. However, this took more time than planned. At the beginning of the project, a lot was still being researched without starting the development of the game. This resulted in the overview being sent too late. A lot of sprites and textures were not thought out early enough, resulting in the implementation of the graphical design after the project.

5.5.3. Communication with the client

The communication with the client was not always clear from the team's side. One of the points that the client made in the middle of the development phase was that they were interested in the weekly planning. This was not sent in the weeks before, which led to the client not being updated on the current state of the project. Later, the team made sure that every sprint was sent to the client. This could have been solved by explicitly designating a member who would be responsible for the contact between the client and the team every week.

5.5.4. COVID-19

An unforeseen circumstance was the COVID-19 pandemic. Due to measures taken by the government, the team was not able to meet in person. This slowed down some of the processes, however, this was resolved by meeting online. However, this was not the only aspect that was influenced. A field test became impossible, due to schools canceling their schedule. This required the team to be more flexible and perform tests with other subjects, rather than the target audience.

6. Conclusion and discussion

Two main objectives were defined at the start of the project, namely educating children about data centers and creating a positive attitude towards IT-related fields. DataFlex educates children through various learning objectives that are embedded within the minigames. It stimulates learning by providing an entertaining and challenging experience to the user. Furthermore, it attempts to give the user a positive attitude towards IT by allowing the user to engage with IT-related concepts while coming out with a positive experience. As this objective is intended to increase the participation of women in IT, DataFlex attempts to stimulate the interest of women by being gender-inclusive and breaking stereotypes.

6.1. Evaluation of product

In this chapter, an evaluation will be given of the final product in order to assess whether the given task was adequately solved. In order to obtain a final evaluation of DataFlex, the final version was completed and provided, along with a questionnaire (see **Appendix G: Questionnaire 2**). Through this questionnaire, an assessment was obtained on the various features. In order to evaluate DataFlex, the requirements will be processed (see figure 1, section 2.3). Based on the aforementioned feedback data, the extent to which these requirements are fulfilled will be discussed.

6.1.1. Educate about data centers

The first main objective for DataFlex was that it had to teach children about data centers and incorporate concepts such as the cloud, data processing, and connectivity.

Educational

As was described (see chapter 3: Design), learning goals were set up in order to teach the children about the different aspects of a data center. On the basis of these goals, the learning goals were coupled to each task within the game in order to educate the user. When asked about the educational value of DataFlex, all the respondents indicated that it successfully taught the basics of a data center, with many of the respondents highlighting the security aspect that was taught.

Fun

The final aspect for fostering a positive attitude was that the game had to be entertaining and engaging for the players. When asked for their opinion on the game's fun factor, all the respondents considered the game to be fun (see **Appendix G: Questionnaire 2**).

Challenging

When asked to rate the difficulty of DataFlex for the target audience, most respondents argued that the game was appropriately difficult. The most difficult aspect was identified by most players to be the reading and following of the instructions (see **Appendix G: Questionnaire 2**).

6.1.2. Create a positive attitude towards IT

The second objective was for DataFlex to create a positive attitude towards IT. This was achieved by creating a positive experience for every user, breaking stereotypes and making the game gender inclusive.

Positive experience

Furthermore, in order to further foster a positive attitude, DataFlex was developed in such a way that every player could have a positive experience. To this end, tutorials and hints were provided throughout the game to make sure that the player doesn't get stuck. The tasks that the player has to complete in DataFlex were also developed in such a way that the player is unable to lose. Additionally, the storyline is used to guide the player toward the right path.

Breaking stereotypes

DataFlex was built to be gender-neutral. Care was taken to have an even division of male and female characters and to also give the female characters important roles. This can be seen in figure 4, which demonstrates that female characters fulfill roles such as manager or engineer. In order to avoid propagating the image of the IT field being a male-dominated world, traits considered as typically masculine such as competitiveness were avoided. Instead, the co-operative aspect was highlighted and players were given the opportunity to work together to solve puzzles and complete tasks.

Gender inclusion

One of the main requirements for DataFlex was that it had to increase gender diversity in STEM (in the case of DataFlex IT specifically) by creating a positive attitude about technology, with a focus on girls in prevocational education. At the forefront of this goal were gender inclusiveness and the breaking of stereotypes. Additionally, a wide variety of aspects of a data center were displayed, as the social aspect and impact of technology appeal more to girls.

6.1.3. Interxion

After completing a demo version, a meeting was set up with a representative from Interxion where the mechanics and gameplay of DataFlex could be played through and displayed. In this meeting, very positive feedback was received, indicating that the product fulfilled expectations.

The art style and overall presentation of the game were thought to be entertaining. Additionally, the presentation of the aspects of a data center was considered well varied, showing not only the technical but also the social aspect. The representative was convinced that DataFlex would be entertaining and educational for kids.

6.1. Non-functional requirements

As specified in the requirements, the game should take a maximum of one hour to complete. All respondents were able to complete the game within an hour. With most respondents having some time left. It should be noted that the respondents were all adults and that DataFlex should prove more difficult for children. Additionally, the ability was added to allow the players to continue playing the game after it has finished. This replayability can provide enough content to fill up the hour, should it be needed (see **Appendix G: Questionnaire 2**).

As shown in the implementation chapter (see 4: Implementation), DataFlex has been developed in a way that allows for extendability, as well as maintainability. Clear documentation of the code allows new developers to expand on the already existing functionalities. To make sure that DataFlex could run on low-end devices performance testing has been done, which is also documented in the chapter 4. Implementation. Furthermore, the logic contained within the various scripts has been tested by unit tests, with a coverage of approximately 74%, which exceeds the original goal of 70%.

6.3 Limitations

There were a few factors that could limit the applicability of the study. Any follow-up studies that reference this report should take these limitations into account. The main limitations are discussed in this section.

6.3.1 Field test

Due to current circumstances regarding a global pandemic, actual testing with subjects of the target age group could not be done. Additionally, the procedures for doing this would not fit the timeline constraints. For these reasons, It was already established from the beginning of the project that testing DataFlex with the target audience would not be feasible and it was instead decided to carry out tests with a group of researchers possessing an educational background. Thus, it could not be validated that the learning objectives were actually reached. If a field test with the target audience were to be done, the game could have been adjusted accordingly. The number of respondents was quite small, which means that meaningful quantitative conclusions could not be drawn. However, as this was expected, the aim was mainly on qualitative feedback (see **Appendix G: Questionnaire 2**).

6.3.2 Research on other aspects

The research on the design of the game was mainly focussed on age and gender, the game was designed mainly to cater to the target audience, girls between 10 and 14 years. More research could be done by also looking at other aspects of the audience, such as educational level and what kind of background knowledge the participants have. By taking these additional aspects into account, the game could be better suited for this audience.

6.3.3 Team diversity

The game was developed by a team consisting exclusively of men. While the design was based on literature, having more diversity in the team could result in some additional insights on how to cater the game to girls. Care was taken to be as unbiased as possible during the project, but there might have been some cases where unconscious bias could have been propagated into the product. Having a more diverse team could prevent this bias.

6.4 Ethical implications

The target audience of the game is children between 10 and 14 years, so special care was taken to ensure that the project was ethically responsible.

Firstly, the product stores no information about the players. Since the players are under the age of 16, special rules regarding the storage and processing of the data generated by the players apply[10]. While the game does run in a browser, no information about the game session is stored on the cloud.

Secondly, any inputs that the player gives are only shown on the machine that the user is currently using. This prevents the players from distributing offensive material through the game. Finally, any elements that might not be suitable for a young audience, like sexist or racist comments, as well as risque character design and dialogues, were purposefully avoided.

6.5 Recommendations

Due to the limited amount of development time, some of the initial ideas did not make it into the final product. These ideas can, however, be implemented in the future to enhance the game experience. How these ideas can be implemented and how the user experience improves with the addition of these functionalities will be explained below.

[10] **Chapter 2 Article 8 of the GDPR**, a regulation on the use of data generated by minors https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&rid=3

Interactable objects

Interactable objects would enlarge the exploration element of DataFlex, while at the same time offering an extra opportunity for the users to learn about data centers. With the help of the flexible dialogue system, dialogues can easily be added to different objects. A dialogue could, for example, be added to the fire extinguishers, in which the need for these extinguishers could be explained. If there are more objects for the player to interact with, there is more incentive to explore the different rooms.

Variables in dialogues

Although the dialogue system is quite powerful, there is currently no functionality to display variables, such as player names or scores, within the dialogue text. The DialogueManager script can be expanded to be able to parse variables. The GameProgression script would also have to be updated to store the desired variables. The usage of these variables would make the game more personalized, which could create a more immersive feeling.

Visual room changes

One of the ways to visualize story progression, rather than solely changing the dialogues, is by changing the appearance of the rooms based on the completion of certain minigames. DataFlex currently has this functionality implemented for the power game, where after the completion of the power game, the rooms get brighter. Whether minigames have been completed or not, is stored within the GameProgression script. Sprites or game objects can then be updated based on these completions.

Progression menu

Currently, various NPCs will point the player in the right direction based on their progression. However, there is no way for the player to see precisely which minigame they have completed or not. By adding a pause menu to the main game, the minigames that still have to be completed, as well as the list of suspects, could be displayed at any moment during the game. This would make it a lot clearer for the players where they would have to go next and could emphasize the mystery element of the main game.

6.6. Scientific contribution

To our knowledge, this is the first Dutch educational game about data centers. Furthermore, various research about the interests in certain game types of children within various age groups, or of a certain gender, existed prior to our project. However, as far as we know, this has not yet been combined with the research done on educational games, in such a way that an educational game which caters to the needs of specific groups can be created. In our case, this was the focus on a specific target group: children aged 10 to 14, with a focus on girls, in pre-vocational education. At last, the game also focuses on societal aspects such as gender

inclusion and diversity. We hope that this project is a step towards a more equal representation of women in IT.

