

# Educational Card Game in Virtual Reality

Using alternative future scenarios to educate players on  
different subjects

Dennis Boon

Bachelor thesis Game Development, HBO-ICT

Amsterdam University of Applied Sciences

Lectorate Play & Civic Media, HvA (Project Live Game Design)

# Educational Card Game in Virtual Reality

Using alternative future scenarios to educate players on  
different subjects

Dennis Boon

March – August 2020

Amsterdam, The Netherlands

Dennis Boon

500728794

06 41 16 47 80

[dennis.boon@hva.nl](mailto:dennis.boon@hva.nl)

Bachelor thesis Game Development, HBO-ICT

Amsterdam University of Applied Sciences

Supervisor: Anders Bouwer

Advisor: Stephan van der Feest

Lectorate Play & Civic Media, HvA (Project Live Game Design)

Weesperzijde 190

1097DZ Amsterdam, The Netherlands

# Contents

Abstract.....	1
1. Introduction .....	2
2. Context.....	3
2.1 Lectorate Play & Civic Media.....	3
2.2 Assignment Description.....	3
2.3 Problem Analysis.....	4
2.4 Research Questions.....	5
3. Methods .....	6
3.1 Literature Study.....	6
3.2 Research Through Design.....	6
3.3 Research Through Evaluation.....	7
3.4 Tools.....	7
4. Related Work.....	8
4.1 Card Games.....	8
4.1.1 Physical Card Games with Alternative Future Scenarios.....	8
4.1.1.1 The Thing From The Future.....	8
4.1.1.2 The King’s Dilemma.....	9
4.1.2 VR Card Games that use Immersion.....	10
4.1.2.1 Manastorm: Champions of G’nar.....	10
4.1.2.2 Cyber Security VR Card Game.....	11
4.1.2.3 Dragon Front.....	12
4.1.2.4 Skyworld: Kingdom Brawl.....	12
4.2 Immersion.....	13
4.2.1 Educational Value of Immersion.....	13
4.2.2 Conclusion on Immersion.....	14
5. Designing an Educational Card Game for VR.....	15
5.1 Design Goals.....	15
5.2 Design Process.....	15
5.3 Reasons for VR.....	16
5.4 Game Concept.....	17
5.5 Learning Goals.....	17
6. Creating an Educational Card Game for VR.....	18
6.1 Creation Process.....	18

6.2 Technical Analysis .....	21
6.2.1 Virtual Reality .....	21
6.2.2 Networking .....	22
6.2.3 User Interface .....	25
6.2.4 Scenario Creator.....	25
6.2.4.1 Graph View API .....	25
6.2.4.2 Scriptable Object .....	28
6.2.5 Turn System.....	29
7. Conclusion.....	31
8. Future Work.....	32
8.1 The Project .....	32
8.1.1 Voice Chat.....	32
8.1.2 Audio & Animations .....	32
8.1.3 Tool tips.....	32
8.2 The Thesis .....	32
8.2.1 Feedback .....	33
References.....	34
Appendix A – Feedback session with teachers .....	36
Appendix B – Testing results.....	40

## **Abstract**

In this thesis the use of alternative future scenarios in an educational VR experience is discussed. There are many VR experiences available and a lot of games with alternative future scenarios on the market, but not a lot of games that combine the both of them in an educational package. By having this insight, the question that came up was: How can alternative future scenarios be used in a multiplayer VR card game for educational purposes? To answer this question, research was done on the value of alternative future scenarios and immersion caused by VR on the decision making process of the player. Each of these aspects had their own research conducted and were brought together in a single VR game. In the game one up to four players have to make decisions based on a dilemma they receive, this dilemma is one out of many players will have to face in a given scenario, the scenarios are something that can be made by for example teachers using the scenario creator which the project provides. The goal of the game is to eventually help players improve their decision making ability by discussing and reflecting with the other players they are playing the game with.

# 1. Introduction

The reason for conducting this research came from the lectorate, they were curious of the added value that Virtual Reality (VR) could potentially add to educational card games. They also wanted to know if there was a possibility to make changes to a VR game based on didactic goals. Before starting on the actual research, some desk research and brainstorming was done on a couple of topics. Different card games were analyzed to see what made them interesting, different ideas on how to implement card game mechanics in a VR experience were written down and some research was done on what kind of didactic goals could be used in the game. By collecting this information, a basic structure of the game could be created.

The reason for writing this thesis is to answer the main question for this specific research. The main question in this case is: How can alternative future scenarios be used in a multiplayer VR card game for educational purposes? This question consists of multiple problems which need to be solved. These problems are alternative future scenarios, multiplayer, VR and the educational card game. The procedure for making this research a success is to create a game that contains all of the components formulated in the question while also being an educational experience for the player.

The game that came out of this thesis is about playing a decision making game in VR with one up to four people. In this turn based game, players will be confronted with a certain dilemma based on the scenario they are given. During these dilemmas players can pick up cards which stand for the available options in the dilemma, when all players have voted the game will continue to the next dilemma, unless there is a tie, then the voting process will start over. What makes the voting process interesting is that players have their own role during the game, for example, one of the players may have to focus on the economy while another player has to focus on defense, this can make for interesting discussions and decision making. This also is one of the main goals of the game, to teach players about decision making while also learning about a certain topic through the scenario. Another interesting aspect to the game is the scenario creator, this gives for example teachers the possibility to create a scenario on their own which in turn can be used during classes to teach students about certain topics.

This thesis is structured as followed:

- **Context:** This chapter discusses the context behind the thesis.
- **Methods:** This chapter discusses the methods that were used for conducting the research.
- **Related Work:** This chapter discusses other related work that was used as inspiration.
- **Designing an Educational Card Game for VR:** This chapter discusses the design process behind the project.
- **Creating an Educational Card Game for VR:** This chapter discusses the creation process behind the project.
- **Conclusion:** This chapter discusses all the results from the thesis and answers the main question.
- **Future Work:** This chapter discusses features that could possibly be added to the project in the future.

## 2. Context

This chapter discusses the context of the research for this thesis. The research focusses on solving a problem for the lectorate Play & Civic Media from the Amsterdam University of Applied Sciences. The following paragraphs will discuss the organization and have a deep dive on the actual problem that needs to be solved.

### 2.1 Lectorate Play & Civic Media

The Lectorate Play & Civic Media is one of the research groups of the CREATE-IT knowledge center of the HvA. Within the lectorate work roughly ten researchers from whom two with specific expertise in the field of card game design and live game design. The lectorate is not only active on a national level, but also international with publications on international conferences, the lectorate also takes care of the international master study Digital Design. There have been a lot of projects from the lectorate but the two most relevant projects for this thesis are ‘Didactics-Driven Development’ and ‘A Digital Prototyping Tool for Card Game Design’, within these projects the lectorate researches and develops intelligent tools for game design for educative (card) games. This is embedded in the Live Game Design (<http://livegamedesign.github.io/>) approach, in which visual and formal languages play a role in modelling, automatic generation and live changing of game aspects.

### 2.2 Assignment Description

To solve the lectorates problem the main problem has to be stripped down to multiple smaller problems first, this will show all the components that are necessary to solve the main problem. These problems are implementing alternative future scenarios, providing the ability for players to play together in multiplayer, creating a VR application and also have it be an educational card based game. These problems will have to be solved by this specific product. With this knowledge the following idea was created: In this game players will have to make choices about certain future scenarios through playing cards that have an impact on the given scenario. Based on the current given dilemma during the scenario, players will have to make decisions and considerations, among other things between personal goals and goals that benefit everyone. By doing this the players will learn about the effect of certain decisions on different alternative future scenarios. The game will be developed in VR because of the interesting interactions that become available and the ability to create a feeling of immersion. Immersion is something that could hypothetically have a huge impact on the educational value of the game. The feeling of immersion would be created through the interactivity that’s available in the game and the feeling of actually being in the world because of the ability to look around in the game world. Another thing that can get the player more immersed in a game is storytelling, when a player is really invested in a games story, he or she can become more immersed in the game, especially when this story is told in a VR environment (Elmezeny, 2018). In this game the story changes based on the players decisions, this could hypothetically make the player more invested in the scenario because of the impact their decisions make.

This idea could work out very well because of the focus on decision making and actually seeing the impact of those decisions, that combined with it being a VR experience, which provides fun interactions, makes it very compelling.

## 2.3 Problem Analysis

When the assignment was laid out, some research was done on how to actually tackle the problem. What parts would be needed to create an educative card game for the VR platform? As mentioned in the previous paragraph the project can be boiled down to four parts: alternative future scenarios, multiplayer, VR and an educational card game. In this paragraph each of these problems will be discussed.

The first problem is alternative future scenarios, these scenarios will play an important part in the creation of the game as they will provide the tools for having an educational experience, this because they will force players to make decisions during the game and by doing so improve their decision making because of the impact said decisions could have on the scenario at hand. What makes these scenarios interesting is that the project will also feature a scenario creator, this makes it possible for, for example teachers, to open the project in Unity (Unity, 2005) and to use the scenario creator to make their own scenarios which can then be built with the game and used in classes for students. In chapter 6 (Creating an Educational Card Game) the technicalities behind this idea will be discussed and explained in detail.