Bibliography

Booy, C. (2012). Trend analysis: gender in higher STEM education. VHTO, National Expert Organisation Girls/Women and Science/Technology.

Brenick, A., Henning, A., Killen, M., O'connor, A., & Collins, M. (2007). Social Evaluations Of Stereotypic Images In Video Games: Unfair, Legitimate, Or “Just Entertainment”?. Youth & Society, 38(4), 395-419.

Card, S. K., Robertson, G. G., & Mackinlay, J. D. (1991, March). The information visualizer, an information workspace. In Proceedings of the SIGCHI Conference on Human factors in computing systems (pp. 181-186).

Dutch Data Center Association. (2019). Onderwijs & Werkgelegenheid. Retrieved Juni 22 2020, from https://www.dutchdata centers.nl/posities/onderwijs-werkgelegenheid/

Easlea, B. (1987). Patriarchy, Scientists, And Nuclear Warriors. 1987, 195, 215.

Gerling, K. M., Birk, M., Mandryk, R. L., & Doucette, A. (2013). The Effects of Graphical Fidelity on Player Experience. In *Proceedings of International Conference on Making Sense of Converging Media - AcademicMindTrek ’13*. https://doi.org/10.1145/2523429.2523473

Hartmann, T., & Klimmt, C. (2006). Gender and Computer Games: Exploring Females’ Dislikes. In *Journal of Computer-Mediated Communication* (Vol. 11, Issue 4, pp. 910–931). https://doi.org/10.1111/j.1083-6101.2006.00301.x

Kinzie, M. B., & Joseph, D. R. D. (2008). Gender differences in game activity preferences of middle school children: implications for educational game design. In *Educational Technology Research and Development* (Vol. 56, Issues 5-6, pp. 643–663). https://doi.org/10.1007/s11423-007-9076-z

Masuch, M., & Röber, N. (2004). Game graphics beyond realism: Then, now and tomorrow. In

Level UP: Digital Games Research Conference. DIGRA, Faculty of Arts, University of Utrecht.

Niederle, M., & Vesterlund, L. (2010). Explaining the gender gap in math test scores: The role of competition. Journal of Economic Perspectives, 24(2), 129-44.

Nosek, B. A., Smyth, F. L., Sriram, N., Lindner, N. M., Devos, T., Ayala, A., ... & Kesebir, S. (2009). National Differences In Gender–Science Stereotypes Predict National Sex Differences In Science And Math Achievement. Proceedings Of The National Academy Of Sciences, 106(26), 10593-10597.

Prensky, M. (2001). Fun, play and games: What makes games engaging. Digital game-based learning, 5(1), 5-31.

Reuben, E., Sapienza, P., & Zingales, L. (2014). How stereotypes impair women's careers in science. Proceedings of the National Academy of Sciences, 111(12), 4403-4408.

Steinke, J. (2016). A portrait of a woman as a scientist: breaking down barriers created by gender-role stereotypes. In *Public Understanding of Science* (Vol. 6, Issue 4, pp. 409–428). https://doi.org/10.1088/0963-6625/6/4/006

Sentance, S., Barendsen, E., & Schulte, C. (2018). *Computer Science Education*. Zaltbommel, Nederland: Van Haren Publishing.

Tekofsky, S., Miller, P., Spronck, P., & Slavin, K. (2017). The Effect of Gender, Native English Speaking, and Age on Game Genre Preference and Gaming Motivations. In *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering* (pp. 178–183). https://doi.org/10.1007/978-3-319-49616-0_17

Ter Bogt, T. F., Engels, R. C., Bogers, S., & Kloosterman, M. (2010). "Shake It Baby, Shake It": Media Preferences, Sexual Attitudes And Gender Stereotypes Among Adolescents. Sex Roles, 63(11-12), 844-859.

Terwogt, M. M., & Hoeksma, J. B. (1995). Colors and emotions: preferences and combinations. *The Journal of General Psychology*, *122*(1), 5–17.

VHTO. (2020). *Missie en visie - VHTO*. Retrieved June 24 2020, from https://www.vhto.nl/over-vhto/missie-en-visie/

Wal, J., & Wilde, J. (2017). *Identiteitsontwikkeling en leerlingbegeleiding* (5de editie). Bussum, Nederland: Coutinho.

Appendix A: Info Sheet

Title of the project: Educational game about data centers for children
Name of the client organization: VHTO
Date of the final presentation: 02-07-2020
Description: To tackle the issue of a lack of interest in STEM of girls in pre-vocational secondary education. VHTO tasked us to create an educational game to spark more interest in IT and specifically around data centers. Moreover, the game should break gender stereotypes.
Challenge: Designing a game for girls between 10 and 14 was quite difficult for a team of 5 men. A lot of time was dedicated to designing a game that specifically caters to such a group.
Research: The research was focussed around the ways to break stereotypes, the design of educational games, the best type of game for the target audience, and general concepts from data centers that are included in the game.
Process: The project started with a Research phase, after research the requirements and design were established in the Design phase. The product was developed in the Development phase. During the development phase, the project was developed using Scrum, this allowed the team to continuously collect and process feedback from the client. In the final phase, the product was assessed and adjusted accordingly.
Product: The product created was a 2D role-playing game with puzzle elements, the game is web-based and can be played within a browser. The product was tested using automated tests, manual tests as well as an analysis of the codebase by the Software Improvement Group. With this test suite, the quality, extensibility, and performance of the product were validated. All of the core features were implemented.
Members of the project team:
Name: Gedeon d'Abreu de Paulo
Interests: Data management, AI, Computer Science
Role & Contributions: Advisor, Developer
Name: Bilal el Attar
Interests: Game/Software Development, Front-and Back-end Development, AI
Role & Contributions: Developer, UI
Name: Ali Al-Kaswan
Interests: Software Testing, Scrum, Back-end Design
Role & Contributions: Developer, Map Design, High-Level Testing, Performance Testing
Name: Nathan Kronstadt
Interests: Software development, Game development, Educational Sciences
Role & Contributions: Developer, Public Relations
Name: Gianni Wiemers
Interests: Software development, AI, Computer networks, Computer graphics, Cyber security
Role & Contributions: Developer, Engineer
Client: S. de Wit, VHTO
Coach: W.P. Brinkman, Interactive Intelligence Group TU Delft
Contacts:
Ali Al-Kaswan: a.alkaswan@gmail.com
Nathan Kronstadt: lufther@live.com
The final report for this project can be found at: http://repository.tudelft.nl/

Appendix B: Project Description

The original project description of the project "Educational game about data centers for children" is stated in this appendix.

The project

Within this project you will create an educational game about data centers for children in the age of 10 to 14. The game has two main goals, namely 1) teaching children about data centers and 2) creating a positive attitude about technology.

Creating a positive attitude

It is our mission to increase gender diversity in STEM fields, including Computer Science. We want this game to create a positive attitude about technology, with a focus on girls in pre-vocational education. In this way we want to stimulate these girls to pursue a career in STEM, and more specific a career involving data centers.
How you will accomplish these goals within a game is something you will, together with the client, research in the first week.

Possible research questions

The research questions can relate to how you can achieve a change in attitude within a game. Also doing research on gaming, gender, and career orientation would, from our point of view, be really interesting. If you have any other research questions which might fit in this project, just let me know and we will see whether it fits.

Client

The project is a collaboration between VHTO and Interxion. During this project you will be working with Shirley de Wit. Shirley is employed by VHTO, the Dutch national expert organization on girls/women and science/technology. Shirley is also a PhD student at the Programming Education Research Lab (PERL) at Leiden University. Shirley has master degrees in both Computer Science and Science Communication from Delft University of Technology

Appendix C: MoSCoW

Main Game

Must have

- The players must be able to walk around in both the x- and y-direction
- The game must be playable by two players
- Non-Player Character guides the player
- The players must be able to interact with NPCs
- The players must be able to select a predesigned character
- The game must have a 2D map with multiple rooms
- The game must represent a data center
- The game must have a screen for each minigame showing the rules
- Minigame: Server Connection game
- Minigame: Concepts game
- Minigame: Data priority game
- Minigame: Marketing game
- Minigame: Password game
- Minigame: Quiz
- The choice to start a minigame
- The game must show women in technical careers

Should have

- Minigame: Power Connection game
- Milestones to dictate the progress
- The user should be able to customize their character
- Audio for gameplay
- The game should give the player hints when he/she is stuck
- The game should show progress through environmental changes
- The player should be able to stop a game by going back to the map

Could have

- Minigame: Power eco-friendly game
- Minigame: Security game
- Minigame: Cooling game
- The game could include a scoring system

Won’t have

- Online gameplay with other players
- Online leaderboard
- Savegames

Non-functional requirements

- The game is playable on Chrome 83.0.4103.97, Firefox 77.0.1, Edge 83.0.478.45 for Windows
- The game is developed in Unity 2019.4.0 using C# 7.3 for scripting
- 70% of code that can be tested in a meaningful manner will be tested is such a way, using automatic testing tools such as NUnit and Moq (where meaningful means that the test actually tests the functionality of the game)
- The game will be developed using Scrum and Gitlab
- The technical quality of the code will be assessed by the Software Improvement Group
- The game will run in a 16:9 aspect ratio

Concepts game

Must have

- The minigame must-have elements that represent parts or a data center
- The minigame must allow the players to move elements
- The minigame must allow elements to be combined
- The minigame must have an end element to create

Should have

- The minigame should have an explanation for each valid combination
- The minigame should show when wrong elements have been combined

Could have

- The minigame could allow for players to hover over elements to see what they are
- The game could give hints when wrong elements are combined

Won’t have

- Feedback for every combination of elements
- Keyboard controls
- Combinations with more than two elements

Marketing game

Must have

- The minigame must have a box to type a slogan
- The minigame must have a button to submit the slogan
- The minigame must show words that need to be in the slogan

Should have

- The minigame should have a checker for bad words
- The minigame should provide an example

- The minigame should have a compliment when an answer is given according to the rules

Could have
- The game could validate an answer by clicking enter

Won't have
- The slogans will not appear elsewhere in the game
- The slogans will not be stored

Quiz game

Must have
- The minigame must have a question box
- The minigame must have multiple-choice answers
- The player must be able to pick any answer

Should have
- The minigame should visualize the score
- The minigame should provide feedback on the chosen answer

Could have
- The minigame could have a stand with an NPC asking the questions
- The minigame could have the two players standing on a stand.

Won't have
- Open questions
- Hints
- A timer for answering questions

Power eco-friendly game

Must have
- The player must be able to rotate parts in the puzzle
- The minigame must have servers
- The minigame must have a source point
- The minigame can only be won when the right amount of energy is reached
- All servers must be reachable from the source
- The game must have different energy sources

Should have
- The game should be able to generate a random puzzle
- A hint in the form of a piece of the board is shown
- Pieces that are brightly colored
- Each type of energy source in the minigame should have a different energy supply

- The minigame should show a visual progression

Could have
- The player could be able to see the number of clicks
- Highlight the paths that are already connected
- A bar with the current amount of voltage

Won't have
- Pipe animations whenever a pipe is connected

Connection game

Must have
- The player must be able to rotate parts in the puzzle
- The game must have servers
- The game must have a source point
- The game can only be won when all servers are connected to the source
- All servers must be reachable from the source

Should have
- The game should be able to generate a random puzzle
- A hint in the form of a piece of the board is shown
- Pieces that are brightly colored

Could have
- The player could be able to see the number of clicks
- Highlight the paths that are already connected

Won't have
- Pipe animations whenever a pipe is connected

Data priority game

Must have
- The minigame must have icons representing different data
- The minigame must be controlled by two players
- The minigame must have a timer
- The minigame must allow for players to collide with an icon
- Each icon must be worth points

Should have
- The minigame should visualize the score
- The minigame should visualize a hint when a player hits a bad icon
- The minigame should have a target value

- The players should walk through each other

Could have
- The minigame could display the scores of the icons when the player collides with them

Won't have
- The players will not have sliding physics

Password game

Must have
- The minigame must have a box to type a password
- The minigame must have a button to submit the password
- The minigame must show instructions for the creation of the password
- The game must not allow passwords that do not fit the description

Should have
- The minigame should give feedback when a password is entered that does not fit the description
- The minigame should have an explanation after each submit, explaining how long a hacker will take to crack the password
- The players should re-enter the passwords
- When the players re-enter the passwords and they made a mistake 3 times, they should get a hint

Could have
- The minigame could have animations of a hacker hacking the password

Won't have
- Password will not be hidden
- The passwords won't be used later in the game

Appendix D: Learning Goals

- In the main game
    - I remember that data centers are fundamental to the Internet
    - I remember that data centers are indispensable for the future
    - I remember the main tasks of a data center
    - I remember the importance of data centers
    - I remember that Interxion is a company that hosts data centers
- Concepts Game: The game with the element to combine
    - I can name elements a data center is composed of
    - I can differentiate different elements of a data center
    - I can define a data center as the home of the cloud
- Connection Game: The game with connecting all the servers
    - I observe that individual servers are part of a whole system
    - I understand that servers are an integral part of data centers
    - I can solve a mathematical puzzle problem
- Data Priority Game: The game where players need to catch good types of data
    - I remember that data centers are fundamental to the Internet
- Power Eco-Friendly Game: The game with connecting the right power sources to reach a certain voltage.
    - I understand that power comes from different sources
    - I understand that sustainable energy resources have less environmental impact than nonrenewable resources
    - I can solve a mathematical puzzle problem
    - I remember that data centers (partly) run on sustainable energy.
- Password Game: The game with the passwords
    - I can explain the importance of security
    - I can explain the constraints that influence password strength
- Marketing Game: The game with creating slogans
    - I remember that maintaining a data center requires expertise from diverse fields
    - I can motivate why different aspects of a data center are important

Appendix E: Project Plan

In agreement with the project supervisor the project plan was created as a slideshow in dutch. This appendix is a translated, textual representation of this project plan.

Background information (provided by the client):

- Large gap between the amount of men and women in STEM fields
- Small amount of female role models
- Stereotypical associations based on gender
- Uncertainty for women regarding capabilities in STEM
- Women lose interest in STEM fields around the age of 12

Goal:

- Create a game to:
    - Increase interest in IT related fields
    - Increase knowledge about data centers
    - Demonstrate the applicability of STEM
    - Break stereotypes
    - Decrease the negative view on women in STEM
- For the target audience:
    - Children between the ages of 10 and 14
    - Focus on female students in pre-vocational education

Research question:

- What kind of game would create a positive attitude towards IT and data centers, where the focus of the game lies on girls between the ages of 10 and 14, that at the same time stimulates them to pursue a career in STEM?

Sub-questions:

- How do educational games work?
- What kind of games work well for the specified target audience?
- How can stereotypes, especially with regards to women and STEM, be broken?
- How do we make the game gender inclusive?
- How do data centers work, what are the tasks of data center employees and how can these concepts be translated to a game?

Requirements/Limitations:

- A maximum playtime of 1 hour
- Deployable and playable on low end (school) machines

- Art is outsourced
- The language of the game should be dutch
- Development time of approximately 6 weeks
- The game should have a positive attitude towards STEM

Planning

Phase one: Research:

- In the first phase, information about gender diversity, educational games and the relation between these two will be researched. Furthermore different types of games will be researched to determine which type best fits the needs of this project. After the decision for the game type is made, the best technologies to develop this game type will be chosen.

Phase two: Design:

- During the second phase, the team will determine the technical feasibility of the chosen solution and develop a blueprint for the project. MoSCoW will be used to compose the functional requirements and the non-functional requirements will be determined as well. The repository, as well as a base for the project, will also be set up.

Phase three: Development:

- Continues development will be used while meeting with the client on a weekly basis and processing the given feedback. The graphical design of the game will be outsourced.

SCRUM

- The development, integration and documentation of the project will be done through SCRUM. This will be an essential part of phase 3. The team will meet on a weekly basis to determine the work for the next week. A retrospective will also be created to create an overview of the problems encountered the past week and how to solve them in the future.

Weekly planning:

Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9 / 10
Set up meetings	Research report	Unity opzetten	Initiële rooms maken	Dialogues start minigame	Marketing game	Customisations	Animaties	Final Report
Research question	Meeting	Repository opzetten	Mechanics concept game	Data priority game	Games tutorial prefab	List progression	Storyline	SIG delivery
	Research	Unity tutorial	Mechanics connection game	Server connection game	Password game	SIG feedback	Fine-tune minigames	Presentation TUD
		Programming skills	Meeting	Conceptual Design	Integration	Power connection game	Integrate sprites	Presentation Interxion
		High level design		Concept game playable	Full game map	Demo meeting Interxion	Game tutorial prefab	Meeting
		Requirements		Quiz game playable	Textures	Meeting	Playable demo	
		Meeting		Counting Sprites	Quiz game		End 'cutscene'	
				Marketing playable	SIG Delivery		Field test (optional)	
				Meeting	Meeting		Draft report	
							Meeting	

Phase four: Presentation:

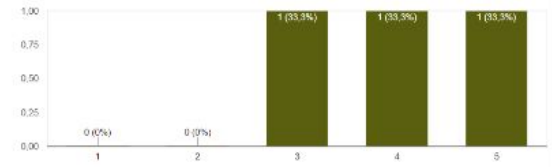
- In this phase the final product will be presented. The final version of the game will be shown through a demo and an overview of the process will be given. A separate meeting with VHTO, as well as Interxion, will be organized to give the client the opportUnity to discuss anything without the time limitation of the presentation aimed towards the project supervisor and bachelor coordinator. The final report, as well as the software product will be handed in to both the client and the supervisor.

Appendix F: Questionnaire 1

Power eco friendly

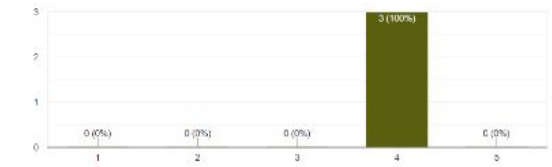
How educational is the game?

3 antwoorden



How entertaining is the game for kids?

3 antwoorden



Worst and best aspects of the game?

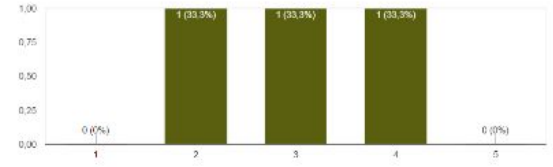
1 antwoord

no shown penalty for if something bad happens. Is there music?

Marketing game

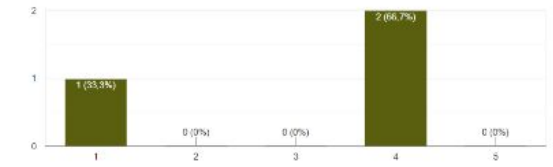
How educational is the game?

3 antwoorden



How entertaining is the game for kids?

3 antwoorden



Worst and best aspects of the game?

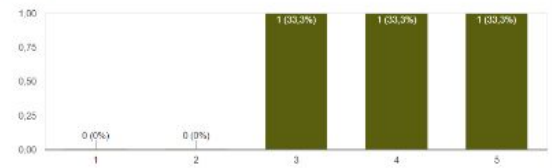
1 antwoord

creativity

Quiz game

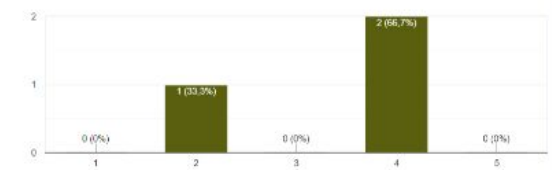
How educational is the game?

3 antwoorden



How entertaining is the game for kids?

3 antwoorden



Worst and best aspects of the game?