The second problem is multiplayer, multiplayer will be a vital part in making the game more educational because it will encourage discussion between players. For example when a controversial dilemma appears and players are on different sides it can bring very interesting discussions and decision making, this is something that will be encouraged in the game as this will only improve the overall experience when playing the game. The technical aspects behind the multiplayer will be discussed more thoroughly in chapter 6 (Creating an Educational Card Game).

The next problem is VR, the interesting thing about VR is the amount of options that become available in terms of interactivity. For the project the platform of choice is the Oculus Quest (Oculus VR, 2019), this platform provides the use of two controllers, these controllers feature tracking which means that it can track the players movement of the controller and translate that information to the game so that the player can see the hand move in the game. Another great feature that VR has is the ability to create really immersive experiences where players feel like they are actually in the game, this is called immersion. Immersion is something that's talked about a lot in the VR space. Immersion is the feeling that you're actually in the world that you're seeing in front of you, this is something that adds great educational value (Freina, 2015). Immersion can be reached by having interaction and user involvement, having these kinds of interactions with the game really makes it feel like you're part of the experience and the world. Immersion is definitely something that is focused on in this thesis as it's an important asset to increasing the educational value of the game.

The final problem is creating an educational card based game. In the previous paragraphs it was explained how the game could offer an educational experience by using alternative future scenarios, multiplayer and VR, but how can the card based aspect improve on this? There are a lot of card games out there that offer some kind of educational value, as will be explained in chapter 4 (Related Work) these card games were a great source of inspiration when it comes to thinking of ways to incorporate mechanics from card games into an educational experience.



## 2.4 Research Questions

The problem analysis led to the following main question that needs to be answered:

- How can alternative future scenarios be used in a multiplayer VR card game for educational purposes?

To answer this question, a set of sub questions must be answered beforehand. The information we get from these questions will be helpful when answering the main question. The sub questions that will be answered in this thesis are as followed:

- How can full immersion be reached in a VR card game?
- How can custom alternative future scenarios be implemented in the game?
- How can alternative future scenarios be used as an educational tool?

### 3. Methods

This chapter discusses the methods that were used to conduct this research. In this case a lot of literature study was done on the topics of immersion, alternative future scenarios and the educational value of these topics. For the VR card game aspect, a couple of similar works were analyzed to look for features that could possibly enhance the experience and ultimately see how others handled building a card game in VR. Research through design was also conducted. Designing a game that keeps the three earlier mentioned topics together and offers great educational value takes some time, but is also very insightful.

#### 3.1 Literature Study

Literature study was very important for this thesis, this is because there have already been a lot of great studies about immersion and it's impact on education. This is an important part of this thesis as the goal of the product is to offer educational value to the users playing the game.

The literature study had two main purposes.

1. Gain a general understanding of the educational value of immersion and alternative future scenarios and getting familiar with the current state of the research in these fields.
2. To use the gained knowledge as a foundation for this work, by using knowledge from research papers.

Publications were obtained from Google Scholar and Google on the topics of immersion, alternative future scenarios and their educational values. In both search engines terms related to 'immersion', 'alternative future scenarios', 'educational value of immersion', etcetera, were used to find relevant publications on these topics. Publications were also retrieved through the reference lists included in found publications. Additionally, publications were obtained via Anders Bouwer. Whenever a publication was not available without purchasing it first, the databanks from the University of Applied Sciences were used to get access to the publication. This way, most of the desired publications were obtained.

The publications obtained up till now provided insight on the current state of research done regarding immersion, alternative future scenarios and their educational values. This information was required to get a general sense of how to implement these methods in an educational VR game. The publications were mostly selected by first reading the abstracts of each of the found publications and decide after that if it was worth reading through it a little more. If that was the case and something else was found that could be interesting for the thesis, it was taken as a reference.

#### 3.2 Research Through Design

During the design process of the game, multiple meetings were held on a weekly basis. During these sessions ideas were discussed that could potentially be added to the game. By discussing these ideas a picture could be created of how the game would be if these ideas were to be implemented, this made the weekly sessions very helpful in shaping the game into the best it could possibly be. Another great thing about the weekly sessions was the multiple concepts that came out of it. These concepts were created by exchanging ideas and putting them together to create something that could be fun but also educational, this was also very helpful for creating the actual concept. In the end the final concept could be checked by looking over the sources and related work that was found, this could be used as proof that the design had potential to work out.

### 3.3 Research Through Evaluation

During the project multiple feedback sessions have been held to see what people outside of the project thought of the concept. This feedback has played a significant role in shaping the project, please see Appendix A for a written version of the feedback sessions. Feedback was not the only thing received through the sessions, additional works and sources were also provided to improve the concept and design of the game.

Next to feedback sessions, there were also plans to have playtesting sessions. Unfortunately because of the current situation in the world this became very difficult to do, luckily some people were able to test the game on location to give some feedback after having actually played to game. For the playtesting session, players started with playing through one scenario, after that players had to fill out a questionnaire to write down their experience, if they wanted, they had the option to play again and to try to get a different ending than the one they had the first time around, and after that, they could take another look at their questionnaire to see if any of their thoughts had changed after playing multiple times. The results of the playtesting sessions can be found in Appendix B

### 3.4 Tools

For the creation of the product the Unity game engine (Unity, 2005) was used. Unity is a cross-platform engine used in the creation of a lot of games. The engine provides the user with a plethora of options for creating game worlds and interesting game mechanics. Unity also supports the usage of scripting which allows users to write scripts in both C# (Wikipedia, 2020-a) and JavaScript (Wikipedia, 2020-b) which can then be implemented in the Unity game engine. Another great thing about Unity is the amount of platforms applications can be developed for, this among other things makes Unity one of the leading game engines in the world because of its relative ease of use when compared to other game engines.

Another tool that was used in the creation of the project is Microsoft Visual Studio (Microsoft, 2014). Visual Studio is a development environment for programmers which is used among other things for writing code scripts that can be used in Unity.

The next tool that was used is Git Extensions (Git Extensions, 2008). Git Extensions is a graphical user interface in which iterations of a project can be managed and saved. This allows for a iterative workflow because of the ability to back up the progress that was made on the project.

For the multiplayer of the game the Photon Engine (Exit Games, 2020) was used. The Photon Engine functions as a useful tool in implementing networking and multiplayer in any Unity engine game. This is done via a server based solution which players as a client can connect to. Using Photon is very fast, reliable and scalable in use, which makes it the number one solution for Unity developers when it comes to developing multiplayer games.

## 4. Related Work

This chapter discusses some of the other media that were used as a source of inspiration for conducting this research. Multiple sources were used, other card games, different VR games, etcetera. The sources with the biggest impact on this research will be presented in this chapter, and after that the term immersion will be discussed by using multiple sources.

### 4.1 Card Games

In this section different card games that had an impact on the design of the game will be presented. These card games will be categorized in two different categories. The first category will discuss physical card games that have alternative future scenarios as a part of their game. The second category will discuss VR card games that use immersion to improve their experience.

#### 4.1.1 Physical Card Games with Alternative Future Scenarios

##### *4.1.1.1 The Thing From The Future*

‘The Thing From The Future’ is an open-ended physical card game (Situation Lab, 2015). In this game the goal of the players is to combine four different types of cards together to create a random future, based on that random future the players have to come up with the most entertaining and thought-provoking descriptions of the near-, medium-, and long-term future they have created. What makes ‘The Thing From The Future’ so interesting is the different kinds of cards, these cards are what make the created future random and fun.

The most interesting cards are the ARC cards, these cards determine how far in the future this event takes place and if it’s either a Grow, Collapse, Discipline or Transform future. Grow is a kind of future in which everything and everyone keeps growing; population, production, consumption, etc. Collapse is a kind of future in which life as we know it has fallen, or is falling apart. Discipline is a kind of future in which things are carefully managed by concerted coordination, perhaps top-down or perhaps collaboratively. Finally, Transform is a kind of future in which a profound historical transition has occurred, whether spiritual or technological in nature. These types of cards really determine in which direction the future will be steered, which makes them really interesting for other games based around creating a random future scenario.

Other cards which are needed to create a future in the game are Terrain cards. Terrain cards describe contexts, places, and topic areas. In a completed prompt (combination of the four types of cards), the terrain card describes where, physically or conceptually, the thing from the future might be found. Two terrains appear on each card in order to provide richer possibilities for the deck. These cards determine in which context the created future will be placed.

Next on the list are the Object cards. Object cards describe the basic form of the thing from the future. The idea behind this card is that the object on the card came from the future which the players have created. This object can provide the story with more context.

The last cards are the Mood cards. Mood cards describe emotions that the thing from the future might evoke in an observer from the present. This is also an interesting part of the game, sometimes you create a future

which gives you a positive emotion but for someone else it could be a negative emotion, emotions are very personal. In this instance the game decides what emotion is connected to the created future, this ensures that the players must work with the emotion on the card and can't be influenced by their own emotions. This can be a very educational experience because you must look at something in possibly a different way.

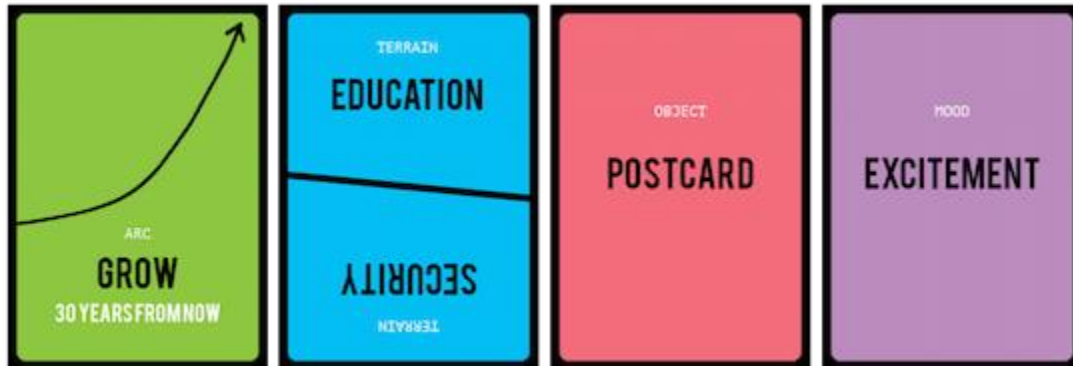


Figure 1: Example of cards used in The Thing from the Future

#### 4.1.1.2 The King's Dilemma

'The King's Dilemma' is an interactive narrative experience with legacy elements, featuring several branching storylines leading to many possible finales and an evolving deck of event cards at its core. Players represent the various houses leading the government of the Kingdom of Ankhist (Silva, 2019).

The most interesting part about this game is how it handles the different dilemma's that appear every turn. Each round a card will be drawn from the 'Dilemma deck', each card poses a problem that the Council must resolve on the King's behalf. This means that players will have to discuss what action they will take, they can discuss and bargain with each other, then finally make a choice, determining the outcome, progressing the story and possibly unlocking more events. What makes the discussion progress so interesting is the voting system. During every dilemma each player can pick three cards, each of these cards offers an option the player can pick, there is the Aye card which means you are for the action given by the dilemma, the Nay card means you're against the action, and finally there is the Pass card, this means that you don't want to choose sides in the given dilemma. One of the benefits of playing the Pass card is that you get Political Power points, these points can be used to make your vote on the next dilemma count more heavily. This can be very handy when there is a specific dilemma that really needs to go the way you need it to go. Another interesting aspect of the voting system is the ability to bribe other players into voting for what you want them to. At the start of the game every player receives twenty coins, these coins can exclusively be used for the bribe mechanic. Sometimes it can be smart to take some bribe offers because the three players with the most coins will receive extra points.

Another interesting feature in 'The King's Dilemma' is the resources system. The Council, which all players are part of, has five resources, Military Strength, Wealth, Colonization, Food and Knowledge. On top of these resources is one more resource that represents the general stability of the council. The stability of the council depends on where the five resources are at, it's very important to keep the stability of the council in mind during the game because the game will end when the stability reaches its lowest or its highest point. The reason why this mechanic is so interesting is because it makes players not only make decisions that are good for their own cause but also for the Council's cause, this can make for some very interesting discussions during the game. Every dilemma has an impact on the resources of the Council, players will want to make sure it stays

stable, but they also want the resources to be at the level they need it to be. At the start of the game when the players pick a house, the players also receive an assignment with that, that assignment is to have the resources at a certain level when the game ends, by fulfilling the assignment the players can earn some valuable points which can make the difference between winning and losing.

Lastly one more interesting thing about the game are the Prestige and Crave points. These points get handed out at the end of the game, depending on the players actions during the game the amount of Prestige and Crave points can vary. Prestige points are points that can be earned by making decisions which are beneficial for the Council. Crave points on the other hand are points that can be earned by making decisions which are only beneficial for your own goals.

What makes this game a great inspiration for an educational card game is the amount of discussion and decision making that is necessary to play the game. It can also be very interesting to see the educational value in having to sometimes choose for something that might not benefit you necessarily but in the end is the better choice.



Figure 2: Assets that are used in The King's Dilemma

## 4.1.2 VR Card Games that use Immersion

### 4.1.2.1 *Manastorm: Champions of G'nar*

For this research multiple VR card games were used as a source of inspiration. Specifically, games that had interesting uses for VR. For example, in the game 'Manastorm: Champions of G'nar' (Daggasoft, 2017) players can play cards to deal damage to certain enemies. The twist here is that the game doesn't automatically deal damage when the card is played, but the player must do a certain action to deal damage to the enemy. For example, the player plays a card that says, 'Shoot three arrows at an enemy', when this card is played, a bow

and arrow appear in front of the player, the player can then pick up the bow and arrow using the controllers and shoot the arrows at the enemy.



Figure 3: In-game screenshot of Manastorm: Champions of G'nar

#### 4.1.2.2 Cyber Security VR Card Game

Another interesting game is the cyber security VR card game by Codemodeon (Codemodeon, 2018). The goal of this game is to raise awareness on how important cyber security is in today's day and age, the game tries to do this by having one player play as the hacker and the other player play as the defender. The hacker plays cards to try and take over the defenders' hardware, in turn the defender must defend these hacks by playing defensive cards. The game also provides clear visual feedback to show what is happening when a card is played. All these assets combined make for an interesting and educational VR game, and something to take inspiration from.

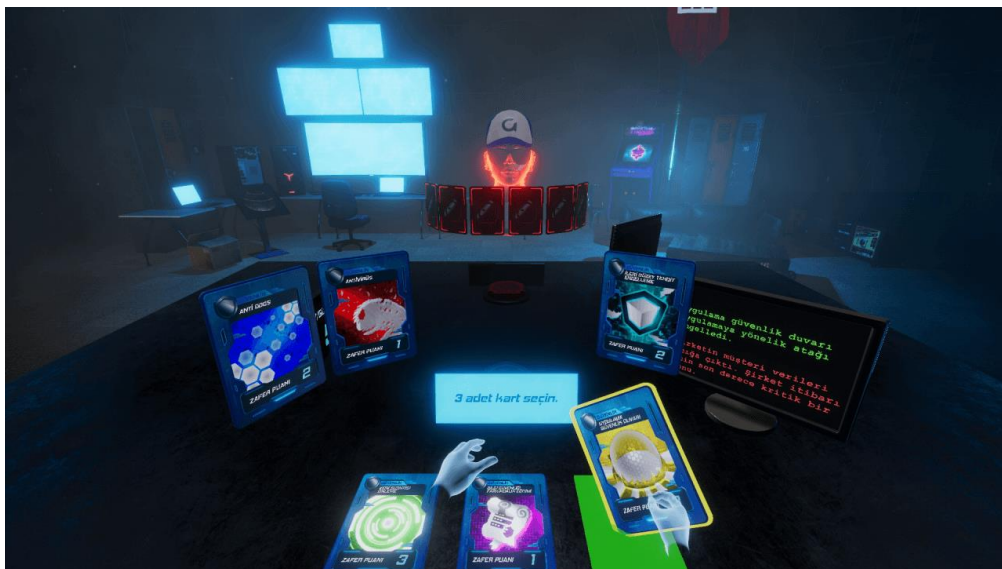


Figure 4: In-game screenshot of Codemodeon's Cyber Security VR card game



### 4.1.2.3 Dragon Front

The game 'Dragon Front' from High Voltage (High Voltage, 2016) also has an interesting feature that complements the use of VR very well. The goal of the game is to destroy the opponents fort by playing cards that spawn minions on the battlefield, these minions will in turn fight and destroy the enemies. The neat thing about this game is the ability to look at a 3D representation of the minion you want to play inside the card, you can actually look through the card to see the model of the minion you are about to play (Figure 5). This is something that could be interesting in a card game about future scenario's, what if you could look through a card and see a visual representation of the future that the specific card will create.



Figure 5: Example of a '3D Card' used in Dragon Front

### 4.1.2.4 Skyworld: Kingdom Brawl

The last game, 'Skyworld: Kingdom Brawl' by Vertigo Games (Vertigo Games, 2019), is a real-time card game in which the goal is to destroy the opponents base. The most interesting thing about this game is the amount of visual feedback the game gives when playing a card. The player can look over the entire battlefield and see minions walking around to defend their base and attack the enemies base. This is something that ties all the previously mentioned games back together. All the games have a lot of visual feedback to make the use of VR more compelling and make the game more interesting to look at in general. This is definitely a distinctive feature that makes card games work and interesting in VR.



Figure 6: In-game screenshot of Skyworld: Kingdom Brawl



## 4.2 Immersion

Immersion is the feeling of actually being there in a game. This feeling can be enhanced by for example the realism of the game world and by realistic atmospheric sounds (Brown, 2004). The feeling of immersion is often critical when enjoying a game, it can be made or destroyed by game characteristics.