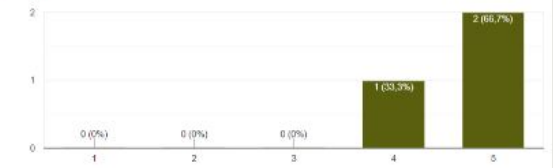
1 antwoord

Winning

Password game

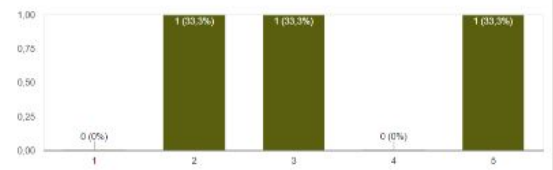
How educational is the game?

3 antwoorden



How entertaining is the game for kids?

3 antwoorden



Worst and best aspects of the game?

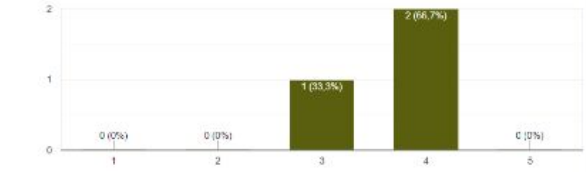
1 antwoord

Memory training but also real life value

Cooling game

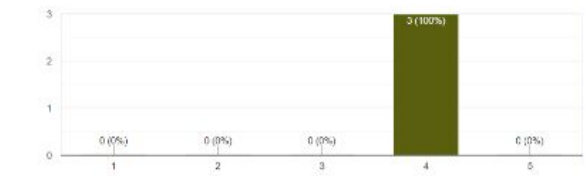
How educational is the game?

3 antwoorden



How entertaining is the game for kids?

3 antwoorden



Worst and best aspects of the game?

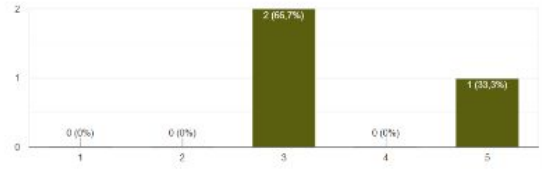
1 antwoord

multiple elements to take account for. no bad things

Security game

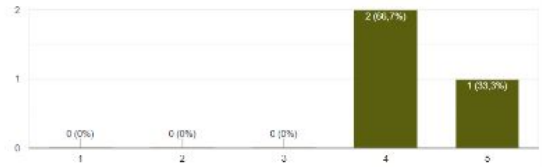
How educational is the game?

3 antwoorden



How entertaining is the game for kids?

3 antwoorden



Worst and best aspects of the game?

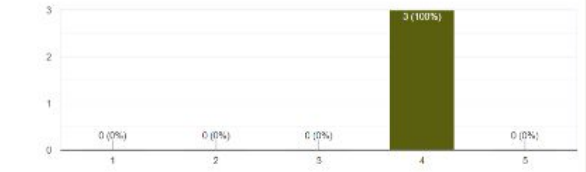
1 antwoord

Difficult on a screen maybe, but educational

Generator start game

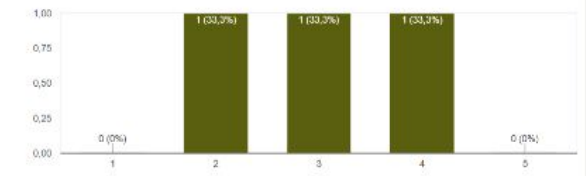
How educational is the game?

3 antwoorden



How entertaining is the game for kids?

3 antwoorden



Worst and best aspects of the game?

1 antwoord

Maybe too simple

General question

How else to support gender diversity?

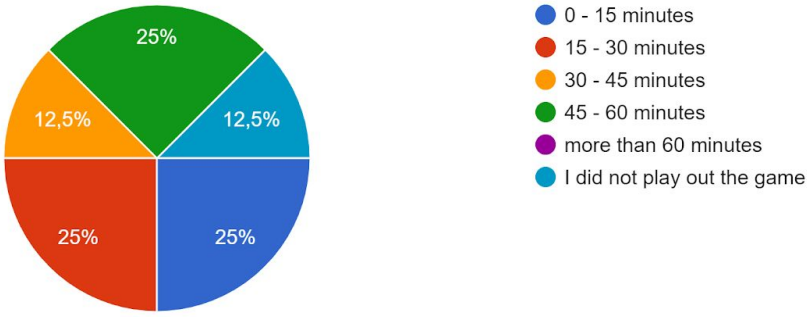
1 antwoord

Just not talk about gender at all. This helps to form a free opinion

Appendix G: Questionnaire 2

How long did you take to play out the game?

8 antwoorden



For the overall game, how difficult do you consider the game for our target audience, why

Difficult enough, errors could be made but the game would help you when you got something wrong

Moderately difficult, i think with an intro of the game it is going to be easier. But overall it is quite doable.

I think it's ok but if you do it to long it will get boring

I think it could be more difficult, because the game was a bit too short or my liking.

Quite complex, due to the target audience probably not wanting to read every explanation

It's oke

Average to difficult. It requires some critical thinking, which is good, but for some it can be hard. They need to read the instructions carefully.

I think the game is fine for the overall target group, it's not that hard, I think the different keys to press to move is a bit weird maybe instead of W better should be the key F. easier for the hand coordination. But overall creative and have to read directions which is good for younger players

What in the game was unclear?

The green energy minigame was for me a little bit unclear

The exits of the different rooms were unclear for me. For non-gamers it can also be unclear how to move both characters.

In the first mission I didn't know what you needed to combine it.

That there was a room on the right.
It was also difficult to know when to go back to the office

in game 2, what icon the network was. I needed the last picture to replicate the example to finish it

A character tells you to go to an upper left room, but there is no door there.
In the game with the password, it said special characters, but it would be nice to have a set of these characters.
The information given before a game can be misleading, I was clicking on things, but it was only to show how to. Maybe begin a sentence like, in the game...
It is not clear what I learned from the game with green energy.

The Energy minigame. The puzzle was a bit unclear, what to connect and where to start. Mostly what the icons are as well.

the second round where u have to go to someone and u enter their office i think that was a little unclear on what I had to do precisely

What is your opinion on the game's fun factor? Please elaborate in as much detail as possible.

Feels like the old pokemon games which i liked a lot when k was the target audience's age

It is a very fun and educational game. The joke about the cat at the end increases the fun factor.

It's fun but for 10 minutes. Me as a kid that hates it to play games for more than 10 minutes. Does like it you always need to do something but sometimes it's to much so i will give it a 6/10

I liked the first game, it was informative while remaining to be a game. The 2nd one just became annoying to finish, the point was clear, it shouldn't need to take that amount of time

It is very fun. I really enjoy the concept of the game. I hope to be able to choose better avatars in a later version.

The game looks fun. It's not that lineair, which is nice. I think the way the game is designed (Pokemon-is) speaks to the target audience. I think they will find the different minigames fun and helpful.

I like that it's similar to Maze game so it makes it more interesting to see where you have entered to

get out of the rooms.

What are the things children will have learned from this game?

What a data center entails

What data centers are, what their function is, how data is stored and how important passwords are.

basics of data centers

The links to a data center, how important passwords are.

Mostly the ins-and outs of a data center. What goes on in there and the broad scale of maintenance that is necessary.

I think the main thing I can get out of this game is that they will learn to follow directions better.

To what extent did this game improve your perception of data centers and how could this effect be enhanced?

Not much

Beforehand I did not precisely know what data centers were and now I could actually explain to someone else. This effect can be enhanced by maybe giving more practical examples. For example relating data centers to the effect it has on the life of a 12 year old.

I didn't play it to the end

Not much, it could be enhanced by including the information in the minigames.
For example, in the password game you could explain why hacking those passwords would take longer.

By trying to make things clear without taking lots of time to complete the game

The password game was the most clear. It explained midgame what dangers are and let you update something. You really learn while playing. And at the beginning the goals were very clear.

It was pretty clear how a day in the life of an engineer at a data center looks like. This effect was pretty much visible with all the different things you had to do to make sure the company was working properly. I think this effect was already pretty much visible. Maybe it can be enhanced through more different tasks and minigames, but in my opinion it was already pretty diverse.

I am not really a gamer so it was a bit new to me, so really can't say

Which minigame is the weakest and how could we improve it?

The green energy game it wasn't clear to me what I had to do

I did not find one of the minigames weak.

The minigame where you connect the servers was too easy/ fast.

2nd, make it smaller

The 2nd game, clearer description of what is expected and why.

The Energy game (2nd one) Maybe a bit more instruction as to the icons (what they mean) and where to start exactly.

Which minigame did you enjoy the most and what aspect of it specifically?

Passwords/ the hacky feel

I like the connection game because of its simplicity.

The minigame where you connect the data center to the energy sources.

the first, see previous answer

catching things from the sky

The first game that you have to connect different elements. It required a bit out of the box thinking and logical thinking at the same time. It's basically the essence of what a data center is and everything that is needed.

the minigame with combining elements, i just like the fact that you had to really analyze each symbol.

Other feedback?

Maybe elaborate on why some things like complexity of passwords are important and make the final quiz test the knowledge gained in the game

- In the first minigame: If something is occupying one of the boxes, you first have to take something out of them before you can put something else in.