Immersion is especially interesting when it comes to VR. The concept of presence is something that gets talked about a lot when it comes to this specific hardware type, this concept is defined as the 'extent to which a person's cognitive and perceptual systems are tricked into they are somewhere other than their physical location' (Patrick, 2000).

Immersion can be divided into three levels: engagement, engrossment and total immersion (Brown, 2004). The first stage is engagement. This is the lowest level of immersion, this level cannot be reached by the player when the game they are playing is not their preference, when the game controls don't give the right feedback and when the player has no willingness to concentrate on the game because of the time investment that is required. If these barriers cannot be removed, the chance the player will have a sense of engagement with the game they are playing are very slim.

From engagement the user can become further invested in the game and reach the level of engrossment. The barrier to reaching this level is game construction. This is when game features meld into one thing that affects the players emotions while playing the game. Some of these features can be visuals, interesting tasks and a plot. At this level of immersion due to the time, effort and attention put in by the player a high sense of emotional investment occurs. This investment makes players want to keep playing, the game becomes the most important part of attention.

The final level of immersion is total immersion, total immersion can also be described as presence. At this point in the scale of immersion the game is the only thing that impacts the players thoughts and feelings. The problem is that presence is only a fleeting experience. The barrier to this level is empathy and atmosphere. Empathy is the growth of attachment to for example a (main) character or team in the game. Atmosphere has the same elements as game construction. The graphics, plot and sounds combine to create atmosphere. What makes atmosphere different from game construction is relevance. The features that are presented by the game must make sense in the context of the game. This is important because of the use of attention. If a player must attend to sound, as well as sight, more effort is required by the player to pay attention. The more attention the player needs to have, the more immersed a player can feel.

Attention is very important when it comes to immersion. Games play with three forms of attention: visual, auditory and mental. The level of immersion felt by the player can be correlated to the number of attention sources needed, as well as the amount of each attention type.

### 4.2.1 Educational Value of Immersion

As mentioned in the previous paragraph, attention is very important to create an immersive experience and to create the highest level of immersion, three forms of attention are necessary, visual attention, auditory attention and mental attention. According to this paper (Cardoso-Leite, 2014), video games can have a huge improvement on the top-down attention of the person playing the game. Of course, not every game will have the same improvement, but especially action-oriented games require a lot of attention. Attention is something that really facilitates learning because when a person focusses his or her attention on something, they will try to figure out what exactly they are looking at, and in the process learn about it.

This is some very interesting information, especially when it comes to the VR platform. Visual attention is something that can be achieved rather easily using VR hardware because players who have the headset on can't really look at anything else other than the screen. Auditory and mental attention get a bit trickier, although in VR auditory attention is something that, if implemented well, can be achieved on a level that simply can't be done on regular video game hardware, this can be done with spatialized audio (Naef, 2002). Spatial audio is any audio which gives you a sense of space beyond conventional stereo, allowing the user to pinpoint where a sound is coming from, whether this is above, below or a full 360 degrees around you. Stereo allows you to hear things in the front, left and right, but you can't get a sense of surround, height or sounds from below the user. With the introduction of the 3<sup>rd</sup> dimension, the user has a sense of the exact location of sound sources from all around. The reason this is so important for VR is because when you put on the headset you are transferred into a virtual world where you can move in 360 degrees. The best way to complement this is by using spatial audio so that the user is encouraged to look around and really take the environment in.

Mental attention is defined by psychologist and philosopher William James as 'taking possession by the mind, in clear and vivid form, of one out of what may seem several simultaneously possible objects or trains of thought. ...It implies withdrawal from some things in order to deal effectively with others.' (James, 2007). This would mean that in the context of VR, users will need to be guided in a way to make sure they pay attention to the correct objects and have the correct thoughts when looking at an object in order to reach the level of mental attention that is required to achieve full immersion. One way of doing this could be to accentuate the objects you want the player to look at and think about, by doing this you make clear to the player that you want them to focus on the object that you, the creator, has marked as important. For example, by using bright colors on objects that are important, colors have been proven to evoke different emotions and behavior by various researchers (Germillion, 2019).

#### **4.2.2 Conclusion on Immersion**

With the knowledge from the previous paragraphs an idea can be created on how important immersion is for creating an educational VR experience. There are three stages when it comes to VR, but for this thesis the final stage will be the most interesting as this is the level to achieve when creating an educational experience. The goal is to create a feeling of total immersion, or presence. To create this feeling, it is necessary that the games atmosphere draws the player in and keeps them hooked on the game. Attention can be broken down in three parts, visual attention, auditory attention and mental attention. The most important part for VR will be the mental attention, how can you make sure players focus on the correct objects. In the case of this thesis an idea could be to make use of colors and make items in the scene of the game which are important have a bright color to accentuate their importance, for example, objects that players can pick-up, like the cards which are used to play the game.

## 5. Designing an Educational Card Game for VR

This chapter discusses the design process behind the educational card game that was created for this thesis to measure the educational value of software like this. This chapter will also tackle the actual design of the game and the learning goals which were implemented in the design.

### 5.1 Design Goals

Since the start of the project there have been a lot of changes to the design goals of the game. At first the idea was to have the game function as a testing environment, in this environment tests could be done to see if emotions and immersion would have an impact on player behavior and decision making. During the design process this idea slowly shifted to being more like a tool that for example teachers could use to educate their students about certain subjects through a scenario, but also to stimulate discussion between students about the given subject which in turn could lead to improved learning experiences. This shift in design goals mostly happened because of the scope at the start of the project, but this will be further explained in the next paragraph.

After the shift in design goals, the goal was now to create a game about alternative future scenarios, this game would be made in VR and have elements of card games. The main idea behind this is to give players insight in decision making, this is done by confronting the players with certain dilemmas. In these dilemmas players get a varying amount of options in the form of cards they can play. During the decision making process, players can discuss with each other about the given dilemma while each player has a different end goal they are going for, this is decided by the role they pick at the start of the game. The end goal to this is to make players conscious of how important decision making is and what the impact of said decisions can be on a certain scenario. After playing the game, the players can reflect on their decisions and the outcome they got at the end, the goal is that by having this conversation, the players will feel inclined to play again to see what would happen if they had made a different decision in a certain dilemma. By doing this players will be able to compare the routes they took to find out which decisions would be best for the role they play or maybe there is even a route that would lead to a win-win situation for everyone.

### 5.2 Design Process

As mentioned in the previous paragraph, there have been a lot of changes and shifts in the design of the game, these changes were the result of meetings which happened almost every week where design decisions and ideas were discussed. Like a lot of projects in general in the game industry, the scope of the project eventually got a little too much to handle, so the scope had to be shrunk down a little bit. At first the idea was to also tackle immersion and emotions as part of the testing goals of the game, it would have been interesting to see how immersion and emotions affected player behavior and decision making but there was just not enough time to create something that would trigger and measure this behavior. That's why it was decided to focus on the alternative future scenarios and how these scenarios can help players to improve their decision making.

During the design process of the game a lot of meetings were held to discuss gameplay mechanics that could make use of the VR hardware, but also offer some educational value to the experience. The way this was handled was by first thinking of interesting mechanics that offered some form of educational value and later think about how to implement that in a VR environment. Another thing that was necessary to figure out was how to implement card game mechanics in a VR game. To answer this question another question has to be

answered first, why create a card game in VR in the first place? This decision was mostly made because of the research that was done, as mentioned in chapter four (Related Work) the games ‘The Thing from the Future’ and ‘The King’s Dilemma’ played a big role in shaping the project to how it is now, and both of those games are card games. Another thing is that it was decided pretty early on that the game was going to be about making decisions and helping players to get self-aware of the impact of said decisions. After a lot of discussions and meetings it was decided that the best way to portray these decisions is by using cards, these cards could contain a title, a description of the decision that is made after playing the card and a picture to give some sort of visual feedback to the player of what the impact could be when playing the card. Lastly, to make sure the creation of a card game was going to be valuable in terms of educational value, sources on educational card games were gathered to confirm that card games can indeed offer a valuable educational experience (Kordaki, 2017). When all of that was figured out, work started on creating the mechanics for the game.

One of the first mechanics that was talked about was to have different future scenarios. As mentioned in chapter four (Related Work), this idea mostly came from the game ‘The Thing From The Future’. How could this mechanic be implemented in a card game? For inspiration the game ‘The King’s Dilemma’ was used, this game features a deep story line in the form of a board game. It also features a card pile of dilemma cards. Every turn one of these dilemma cards is played and players must discuss the given dilemma and eventually make a choice, are they for, against, or do they not want to vote at all. This is something that was very interesting and that was implemented in the actual concept of the game. Because of this mechanic, story lines could be created in which players make decisions that would impact the future outcome. Creating an entire story line for this particular prototype would be a lot of work, so for the design of this game it was decided that there would be a small story line to show off how the game works, but in the future there could be a system which, for example, teachers could use to add more storylines that connect to their specific areas of expertise.

Teachers also play an important role in making sure that the game has educational value. The game is mostly designed for students to be played in the classroom. Teachers could use the game as a platform to teach students about their specific subject, after the students are done playing the game the teacher could talk to them about it and ask questions about the subject and decisions that they made. By doing this it can be checked if the game has indeed helped the students in gaining knowledge about the subject the teacher teaches. Students will not only be able to gain knowledge because of playing the game,\* but also because of the discussions they will have with other students about a given dilemma.

Overall, a lot of the design choices were made based on inspiration from other works. This was mostly done because these works have already proven themselves as working products. Another great thing about getting inspiration from different works is getting new ideas based on the work that you’re researching, this can make the product more unique and maybe even better.

### **5.3 Reasons for VR**

The most asked question during the design process of the game was why VR? This question was actually quite hard to answer because a lot of the reasons that were given could also be said for a regular PC experience. In the end the following list of reasons was formulated to explain why VR as a platform was chosen for this specific project:

- VR can create the feeling that you are together with other people sitting at a table because of the avatars. It can also create a sense of communion (teambuilding, presence, etc.). This in turn can also be seen as an educational experience because you will have to talk with the other players and discuss options (Warmelink, 2017).
- VR can create a feeling of immersion by making use of the environment, lighting, audio and interactivity which take the attention of the player. These features can also be used to direct the player in what they need to focus on in specific parts of the experience (Cardoso-Leite, 2014).
- VR makes it possible to show events in the game in a realistic way because the player feels like they are actually in the game. By clearly showing the impact certain decisions have in VR, the player might feel certain emotions because it seems real (Slater, 2018).
- VR can enhance the attention users have while playing a certain game (Cho, 2002), this could make it easier to research if certain actions in the game have an influence on player behavior and decision making.

## 5.4 Game Concept

An educational card game in which players have to work together (or play alone) but also keep their own agenda in mind when making decisions based on a given dilemma. The players go through an alternative future scenario, each scenario consists of multiple dilemma's, these dilemmas can be different based on the choices the players make together. During a dilemma, players can vote for two up to four options using their voting cards. The game also features resources which connect back to every players personal agenda, resources are shared among all players and change after every dilemma. Every scenario can have multiple endings, the endings the player get depend on choices that were made and the status of each of the available resources. The goal of the game is to teach the players about decision making, the impact of said decisions and discussing certain issues that arise from decisions that were made. Another goal is to use the scenarios to teach the players about certain subjects, scenarios will also be able to be added in by for example teachers so they can create scenarios which are specifically tailored towards their respective area of expertise.

## 5.5 Learning Goals

During the design process it was important to have a list of learning goals ready to make sure that the features that were added to the game would tackle the learning goals from this list:

- Insight in how a particular goal could influence the decision making of a player.
- Insight in what the consequences can be of good/bad decision making in a certain scenario.
- Insight in egoistic choices compared to choices 'for the common good'.
- Insight in how certain situations can arise from consecutive (bad) decisions.
- Insight in to what extend the economy (resources) impacts decision making.
- Insight in short-term thinking compared to long-term thinking.
- Insight in how trust can influence player behavior.
- Insight in the importance of communication when making joint decisions.
- Insight in weighing different interests when taking decisions.
- Insight in consequences of certain decisions on (possible) outcomes of all stakeholders. (Some players win, but some lose).
- Insight in unforeseen consequences of a decision (only possible in hindsight).

## 6. Creating an Educational Card Game for VR

This chapter discusses the creation process behind the educational card game that was created for this thesis to measure the educational value of software like this. This chapter will also tackle the thought process behind some of the decisions that were made in the creation process.

### 6.1 Creation Process

Before starting the creation process of the game, a plan was made to make sure that every component got added in the correct order. It was important to first have the VR components in working order. After that when creating a multiplayer game, setting up the networking systems first is the smart thing to do because when you get started on the other components of the game, the only thing you have to do is to hook those components onto the already existing networking system. This is exactly how this project started as well. The next paragraphs will go deeper on the actual implementation of the components and features that are discussed here, this paragraph talks strictly about the creation process.

As mentioned, the first thing that got implemented were the VR components, luckily these were provided by the Oculus package, this made the actual implementation of the VR components a lot easier.

The first thing that got implemented after that was the networking systems which were provided by the Photon Engine from Exit Games. By using the Photon Engine a lobby scene could be created, in this scene players could connect to other players by creating a room or joining an already existing room. When players are connected to the room, the host of that room can start the game which will bring them to the next scene, in this scene the actual game is being played.

After having all the networking systems up and running, work started on the actual game. First a simple environment was created so that player locations could be set and some of the visuals could already be checked in the VR environment.

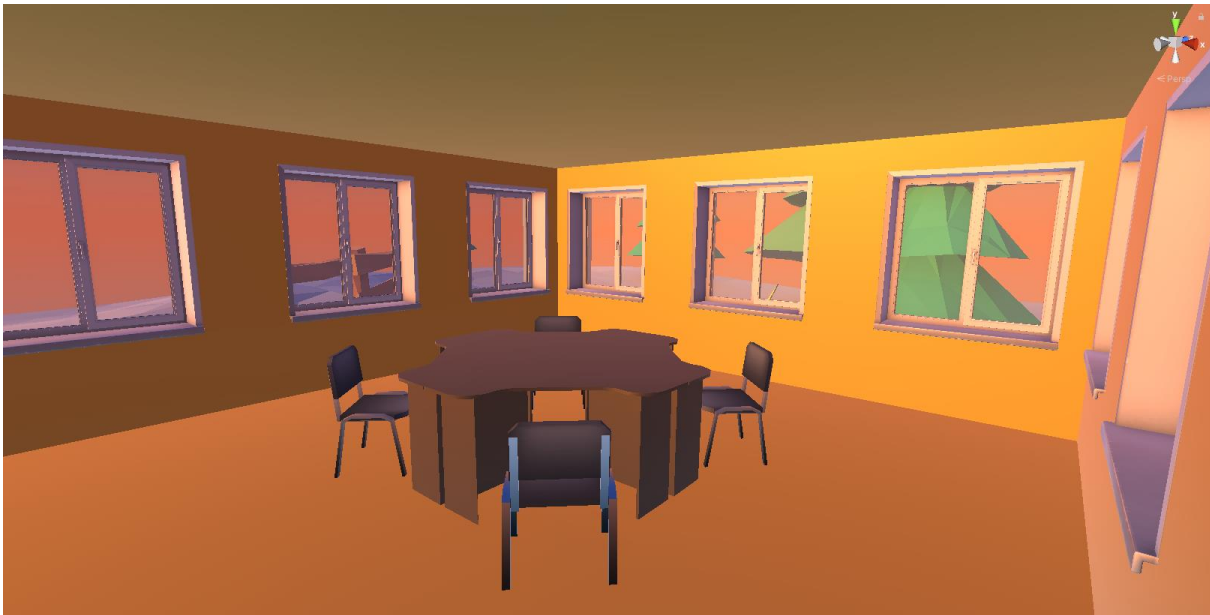


Figure 7: Environment for the VR game

When the environment was done and all player locations were set, work could start on some actual mechanics. The first one was choosing a role. For this a couple of components were necessary to make this possible, firstly the UI (User Interface), for this a canvas was created, this was also done for the main menu of the game. In this canvas there were buttons for each of the four available roles. Next is interaction with the UI element, this is something that can be quite finicky at the start because it has to translate to a VR experience. Luckily the Oculus package provides some help for this. For the interaction the Oculus laser pointer and event system were used to get the UI to work in VR. A couple more of these canvasses were used in the game but because they are mostly similar only this one will be discussed for now.



Figure 7: One the of the canvasses used in the VR game

When all of the players have picked their role, the game can finally begin. But for the game to begin a scenario is needed, this is something that will be explained in much greater detail in the next paragraphs, for this the Unity Graph View API (Application Programming Interface) (Unity, 2020) was used, this API gives users the ability to create graphs within Unity. For the project the API was used to create story trees and when a story tree was created and saved, the data of that story tree could then be used in the game. This graph editor can be accessed quite easily because the scenario creator has been implemented in the Unity editor, this makes it fairly easy to start creating a new scenario.