- In the beginning of the game the screen was a bit too dark

- The first sentence the doorman says is not correct. :)

Block out the cafeteria, is an useless room throughout the game to go to and look for things to

progress

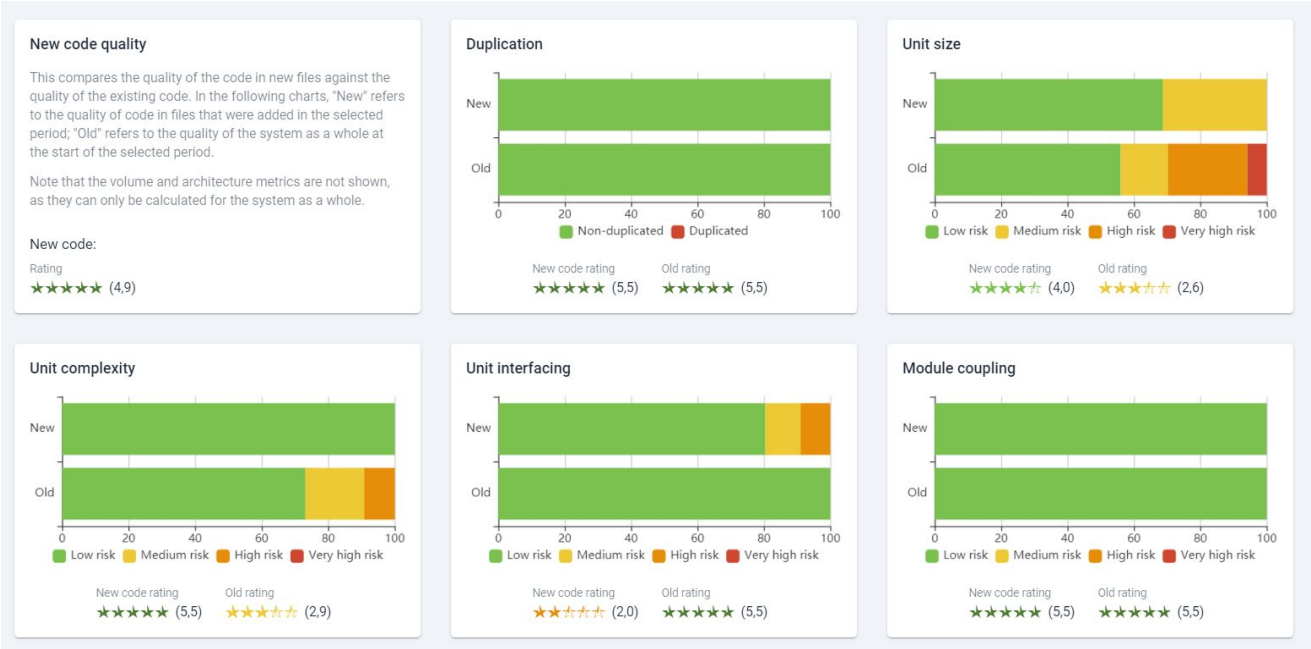
There is no real real ending. The other characters are saying things about the cat, when you continue walking. A big notification with GAME WON would be nice.

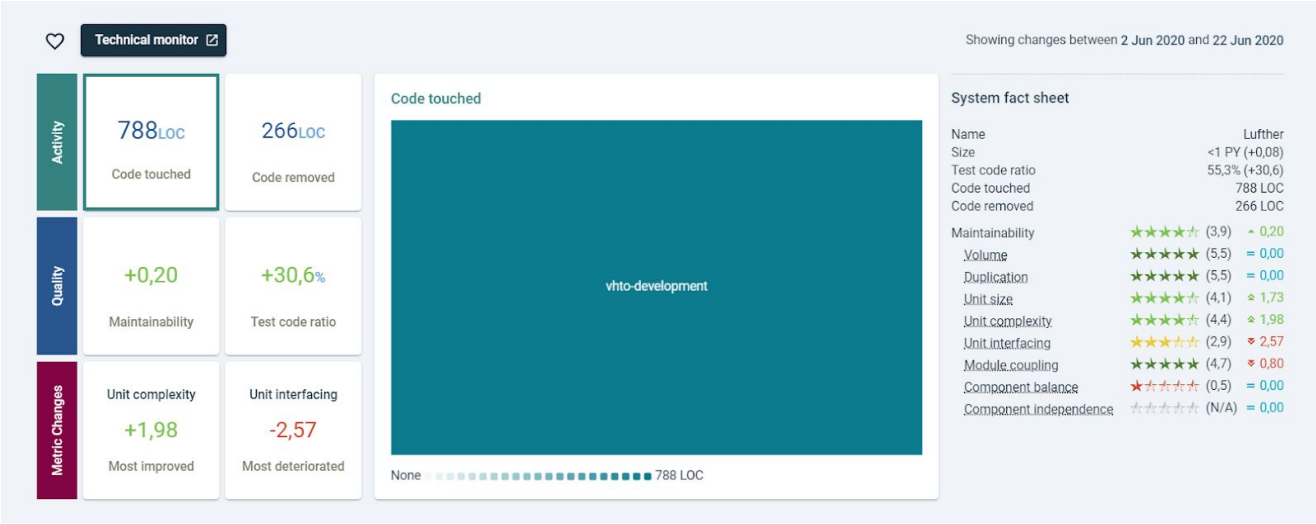
The game with slogans can be a bit more. It seems too simple. You can just fill in the word with another letter.

Love the concept of the game. The diverse minigames are sweet as well. Maybe to promote multiplayer and co-operation, make the minigames a bit more multiplayer oriented. Maybe take turns or something like that.

goodluck

Appendix H: SIG Feedback





Appendix I: Dialogue .txt commands

The dialogue system parses .txt files and changes the dialogueBox according to these files.
The desired .txt file has to be added to a dialogue component.
This dialogue component should be added to an object.
Multiple commands can be used within the dialogue files, which will be explained below.

*STAGE:
To indicate dialogue that belongs to a certain stage of an NPC the -Stage:x command is used (This command can be seen as a pointer).
The stage should always end with a -EndDialogue or a -UpdateStage command.
x is in this case replaced with the number of the stage.
In most cases x starts at 0 and increments by 1 per stage. If an object or NPC reaches an unspecified stage, the dialogue box will automatically close.

For example:
-Stage:0
-Stage:1
...
-Stage:46

*SUBSTAGE:
Different dialogue options might lead to different dialogue lines, however, these lines will be contained within the same stage.
Therefore pointers within a stage have to be added. This is done through -Substage:x
x counts from 0 and increments by 1 per stage.
This is extremely important since these numbers refer to array indices and the count resets per stage.

*CHOICE:
There are 2 types of dialogues boxes. One handles normal strings, with the option to either proceed or close the dialogue.
However the second dialogue box offers the player TWO dialogues options (The quantity has been limited to two, to save development time).
To use the second type of dialogue box, containing the choice, the command -Choice is used.
The question that should be displayed in the choice dialogue box is specified before the -Choice command.
-Choice is followed by 4 lines:
2. The text that should be displayed within the first option button (mind the length).
3. The action that should be added to the first option button.
2. The text that should be displayed within the second option button (mind the length).

3. The action that should be added to the second option button.

An example:
Can I help you with something?
-Choice
Yes
-GoToSubstage:5
No
-EndDialogue

*GOTOSUBSTAGE:
As explained earlier, pointers within a stage can be added through -Substage:x
To access these substages the command -GoToSubstage:x can be used, where x corresponds to the x of -Substage:x.

*ENDDIALOGUE:
-EndDialogue can be used to close a dialogue, without updating the current stage.

*SCENE:
To change the Unity scene from a dialogue option -Scene:x has to be added.
x is the name of the desired scene, which is case sensitive.
Also make sure that there are no spaces, etc. after the scene name, since this will be included within the name when parsed.

*UPDATESTAGE:
-UpdateStage is used to close the dialogue and update the NPC dialogue stage.

*FORCEDIALOGUE
-ForceDialogue can be used in the case where the player should not be able to close the dialogue box.
For example after a minigame.
-ForceDialogue should also be used in most cases where the dialogue contains an -UpdateStage command, since this might mess up global NPC updates otherwise.

Appendix J: Research Report

2 May 2020, Delft University of Technology

Gedeon d'Abreu de Paulo
Bilal el Attar
Ali Al-Kaswan
Nathan Kronstadt
Gianni Wiemers

Table of Contents

1. Introduction

Technology is a rapidly growing field, with an increasing need for engineers and scientists. Unfortunately, there is a global phenomenon of the underrepresentation of women in science, technology, engineering, and mathematics (STEM) studies and careers (Booy, 2012). This is also called a “leaking STEM pipeline”, where the interest in STEM declines throughout education. The aim of this paper is to propose a design of an educational game (EG) that will stimulate interest in a STEM field, using available literature as a basis. This paper will present a background on the problem at hand, an assessment of the game design, and a comparative analysis of ideas for games. The results of the analysis will yield a game that will be developed. Based on this game idea various technologies and methodologies will be evaluated in order to best develop the final product.

2. Problem definition & analysis

Women are underrepresented in STEM-fields, especially in the Netherlands (Booy, 2012). Scientific studies have shown that girls tend to turn away from science subjects around the age of 14 (Booy, 2012). VHTO is the Dutch national expert organisation on women in stem.

Together with VHTO, Interxion also plays an important role in this project. Interxion is a company that provides carrier and cloud-neutral colocation data center services. They believe that data centers are relatively unknown to children and this field should get more attention.

2.1. Problem analysis

In order to balance out the problem of underrepresentation of women in STEM, Interxion and VHTO came up with an idea to create an educational game to attract more people, especially girls, to technological fields (e.g. data centers). This game has two main objectives, directed towards children at the age of 10 to 14 and the game should be playable in one hour, since the game will be used in class:

- The game should create a positive attitude about technology, with a focus on girls in pre-vocational education. This way, girls could be stimulated to pursue a career in STEM.
- The game should teach children about data centers; how data centers work and their importance (in the future).

Since this may be too broad, the approach focuses on IT and data centers.

In order to accomplish these goals, the following central research question has been formulated:

What kind of game can help create a positive attitude towards IT and specifically data centers, with a focus on girls between 10 and 14, in such a way that these girls are stimulated to pursue a career in STEM?