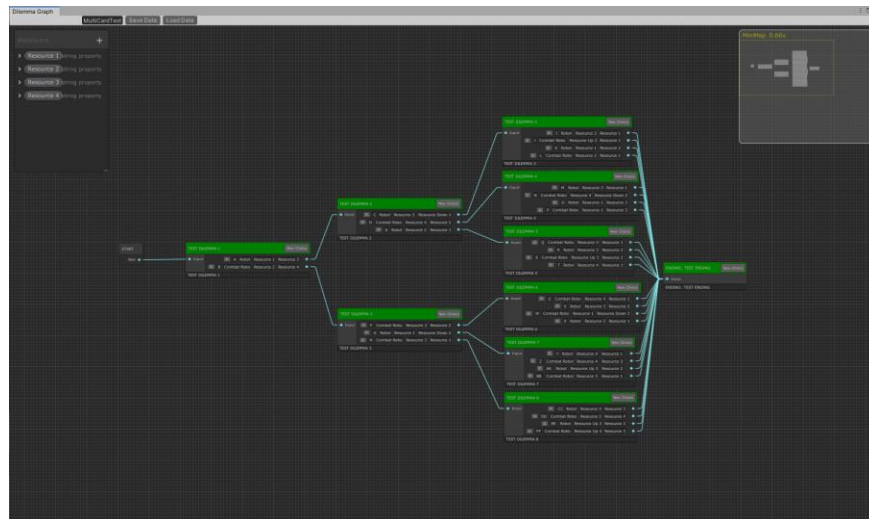


Figure 8: Example of a scenario in the scenario creator

The implementation of the scenario creator was probably the most difficult aspect during the creation of the game. This was mostly because there were multiple different components involved that all had to work together in a cohesive way, for example, one of the things that caused the biggest headaches was making the scenario turn into data that could be used by other scripts in the game. This was fairly difficult to do because there first needed to be a basic understanding of how the Graph View API actually worked before work could even begin on accessing all those features and turning them into workable data.

The next part in the puzzle was creating a board in front of the player with info about the game, this info consisted of the current dilemma, the controls of the game, the state of the resources, a general description of the scenario and a description of the role the player had picked before. This info was shown by again using a canvas. The scenario description, role description and controls text can be written in the text fields manually, the other text fields work a little differently. The current dilemma value is gathered from the scenario tree that's currently in play, this also true for the resources. Both these values will be updated at the start of a new dilemma based on the choices the players make. With all of this ready the base of the board can be created, there is however one more thing that needs to be added to the board before it can be implemented in the game, the voting board. By having this board players can put the card they want to vote on in the voting board. After all of that the board will look like this:

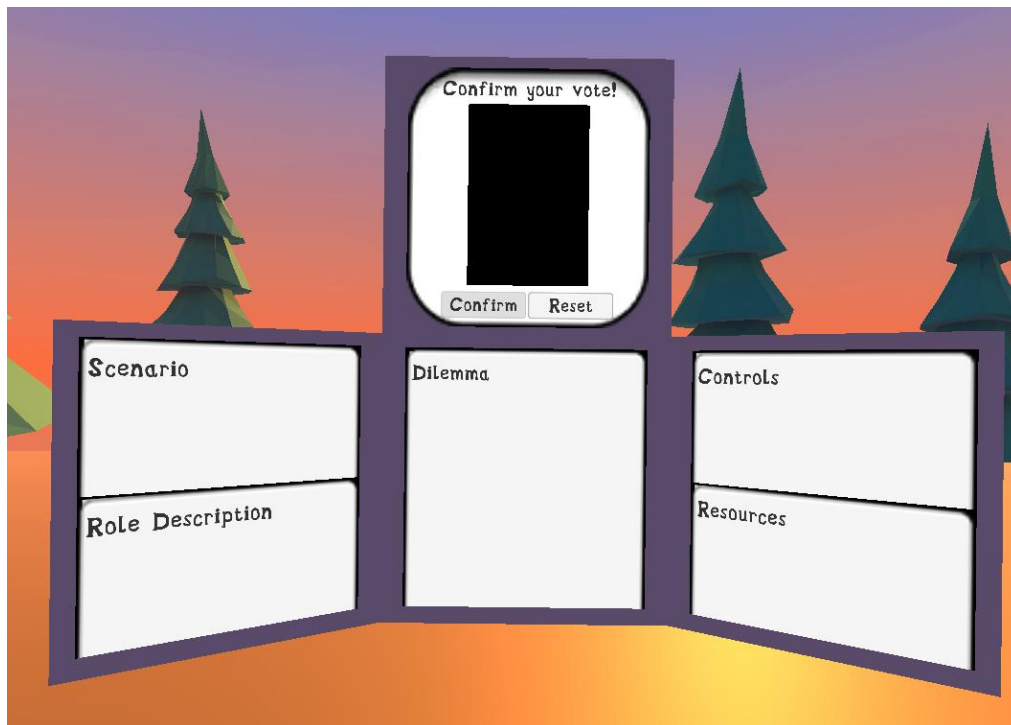


Figure 9: Information board that's used in the game

Now that the voting board is available, the other thing that's necessary are the actual cards. The cards consist of multiple components, these components are the title, a picture and the card description. These components are all part of the canvas of the card. The title of the card is static (Voting Card) but the picture and description can be different, these are obtained from the currently active scenario tree. At first the plan was to have the ability to drag pictures straight into the scenario creator but that idea became a little too difficult to implement with the amount of time left, right now the user can put in the name of the picture they want the card to have and the card will have that picture, if the picture actually exists in the correct folder that is.





Figure 10: Example of the cards used in the game

The last component that will be discussed in this paragraph is the turn system. The turn system is created by using the Finite State Machine (FSM) design pattern. By using this design pattern, programmers can create a system in which different states can be created and traversed through. This system works really well for turn based games like this, in this games case there are two states, the state in which the players can vote and the state in which the impact of the players decision gets executed. When all players have voted, the state switches to the next one and when all of the actions in that state have been executed, the current state will return to the previous one, this makes for a very fluent system and it's also fairly easy to manage because every state has their own script which makes it really clear during debugging where an error is.

## 6.2 Technical Analysis

This paragraph will go more in-depth on all the components of the game and the technique behind them, starting with the Virtual Reality implementation.

### 6.2.1 Virtual Reality

It was decided very early on that the Oculus Quest was going to be the platform of choice for this project, there were some plans for having support for the Oculus Go as well but these were scrapped because of received feedback and the fact that the platform is not really supported anymore by Oculus themselves. The main reason for going with the Quest is because of it being a stand-alone device, which means that there is no extra hardware necessary for the device to work. Another great feature of the Quest is that while it is a stand-alone device it does have two controllers which are being tracked by the headset via inside-out tracking.

The first part of setting the game up for VR is by downloading the Android configuration for Unity, this allows the user to switch the build platform in the editor to Android which is the operating system used on the Oculus Quest. After that there are a couple of settings in the editor's project settings that have to be adjusted for VR, first the actual VR support has to be enabled, after that the Vulkan rendering API has to be removed from the graphics API's because Vulkan is not supported on the Quest, so leaving Vulkan there could cause troubles in the future.

When the settings have been adjusted work can start on putting all the necessary VR components in the scene. But first, the Oculus package has to be imported from the Unity Asset Store, this package provides pretty much everything you'll need to get started on an Oculus VR Game. By using the OVRPlayerController prefab

from the Oculus package, a basic player object can be implemented into the game, this prefab provides player movement, hands and three cameras, one for the left eye, right eye and the center eye, these are all necessary to show the game in the right way when putting on the headset. While the basic player prefab from the package does come with separate game objects for the hands, the hands are actually not visible just yet. Luckily Oculus provides a solution for that as well, in the Oculus package are other prefabs that can be dragged on top of the hand objects from the player prefab to make the hands visible and react to player inputs. Because the game uses cards and VR is a platform that provides a lot of player interactivity, it would of course be fun to be able to pick up the cards in the game, once again Oculus provides a solution for this. In the Oculus package are two scripts, the OVR Grabber and OVR Grabbable scripts, these scripts are used to implement the ability to pick up objects in the game. In this case the first script (Grabber) has to be put on the hand objects, this gives the hands the ability to pick up objects that have the second script (Grabbable) applied to them by holding the correct button input.

When the VR components are ready, the best thing to do is to test if everything is in working order. This works a little differently for Android devices though. Normally in Unity development you can press the play button in the editor to test the game, this however does not work for the Oculus Quest, it is possible by using the Oculus HMD (Head Mounted Display) simulator but during the development of the project using this simulator wasn't a success. To test the game on the Quest a build has to be created, if the Quest is connected to the working PC via USB the game should be immediately installed after the building process is done. During the project there were some problems with this however, sometimes after building the new version wouldn't overwrite the older one. To fix this the ADB (Android Debug Bridge) was used, this is a helpful tool for Android development where you can use Command Prompt to issue commands that can be helpful for testing Android applications. In this case the ADB was used to first uninstall the old version of the project and afterwards install the new version, this solved the previously mentioned issue of the game not overwriting correctly. Another great feature ADB offers is the ability to see debug logs in the Command Prompt, normally when you test on the Quest you can't see these logs but by using the ADB commands this is possible to do, it offers some much needed insight in where things go wrong when testing the game.