To develop such a game, non-functional-and functional design goals have been set. These are described in chapter 4. In order to solve the problem, the following sub-questions have been formulated:

- *How can we break stereotypes regarding women and STEM?*
- *How are educational games used?*
- *What kind of games are popular for young children, especially girls, between the age of 10 and 14?*
- *How can one make a game gender-inclusive?*
- *How do data centers work?*

Based on the results of the sub-questions the central research question will be answered. In light of the result, a game will be chosen to develop. This report covers the design choices of the chosen solution.

3. Objectives of the educational game

In order to reach the two objectives of the game, described in section 2.1, a list of elements that need to be taken into account has been formulated. These are as follows:

- The game should create a positive attitude towards data centers and IT
- The game should be gender-inclusive
- The game should be suitable and fun for the target audience; children in the age of 10-14 with a focus on girls in pre-vocational education
- The game should explain the workings and importance of data centers

These goals are described in this chapter, but first the concept of educational games must be explored.

3.1. Educational games

An educational game (EG), also known as a serious game, is a game that supports a learning objective (Romero, 2019). These learning objectives are determined before creating the EG and can be constructed according to the taxonomy of BLOOM (Bloom, 1956). In this day and age this taxonomy is still used by teachers and schools. The taxonomy consists of the learning goals: remembering, understanding, applying, analyzing, evaluating, and creating. By using clearly stated objectives, the players will be able to succeed better in acquiring the knowledge.

3.1.2. Purpose of EGs

“EGs permit the development of playful learning situations related to disciplinary areas and to 21st-century skills such as collaboration, problem-solving, and creativity” (Connolly et al., 2012). According to Prensky (2001), “EGs are based on analogies of a real-world situation, making

learners able to play with the concepts in order to get a better understanding". However, in order for the learners to understand the analogy, it is important to give clear boundaries and to clearly link back to the concept (Sentance, Barendsen & Schulte, 2019)

3.2. Change in attitude: breaking stereotypes regarding STEM

One problem which forms a clear plague to the scientific world and deserves attention, is the representation of women in STEM fields. STEM includes a wide range of fields, and the level of underrepresentation of women among these fields varies greatly (Meyer et al., 2015). Specifically, fields for which success is commonly thought to be achieved through raw skill and brilliance (e.g. math-intensive fields), rather than effort and practice (e.g. life sciences), are correlated with a higher degree of underrepresentation. This thought, combined with stereotypes against the mental faculties of women held by teachers, parents and peers can lead to women opting not to choose for these fields, as they may believe they do not have the capacity (Meyer et al., 2015). The resultant homogeneity and lack of diversity can have clear negative effects, possibly leading to inaccuracy in processing information and a lack of objectivity in making decisions (Apfelbaum et al., 2014). With more diversity, individuals can be confronted with social differences and thus more easily consider a different perspective (Apfelbaum et al., 2014). From the above, it can be seen that stereotypes negatively affect the representation of women in STEM and that this underrepresentation can have wider implications on decision-making and information processing.

3.2.1. Causes and solutions to stereotypes

Stereotypes can impact the consciousness of women starting from a young age, thereby affecting their concept of STEM and their attitude towards it. This may be reinforced by the media, where the skill and experience of women scientists are often being downplayed. For example, the highlighting of their femininity more than their technical expertise, their portrayal in more supportive roles in the laboratory rather than as leaders, the mentioning of the struggles associated with being a woman scientist and motherhood and the presenting of science as requiring masculine traits such as emotional detachment and physical strength (Steinke, 2016). Thus, when portraying women scientists in the media, it is important to avoid these pitfalls. In order to foster a more positive image, women should be represented in distinguished roles that clearly focus on their expertise as a scientist rather than displaying aspects such as domesticity and their femininity, and in a way that does not portray the scientific world as requiring masculine traits. In a study done by Bodzin & Gehringer (2001), two fifth-grade classes were asked to draw a scientist and were then visited by physicists, who discussed their jobs. After this visit, they were again asked to draw a scientist, and the image they drew clearly changed, suggesting that stereotypes can be directly affected by exposure. Thus, a way to tackle stereotypes, especially at an early age, could be to expose students to informational and gender-neutral informational depictions of STEM. An example of such an exposure is the use of classroom videos, which can help increase interest in STEM for both male and female students (Wyss, 2012). In these ways, stereotypes can be combated, leading to a more egalitarian and equal view and state of STEM fields.

3.3. Gender inclusion and girl-centric design

The goal of the game is to increase gender diversity in STEM, and it is meant for a diverse audience with a focus on girls in pre-vocational education. Thus, designing a solution will need to keep two human-centered design principles in mind; culturally responsive-and girl-centric design (Sammet & Kekelis, 2016).

3.3.1. Culturally responsive design

The first premise of a culturally responsive design process is that there are no homogeneous, gender-neutral end users of the program (Sammet & Kekelis, 2016). This means that the game should be explicit with respect to the end users also being girls.
An important part of making culturally responsive design is gender inclusion. The Hypatia project, a project that tries to engage teenagers in STEM fields, has designed a framework of gender inclusion criteria (Achiam & Holmegaard, 2016). This framework contains the following criteria that correspond to the scope of the project:

- *"Avoid presenting learners with strongly gendered activities that may contribute to the internalization of 'female' or 'male' identities".*

The game should avoid presenting strongly gendered activities. An example of this is the interest of women with regard to computer science, where they tend to be less interested when the learning environment includes stereotypical objects, such as Star Trek posters or video game boxes (Sammet & Kekelis, 2016).

- *"Ensure that the diversity of science is represented to the largest extent possible in the activity".*

Since there will be dealt with a diverse audience, the diversity of the subject matter should be presented.

- *"Challenge learners to depart from their preferred interests and widen their engagement in science".*

Stereotypes often play a role from the learner's perspective. A game designed for inclusion should challenge these views.

- *"Ensure that the activity has a balanced approach to participants' learning preferences".*

Since everybody has a different way of learning, the solution should take a diverse method of teaching into account.

3.3.2. Girl centered design

Girl centered design means that the design of the game should be approachable to girls. This does not mean that the game should be in pink bright colors, but rather take into account how girls learn certain subjects and how girls could become interested in STEM subjects. An example of this is the method of presenting certain content. If physics is presented in a technical, mechanical or electrical context and if it is spectacular, violent or explosive boys

generally find it interesting, while the interest of girls lay on different contexts such as health, the human body, medicine, ethics and aesthetics (Sjøberg & Schreiner, 2012).

3.4. Target Audience

A survey on the preference of genres (GameTree, 2019) shows that women strongly prefer casual, adventure, and party games, while their male counterparts showed a strong preference for action and sports games. Strategy, fighting and role-playing games are commonly favored between both genders.

Games feature one or multiple motivating factors that push the player to play the game. When comparing the motivation factors with gender, women seem to favor games that have strong relationship factors (Yee, 2006). Another study by (Tekofsky et al., 2017) shows that women seem to prefer games with fantasy, story, and relationship elements. Similarly (Hartmann & Klimmt, 2006), found that women tend to prefer role-playing, adventure, and peaceful build-up games (e.g. The Sims).

GameTree (2019) identifies that younger audiences (aged 13-17 years) have a slight preference for fighting and action games while showing a negative trend towards casual, strategy, and role-playing games. This difference might, however, be caused by the platform preferences of this age group. This age group gravitates towards mobile platforms, while role-playing and strategy games are very rarely released on mobile platforms. This trend doesn't continue for older groups, the age group 18-22 years old shows a slight positive trend towards role-playing, strategy, and casual games (GameTree, 2019).

A study on game preferences of middle school children (aged 12-14) in the United States (Kinzie & Joseph, 2008), shows that girls of this age group show a preference towards creative and explorative games, while boys prefer active and strategic games. To cater to both girls and boys, the researchers recommend emphasizing explorative elements, as seen in most role-playing games. Even though girls prefer this genre more than boys, children overall show a strong preference for this genre (Kinzie & Joseph, 2008).

Problem-solving play, as seen in puzzle games, while not as popular as explorative play, shows no bias in preference between boys and girls. This genre is also relatively good for incorporating other concepts to teach to children (Kinzie & Joseph, 2008).

3.5. Data centers

One of the goals of the final product is to educate the users about data centers. Data centers store and process data for all different kinds of applications (*Dutch Data Center Association*, n.d.-a), from bank payments and public transport, to social media and music/video streaming. These data centers, often also referred to as *the cloud*, rent servers to companies. Bigger companies such as Facebook or Google often have their own data centers. These data centers

are mostly located in countries with good digital infrastructure and high internet speeds since this drastically increases the speed of the servers (*Dutch Data Center Association*, n.d.-b). Since many vital applications that are used on a daily basis, by for example law enforcement or hospitals, are available due to these data centers, it is extremely important that they keep these services available at any time. An in depth explanation of four main aspects of data centers that play a key role in this availability will be explained in the next sections.

3.5.1. (Cyber)security and safety

The security of data centers is extremely important since the servers within these data centers might contain personal information or keep vital applications up and running. These servers do not only have to be protected against cyber-attacks, but there are also a lot of physical security measures in place to keep them safe. Security guards, receptionists, and surveillance cameras are often used to monitor the activities in and around the data centers. To keep unwanted guests from accessing the building, data centers are often surrounded by fences with security checks. Furthermore, to ensure that only authorized personnel access the servers, employees often have to authenticate themselves with key cards or biometric scanners when entering a certain room or area within the data center.

Another point of attention within data centers is safety. There are a lot of machines that require a lot of energy, packed closely together. Although these machines can generate quite a lot of heat, the risk of fire originating in one of the servers is very low, since these machines have built in measures that shut the machine down before ignition (*Risk Considerations for Data Center Fire Protection - SFPE*, n.d.). The greatest risk of fire within a data center in fact comes from miscellaneous items. To prevent a fire from growing and keep the equipment protected during a fire, fire detection systems are in place that cut the power to specific server bays when smoke starts to develop. On the occasion that a machine does catch fire, extinguishing systems are in place which contain a gas that lowers oxygen levels to extinguish the fire.

3.5.2. Server rooms

The server rooms are the heart of the data center. This is where all the data is stored and where all the computations are done. These server rooms contain three different bays. The first being the server bays that contain the processors, graphics cards and other components used to process the data. The second type of bays are used for storage, these bays contain a lot of hard drives. The last type of bays, called administration bays, manage the storage and server bays and the network.

3.5.3. Power

Power and cooling are two very important aspects of data centers as well. Data centers around the world used 200.000 GWh in 2015 (Andrae & Edler, 2015), which is equal to the consumption of roughly 18 million U.S. households (U.S. Energy Information Administration (EIA), n.d.). As mentioned before, it is extremely important that these data centers are available all the time, which is why every server has two power supplies, one being the regular electricity

net, the other being a backup generator in case of a failure. This high usage of electricity is caused by two sources, one is running the servers and the second is the cooling of the servers.

Cooling

As mentioned in the previous section, these machines can get quite hot. To increase the performance of the servers and reduce the risks of fire, it is extremely important to cool the servers. This cooling is often done in one of two ways, either by air, fans push the hot air out and blow cold air in, or by water, where cold water flows through hot areas within the server and absorbs the heat. To decrease the amount of energy that is consumed, green energy sources such as solar panels or seasonal thermal energy storage are often used.

4. Solution

To create a solution, the functional requirements of the final product will first be composed. These functional requirements will be the base for the formulation of the non-functional requirements.

4.1. Functional requirements

Creating an EG that helps children, in the age of ten to fourteen, develop an interest in IT, and gain knowledge about data centers is what will be the desired result of the project. The next sections will elaborate on the development of the game.

4.1.1. Game objectives

The game objectives give a clear view of what the game should be able to do. A child considers winning a game as the main objective (hello Motherhood, 2018). However, the game in development will consist of more game objectives. Each of the objectives will be discussed with experts to find out what the best way is to achieve the goal. The goals will be measured through scores, time, and a quiz at the beginning and end.

Fun

The main goal of the game is for the kids to enjoy playing the game. The feeling of engagement and the experienced fun of the players is the most important aspect for the EG in development. The players are more likely to be interested in the game if they have a very positive emotional connection to the game (Levine & Pizarro, 2004). The game will achieve the fun goal by choosing a game type that is of interest to the target audience and by adding fun game elements.

Educational

The game should encourage the player to learn by playing. The better the players understand the concepts in the game, the better they will remember them (Berg & Petersen, 2013). To

achieve this, the concepts within the game need to be clear and understandable for children in pre-vocational education. According to (NRO, 2016), a game with clear instruction will have a positive effect on the learning process. The best way to express these concepts in the game will be discussed with experts in the field of computer science and social science. Another important point is that no prior knowledge of data centers is required to finish the game.

Challenging

To continuously engage with the game, players need the game to be fun but also challenging. “A game or activity can quickly lose its draw when a child is no longer challenged” (Hello Motherhood, 2018). The target audience for the game is children between the age of ten and fourteen in pre-vocational education. In section 3.4, it has been discussed what games would best fit the target audience according to literature.

Winning

Winning the game is not the main goal of the game, but it is a goal that children cherish. The game needs to be playable and winnable within 45 minutes. A feeling of winning can be achieved in multiple ways. The game in development will achieve this either by completing the games challenges or by having a high score. According to NRO (2016), pre-vocational education students that have prior knowledge on the subject learn more when they are in competition with others. Pre-vocational education students that have no prior knowledge prefer working in groups to gain knowledge (NRO, 2016), they will have a winning feeling by getting to the finish line with the team member(s).

Behavior change

The game should provide a behavior change towards IT and data centers. As the game gives rewards (e.g. points, credits, or gifts) for certain behavior or choices in the game, this can stimulate a positive attitude towards IT (Lieberman, 2006). According to Lieberman (2006), players tend to identify themselves with the characters, meaning that success and failure are mapped directly to the player's ego and self-image in the experience. The goal of the game would be to optimize this aspect by not allowing for heavy punishments for failures. This is also an aspect that is discussed in section 4.1.5.

4.1.2. Possible solutions

To explore the possibilities with regard to the final product, three game ideas have been explored based on the findings of the previous chapters and the wishes of the client. The three ideas and the final choice will be explained in this chapter.

Builder game

Peaceful buildup games were one of the preferred game genres by women (Hartmann & Klimmt, 2006) and at the same time have a big creative aspect, which is also preferred (Kinzie & Joseph, 2008). The different components needed to create a successful data center can be

explicitly shown through a builder game. Through random events various hazards or technical problems that can occur in a data center, and the tools that can be used to mitigate or prevent these, can be demonstrated. Another advantage of a builder game is the fact that there is a visual progression, the data center will expand the further they get into the game. However, one of the downsides of a builder game is the fact that there might be no clear ending, because an important aspect of game design is having a clear, quantifiable end goal (Salen et al., 2004). Other building games like Zoo Tycoon[11] can go on for a very long time. Since the game will only be played for up to an hour, it should have an ending where the users feel like they are finished with the game.

Figure 1: Zoo Tycoon: An example of a builder game. Retrieved from https://themeparkfreaks.eu/pretparkgames-zoo-tycoon-2/

2D RPG

As mentioned in section 3.4, one game element that is interesting for both boys and girls between the ages of 12 and 14 is exploration (Kinzie & Joseph, 2008). With a 2D role-playing game (RPG), the users will be able to walk around freely, which increases the exploration factor of the game. The user can learn about specific parts of a data center by interacting with game characters or playing minigames. However, if too much freedom is given to the user, the game might become unclear. Another problem can occur when the various tasks the player has to complete do not have any form of coherence, it might not be very interesting if there isn't a clear end goal that has to be achieved.

[11] Zoo Tycoon (2001), a builder game developed by Blue Fang Games https://www.mobygames.com/game/zoo-tycoon/

Figure 2: An example of a 2D RPG. Retrieved from https://arktentrion.itch.io/2d-rpg-kit

Cluedo-like escape room

Games with story elements are also an effective choice when trying to keep women engaged (Tekofsky et al., 2017). Cluedo[12] is a game where a murder mystery has to be solved by finding various clues in various rooms. This concept can be translated into an educative data center game, where users have to find 'clues' within different rooms or locations in and around a data center, with which they have to solve some mystery. A risk with this approach however is the difficulty of clues for our target users. This could mean that they could get stuck at a certain point in the game, not being able to finish the game.

4.1.3. Final Product

Each game type has its own advantages and disadvantages. To come to a final idea, concepts from the different ideas have been combined into one. The base of the game will be a 2D RPG, since the exploration factor that comes with this type of game will interest both boys and girls equally. To prevent the lack of coherence that can come with this type of game, and to include puzzle elements, the story elements of the Cluedo-like idea will be integrated into the game. To prevent users from getting stuck, the story will automatically progress after playing a minigame. Additionally, clear instructions about where to go or what to do will be visible at all times. To increase the incentive to advance within the game, a form of visual progression, as is also present in the builder game, will be added to the final idea. These aspects combined will help form the final product.

4.1.4. Game plot

Levi is the manager at Interxion and has the suspicion that someone broke into the data center. Levi does not know who it is and what they want. However Levi is pretty sure that the intruder is still inside of the data center and wants him or her caught. The player is an engineer at Interxion, when the player arrives at work Levi explains the situation and asks for their help. While looking for the intruder there are various problems in the data center that the player as an engineer will have to solve. The player has to make sure to capture the intruder before they escape.

[12] Cluedo, a murder mystery game by Hasbro: http://www.cluedo.com/

4.1.5. Mechanics

The game is played by two players on a single machine. These two players will be able to walk individually through the game, but will solve mysteries together. They will not be able to walk too far away from each other.
Within the game, there are minigames that allow the player to achieve the learning objectives. These minigames will be related to the storyline of the game. The minigames will mostly consist of cognitive learning activities (e.g. memorize, structurize and relate). Walking around in the data center will also be a learning activity, as the player gets the chance to explore the data center.

To create a gender-inclusive game, players can choose which gender they want to play with. As for the points of gender-inclusive design made in section 3.3., the minigames will introduce the diversity of the subject and also accompany multiple teaching methods. On top of that, to combat stereotypes from the learner's experience, female engineers will be shown to the players. As for girl-centric design measures, the topic will be related to subjects which interest girls more, such as health (Sjøberg & Schreiner, 2012).

4.2. Non-functional requirements

To create the final product, differing technologies and methodologies have to be evaluated and chosen. These choices will be made with different design goals in mind in order to ensure a sustainable production cycle. The design goals are: maintainability, performance, and accessibility. As the final product has to be open to further development, maintainability is a crucial aspect in order to ensure that the code base is extendable and robust. Because the product will be deployed on computers of a high school classroom, it is important that it is optimized enough to run smoothly and be playable on a low-end (windows) computer. Finally, given the timeframe of the development phase, the chosen technologies and methodologies should be familiar, intuitive, and accessible enough to ensure development can proceed in a timely manner.

Version Control

To allow for seamless collaboration within the development team a version control system must be used. Version control allows the team to keep track of changes and to work in parallel. While there are many version control systems around, Git is the industry standard for agile development. The team and client also agreed on using Gitlab[13] as the Git-repository. Gitlab offers built-in Scrum management tools.

[13] Gitlab, a git repository manager and other features: https://about.gitlab.com/

Development Framework

The development phase of the project will be managed using the agile-scrum methodology. The phase will be split up in one-week sprints, each sprint will have its own well-defined deliverables. At the start of the sprint, the team will decide on the tasks at hand, with every sprint ending with a sprint review and retrospective. Scrum allows the team to quickly adjust to the wishes of the client, who is to be heavily involved in the development process.

The development of the game will be done through a framework similar to the 'design, play, experience' framework (figure 3). The framework allows for iterative development, where the first phase is designing, the second prototyping, and the third play-testing, which is evaluated to link back to the design phase (Winn, 2009). Instead of using the target audience in the playtest, experts will function as the evaluators. The feedback of the experts will be used to link back to the design and fine-tune it.

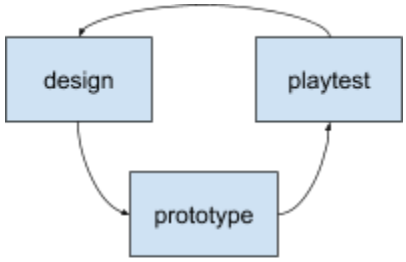


Figure 3: *Iterative design process*

Game Engine

Given the design goals, a selection of game engines were considered. An important criterion, especially given time constraints, is the accessibility of the engine and the ease with which development can proceed. The solutions considered were those that were tailored towards the development of 2D games and were not genre-specific, such as the RPGmaker[14] engine. A popular and accessible option is GameMaker Studio 2[15], which features numerous 2D development faculties, enjoys a fast set up time and allows rapid prototyping. However, the proprietary programming language it uses can make it difficult to port it to other development environments. Additionally, there is a cost attached to the license. Another option that was considered was the libGDX game development framework. This framework is open source, supports cross-platform development and is powered by Java. However, due to the fact that it is only a framework, it is very bare-bones and does not provide the tools which can be found in a fully integrated game engine that helps to facilitate the production cycle. One solution which does not contain the previous problems is Godot, a fully-featured, open-source, and free game engine with a Python-like native language.

Finally, the Unity[16] engine was considered, which also includes a fully integrated editor and is effectively free for the current use case. Furthermore, it has extensive documentation, is simple to develop with and uses a language that is accessible for the team. For these reasons, Unity was chosen.

[14] RPGmaker, an engine for the development of RPG games: https://www.rpgmakerweb.com/
[15] GameMaker Studio 2, a game engine published by YoYo games https://docs2.yoyogames.com/
[16] Unity: A cross platform game engine: https://Unity.com/

Testing

To ensure the maintainability of the game, it must be properly tested on multiple levels. For low-level testing, unit tests will be used. Unity offers the Unity Test Framework (UTF), which is based on the NTest library. UTF is the only framework which is capable of fully testing the Unity game engine, so it is basically the only choice.

High-level tests, which mostly focus on the presentation of the game, can not be automated easily. The available frameworks that can automate high-level tests for games not very accessible and maintainable. Thus, it was chosen to do high-level tests manually, by writing extensive documentation detailing every step of the testing procedure and running these tests before every merge. Additionally, the game will be regularly tested by an external group of individuals with an educational background, whose feedback will be processed to ensure that the game entertains and informs effectively.

It is also very important to frequently test the performance of the game, to make sure that the game is still runnable on low-end systems. For this, Unity offers the Performance Testing Framework, which is an extension to UTF. By automating this test, it can be ensured that every addition to the game still satisfies the performance requirements.

5. Conclusion

Based on research, a game was conceived that tries to solve the underrepresentation of women in STEM. This is done by breaking stereotypes regarding women in STEM and increasing interest in IT using an educational game. Additionally the end-user will also be informed about data centers. The game was designed in a way that is engaging for children, especially girls, between the age of 10 and 14 while at the same time trying to keep the game as gender-inclusive as possible. Different solutions and technologies were considered in order to best satisfy the requirements. The final design is a 2D puzzle game with RPG elements, which will be developed using the Unity game engine using Agile-Scrum development. Finally, during the development process, continuous evaluation and testing will be conducted in order to optimize the final product.

References

Achiam, M. & Holmegaard, H.T. (2016) Good practices on gender inclusion in stem communication. Retrieved 26th April, 2020 from http://www.expecteverything.eu/file/2016/12/Deliverable-2.2-Good-practices-on-gender-1.pdf

Andrae, A. S. G., & Edler, T. (2015). On Global Electricity Usage of Communication Technology:

Trends to 2030. *Challenges*, *6*(1), 117–157.

Apfelbaum, E. P., Phillips, K. W., & Richeson, J. A. (2014). Rethinking the Baseline in Diversity Research: Should We Be Explaining the Effects of Homogeneity? *Perspectives on Psychological Science: A Journal of the Association for Psychological Science*, *9*(3), 235–244.

Booy, C. (2012). *Trend analysis: gender in higher STEM education*. VHTO, National Expert Organisation Girls/Women and Science/Technology.

Berg, M. A., & Petersen, S. A. (2013). Exploiting Psychological Needs to Increase Motivation for Learning. Serious Games Development and Applications, 260–265. https://doi.org/10.1007/978-3-642-40790-1

Bloom, B.S. Taxonomy of Educational Objectives. Vol. 1: Cognitive Domain; Edwards Bros.: Ann Arbor, MI, USA, 1956; pp. 15–24.

Bodzin, A., & Gehringer, M. (2001). Breaking science stereotypes. Science and Children, 38(4), 36.

Connolly, T. M., Boyle, E. A., MacArthur, E., Hainey, T., & Boyle, J. M. (2012). A systematic literature review of empirical evidence on computer games and serious games. In *Computers & Education* (Vol. 59, Issue 2, pp. 661–686). https://doi.org/10.1016/j.compedu.2012.03.004

Dutch Data Center Association. (n.d.-a). Retrieved June 15, 2020, from https://www.dutchdata centers.nl/data centers/wat-is-een-datacenter/

Dutch Data Center Association. (n.d.-b). Retrieved June 15, 2020, from https://www.dutchdata centers.nl/data centers/waarom-zijn-er-datacenters/

GameTree. (2020, January 14). Industry Results: Genre and Platform Preferences (Age & Gender). Retrieved April 28, 2020, from

https://gametree.me/blog/global-gamer-insights-report/#Difference_in_Genre_Preferences_by_Gender

Hartmann, T., & Klimmt, C. (2006). Gender and Computer Games: Exploring Females' Dislikes. In *Journal of Computer-Mediated Communication* (Vol. 11, Issue 4, pp. 910–931). https://doi.org/10.1111/j.1083-6101.2006.00301.x

Kinzie, M. B., & Joseph, D. R. D. (2008). Gender differences in game activity preferences of middle school children: implications for educational game design. In *Educational Technology Research and Development* (Vol. 56, Issues 5-6, pp. 643–663). https://doi.org/10.1007/s11423-007-9076-z

Levine, L. J., & Pizarro, D. A. (2004). Emotion and Memory Research: A grumpy overview. In Social Cognition (Vol. 22, pp. 530–554). California, United States of America: University of California.

Liles, V., & hello motherhood. (2018, 5 december). The Objectives of Kid's Games & Activities. Retrieved 30th April 2020 from https://www.hellomotherhood.com/article/560965-fun-activities-for-kids-on-telling-the-truth/

Meyer, M., Cimpian, A., & Leslie, S.-J. (2015). Women are underrepresented in fields where success is believed to require brilliance. *Frontiers in Psychology*, *6*, 235.

NRO. (2016, 15 juni). Educatief rekenspel is effectief voor vmbo'ers. Retrieved 30th April 2020 from https://www.nro.nl/educatief-rekenspel-is-effectief-voor-vmboers/Olesen, J. (2019, 2 augustus). How Does Color Affect Memory? Retrieved 30th April 2020 from https://www.color-meanings.com/how-does-color-affect-memory/

Prensky, M. (2001). Fun, play and games: What makes games engaging. Digital game-based learning, 5(1), 5-31.

Risk Considerations for Data Center Fire Protection - SFPE. (n.d.). Retrieved June 15, 2020,

from https://www.sfpe.org/page/FPE_ET_Issue_76/Risk-Considerations-for-Data-Center-Fire-Protection.htm

Romero, M. (2019). Computers in Secondary Schools: Educational Games. In *Encyclopedia of Education and Information Technologies* (pp. 1–4). https://doi.org/10.1007/978-3-319-60013-0_31-1

Sammet, K. & Kekelis, L. (2016). Changing the Game for Girls in STEM. A white paper on best practices and learnings from leaders in the field. Retrieved 27th April, 2020 from www.itu.int/en/ITU-D/Digital-Inclusion/Women-and-Girls/Girls-in-ICTPortal/Documents/changing-the-game-for-girls-in-stem-white-paper.pdf

Salen, K., Tekinbaş, K. S., & Zimmerman, E. (2004). *Rules of Play: Game Design Fundamentals*. MIT Press.

Sjøberg, S., & Schreiner, C. (2012). Results and Perspectives from the Rose Project. In *Science Education Research and Practice in Europe* (pp. 203–236). SensePublishers, Rotterdam.

Steinke, J. (2016). A portrait of a woman as a scientist: breaking down barriers created by gender-role stereotypes. In *Public Understanding of Science* (Vol. 6, Issue 4, pp. 409–428). https://doi.org/10.1088/0963-6625/6/4/006

Tekofsky, S., Miller, P., Spronck, P., & Slavin, K. (2017). The Effect of Gender, Native English Speaking, and Age on Game Genre Preference and Gaming Motivations. In *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering* (pp. 178–183). https://doi.org/10.1007/978-3-319-49616-0_17

U.S. Energy Information Administration (EIA). (n.d.). *How much electricity does an American home use? - FAQ - U.S. Energy Information Administration (EIA)*. Retrieved April 28, 2020, from https://www.eia.gov/tools/faqs/faq.php?id=97&t=3

Winn, B. M. (2009). The Design, Play, and Experience Framework. Handbook of Research on Effective Electronic Gaming in Education , 1014–1015. Retrieved 30th April 2020 from http://gel.msu.edu/winn/Winn_DPE_chapter_final.pdf

Wyss, V. L., Heulskamp, D., & Siebert, C. J. (2012). Increasing middle school student interest in STEM careers with videos of scientists. International Journal of Environmental and Science Education, 7(4), 501-522.

Yee, N. (2006). Motivations for Play in Online Games. In *CyberPsychology & Behavior* (Vol. 9, Issue 6, pp. 772–775). https://doi.org/10.1089/cpb.2006.9.772