## 6.2.2 Networking

As mentioned, Photon was used to make the multiplayer available for the game. Photon is server based which means that players will first have to make a connection with the server as a client, this is immediately done when the game starts at the main menu with the following line of code:

```
public class NetworkController : MonoBehaviourPunCallbacks
{
    0 references
    void Start()
    {
        PhotonNetwork.ConnectUsingSettings(); // Function that will try to establish a connection with the Photon server
    }

    public override void OnConnectedToMaster() // Callback function for when the first connection is established.
    {
        Debug.Log("CONNECTED TO " + PhotonNetwork.CloudRegion + " SERVER");
    }
}
```

Figure 11: Code for connecting to Photon server

This code uses a basic method provided by Photon that tries to connect to the server using the given settings in the Unity editor. These settings basically consist of a reference to the server so that the program knows to which server it needs to connect and in which region this server is based. When the connection is made a callback function will be called that shows that the connection is made and the player can move on to the lobby. When in the lobby the server will send all available rooms to the player in a list which will then be visualized as a list of buttons the player can press to join any of the rooms they want. If the player decides to make their own room they can press the create room button in the lobby, this will create a new room instance in the server which will then be available to see for other players because it is now part of the available rooms list. Creating a room is done through the following line of code:

```
public void CreateRoom()
{
    Debug.Log("CREATING ROOM..");
    RoomOptions roomOps = new RoomOptions() { IsVisible = true, IsOpen = true, MaxPlayers = (byte)4 };
    PhotonNetwork.CreateRoom("Room " + Random.Range(0,100), roomOps);
}
```

Figure 12: Code for creating a new room instance

What this code does is it will set the room options, these are static and cannot be changed because any room can have up to four people and they have to be open and visible for other players. After that the room gets created using the information given by the room options. These are all pre-written methods by the Photon developers so a lot of the actual server code is already written which is nice. When the room is created, the player who created the room becomes the host of the room, or in other words the master client of that specific room. The host of the room can at any time start the game, so if the player wants to, the game can also be played in single-player. Another thing to note is that when the other players enter an existing room their scene gets automatically synced to the one of the host, this is necessary because otherwise every player will have a different instance of the scene which can make the game become quite messy really fast.

When the players eventually join the game together there isn't really that much networking happening because the connection between the players is already there. There is only one component in the game that gets updated in real-time, which is the location of the player avatar. The players in the game can see each other's avatar moving in real-time, this is done through the following code:

```
public void OnPhotonSerializeView(PhotonStream stream, PhotonMessageInfo info)
{
    if (stream.IsWriting)
    {
        stream.SendNext(playerGlobal.position);
        stream.SendNext(playerGlobal.rotation);
        stream.SendNext(playerLocal.localPosition);
        stream.SendNext(playerLocal.localRotation);
    }
    else
    {
        this.transform.position = (Vector3)stream.ReceiveNext();
        this.transform.rotation = (Quaternion)stream.ReceiveNext();
        avatar.transform.localPosition = (Vector3)stream.ReceiveNext();
        avatar.transform.localRotation = (Quaternion)stream.ReceiveNext();
    }
}
```

Figure 13: Code for streaming player avatar location and rotation

This method will be constantly called for each of the avatars in the game. First the method will check if this object is writing data to the server or not, if that is the case this avatar instance will write the global and local location and rotation of the avatar to the server. If it's not writing anything then it will receive the position and rotation from the server and apply those to the avatar instance.

After that the only network interactions happen during specific actions through RPC (Remote Procedure Calls) calls. RPC calls are a technique that allows a program on one device to execute code on another instance of the same program on a different device without the code even needing to be there on the other device. This can be very helpful in a couple of cases, to explain this technique the RPC call that happens after the voting procedure will be explained. When a player submits their vote they will first have to press the confirm button after putting one of the cards on the voting board, this will trigger the following code:

```
[PunRPC]
0 references
public void RPC_ConfirmCard()
{
    Debug.Log("PLACED CARD: " + placedCard.name);
    if (placedCard.name == "GameCardOne")
    {
        Debug.Log("VOTE FOR OPTION ONE");
        MPGameManager.Instance.optionVotes[0]++;
    }
    else if (placedCard.name == "GameCardTwo")
    {
        Debug.Log("VOTE FOR OPTION TWO");
        MPGameManager.Instance.optionVotes[1]++;
    }
    else if (placedCard.name == "GameCardThree")
    {
        Debug.Log("VOTE FOR OPTION THREE");
        MPGameManager.Instance.optionVotes[2]++;
    }
    else if (placedCard.name == "GameCardFour")
    {
        Debug.Log("VOTE FOR OPTION FOUR");
        MPGameManager.Instance.optionVotes[3]++;
    }

    MPGameManager.Instance.playersReady++;
    MPGameManager.Instance.CheckIfVoted();
}
```

Figure 14: Code for checking if all players have voted

As you can see this method is marked as a RPC call, this means that the results of this method will be available for everyone in the game. First the card in the voting slot will be checked, the result will increment one of the four available options in the game. After that the amount of players ready will be incremented and the final check method will be called, this method checks if the amount of players that have voted is the same as the amount of players in the game. If that's the case another RPC call will be called that ends the turn for everyone in the game. This is one example of a RPC call being used in the game, these are only being used when an action from all players is necessary, for example during the voting procedure, this is because for most of the time there is no interaction between the players on a network level.

### 6.2.3 User Interface

There are a couple of things that make UI fairly interesting in VR, some things work a little differently than with a regular PC game. Luckily Oculus provides some components that help making the regular Unity UI work with VR. One of those things is a prefab which can be found in the Oculus package, this prefab consists of three items, a laser pointer, a sphere and a new event system. Especially the new event system is very important because the regular one Unity provides is not compatible with VR. But first the laser pointer and the sphere, the laser pointer can be attached to one of the two available Oculus Quest controllers in the event system, the laser pointer functions as a pointer with which players can point at UI elements in the scene. At the end of the laser pointer is where the sphere is attached, this sphere functions basically as a regular mouse cursor on a PC. Now back to the event system, this system handles the input given by the player when pointing at an eligible UI element, for example, when the player points at a UI button and presses the correct button on the Oculus controller, the event system will trigger the action that corresponds with the clicked UI button.

Another thing that works a little differently is the Unity canvas. For starters the regular rendering mode does not translate well to a VR environment, in this case the world space mode was chosen because it lets the user place the canvas anywhere in the scene without having to worry about the screen space. After that the event camera has to be changed to one of the VR cameras because otherwise the UI will not be rendered in VR. Lastly the Oculus Ray Caster script has to be added to the canvas object, this allows for the pointer to point at UI elements on the canvas, otherwise the pointer will just go through the canvas without actually pointing at anything. This approach has been used on all of the UI elements in the game, the main menu, the role selection screen, the information board, the cards, etc.

### 6.2.4 Scenario Creator

One of the main components of the project is the scenario creator, this makes it possible for the user to create a story tree using a node-based system in which nodes can be created which represent the dilemmas in the story, in these nodes up to four options can be added, these options can then be connected to the dilemma that has to come after the option has been selected. The components that make all of this possible are Unity's Graph View API and Scriptable Objects. This paragraph will first discuss the Graph View API.

#### 6.2.4.1 Graph View API

The Graph View API provides a multitude of options for creating graphs in Unity. The reason for using this API is because of its node system and visual interface. A node system is very common when it comes to dialogue trees, a node can represent a part of the dialogue and in this dialogue node are multiple options that can be chosen by the player, this is a very coherent way of managing dialogue or in this case dilemmas inside of a node tree. The visual interface that the API provides was also very appealing because of its rather simplistic look which was something that could be very helpful when providing this tool to teachers for example. The simplistic aesthetic could make it easier for them to understand how the system works. Something that also makes the visual interface very helpful is the ability to easily create connections between nodes through the options that can be added to the dilemma nodes, its basically just drag and drop.

To get access to the Graph View API every script that wants to use the API has to include the Graph View namespace, this allows for the use of the code from the API. The first step in creating a graph window is by writing the following line of code:

```

[MenuItem("Graph/Dilemma Graph")]
0 references
public static void OpenDilemmaGraphWindow()
{
    var window = GetWindow<DilemmaGraph>();
    window.titleContent = new GUIContent(text:"Dilemma Graph");
}

```

Figure 15: Code for creating a graph window

The first part of the code adds a menu item to the top of the Unity editor, this is a button and when pressed it will execute the method that's also shown in the figure. This method creates a window with nothing in it, right now at least. To fill in the window with all the necessary components for creating a scenario the following methods get executed when the window is opened:

```

private void OnEnable()
{
    ConstructGraphView();
    GenerateToolbar();
    GenerateMiniMap();
    GenerateBlackBoard();
}

```

Figure 16: Executed methods for when the graph window is opened

The first method is the Construct Graph View method, this method creates a new instance of another script called Dilemma Graph View. This script contains all the code needed for creating nodes, moving nodes, connecting nodes, interacting with nodes, etc. The code uses a lot of pre-written code from the Graph View API and uses it to visualize the scenario creation process. For example, one of the features this script adds is the ability to create new dilemma nodes, this can be done by pressing the right mouse button anywhere in the window and selecting the dilemma node, this will create a new dilemma node instance within the window as shown in the following figures:

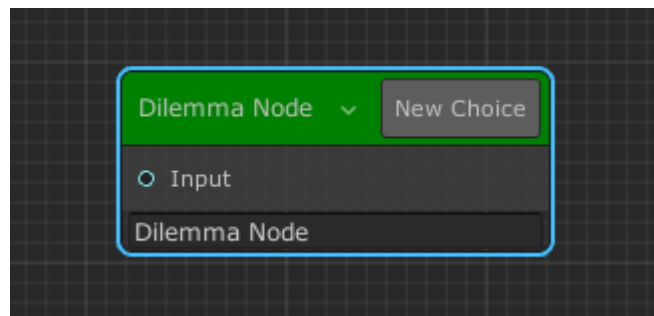
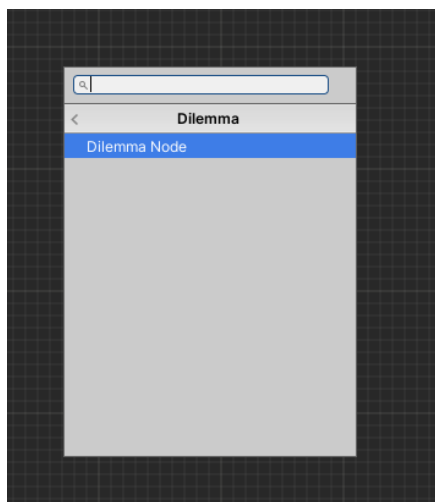


Figure 17 & 18: Example of creating a new dilemma node in the graph window

As can be seen in figure 18 there are a couple of interesting components to be found in the dilemma node. The text field at the bottom of the node is where the actual dilemma has to be filled in, above that is the node input, this is where the dilemma connects with another dilemma through the option port. The input port of a dilemma node has a capacity of multiple connections, this means that multiple option ports can connect to the same dilemma node, this can be really helpful when trying to keep the scenario tree as clean as possible, because you don't have to create a new dilemma node for every option when some of the options have the same outcome. The final part of the dilemma node is the button at the top right, when pressing this button a new option port will be created, this looks as followed:

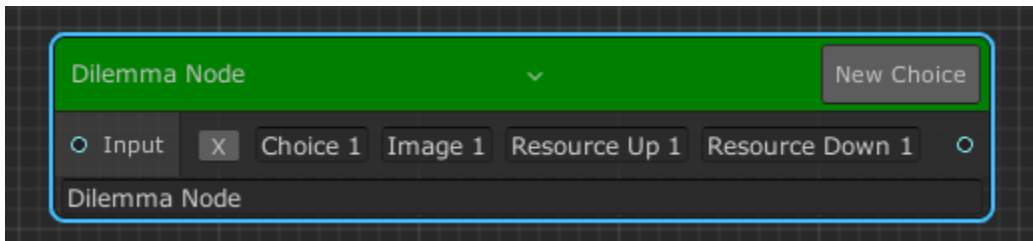


Figure 19: Example of an option port in a dilemma node

As can be seen in figure 19, the option port comes with a number of text fields that can be filled in with information. From left to right, the first text field contains the description of the option which can be seen on the card in the game. The next field contains the name of the image that needs to be shown on the card, in order for this to work the image needs to exist in the correct folder in the editor, if the image doesn't exist the card will just show a blank white image. The final two text fields contain the names of the resources that go either up or down, the names of these resources have to be available in another part of the graph window for it to work, namely the blackboard. The blackboard is another component that's added to the graph window as can be seen in figure 16, the blackboard basically functions as a list of properties, which in this case are the resources of the game. The user can use the blackboard to add resources to the game and give them starting values as can be seen here:

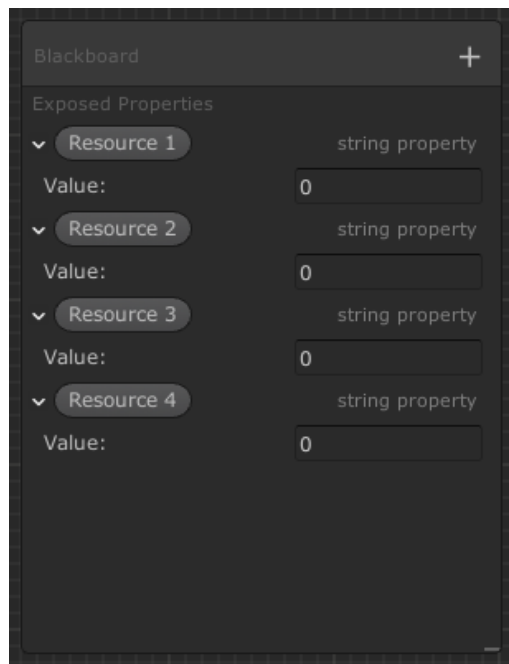


Figure 20: Blackboard with resources in the graph window

The next part of the graph window is the toolbar, the toolbar in this window is used give the scenario a name but also to search for a specific scenario when loading a scenario. There are also two buttons that can be found on the toolbar, these are the save and load data buttons, as the names already give away the save button is used for saving the scenario which turns the scenario into a scriptable object containing the data from the scenario tree. And of course, the loading button is used for loading in already existing scenarios. Last but not least is the mini-map, the mini-map can be used to navigate through the window which can be very helpful when the node tree gets a lot bigger. Another helpful tool the mini-map provides is the ability to click on nodes which causes the window to zoom in on that specific node. The mini-map looks like this:

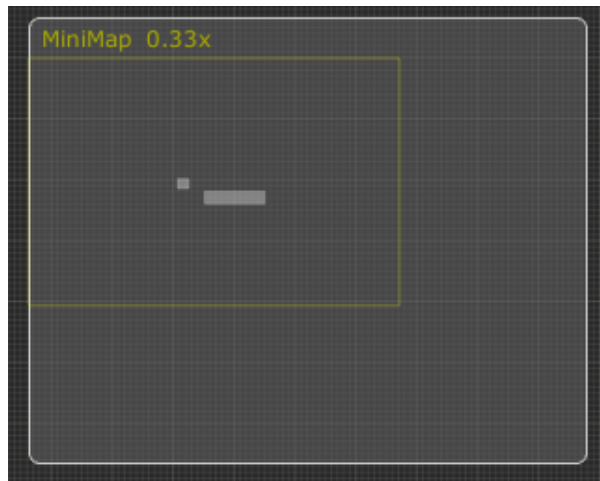


Figure 21: Mini-map which can be used in the graph window

#### 6.2.4.2 Scriptable Object

So where is all of this data being stored? For this, Unity's Scriptable Object class is used. A scriptable object is a data container that can be used for saving large amounts of data, independent of class instances. One of the main reasons for using a scriptable object is to reduce the amount of memory that's being used when the game is running. This is especially helpful when it comes to unchanging data which is exactly what the scenario data is, that data already exists and isn't changed during runtime, it's only being read during runtime.

All the data from the scenario node tree gets saved in one scriptable object called Dilemma Container, this scriptable object consists of three separate lists with data, one list for every data type, one for the dilemma nodes themselves, one for the option ports and one for the resource properties. Each of these contain their own set of variables, for example these are the variables that can be found in the option ports list, also called Node Link Data:

```
public class NodeLinkData
{
    public string BaseNodeGuid;
    public string PortName;
    public string CardImageName;
    public string ResourceUpName;
    public string ResourceDownName;
    public string TargetNodeGuid;
}
```

Figure 22: Example of one of the available lists in the dilemma container



You might be wondering what the BaseNodeGuid and TargetNodeGuid are for. When a dilemma node is created the node will receive a randomized GUID (Globally Unique Identifier), this is used in the node tree to be able to traverse to other nodes, in the case of the BaseNodeGuid this stands for the GUID of the dilemma node where the option port is in. The TargetNodeGuid is the GUID of the dilemma node where the option port goes to when that option is picked in the game. Having these two GUID's available in the option ports makes it really easy to decide the next dilemma after the voting procedure has concluded in the game.

When the user decides that the scenario they have been working on is done, they can press the save button in the graph window to save all the data from the scenario node tree and turn it into a scriptable object, which in turn can be read by other scripts that are used in the game. The saving process creates a new instance of the dilemma container, after that all the nodes will be added to the container by looking at each of the nodes in the tree individually and checking the connections between them. After that all the option port properties will be saved in a separate list and lastly the resource properties will also be saved in their own respective list. When the user wants to continue working on an existing scenario node tree they can load in the scriptable object by typing in the correct name of the scriptable object and pressing the load button in the graph window. The loading process works as followed: First the whole graph window will be cleared to make sure there are no loose nodes hanging around anywhere when the existing node tree gets loaded in, after that the node data from the scriptable object will be read and that data will be used to rebuild the entire node tree from the ground up. When all the dilemma nodes are in place, more data will be read from the scriptable object, that data is the connection data, with this data the connections between the dilemma nodes will be made. Lastly, the resource properties will be added to the blackboard. When the loading process has concluded, the user is free again to continue working on the scenario node tree.

Accessing the scriptable object is done by adding a reference to the Dilemma Container script to the script you want to access the object with. By doing this the script can use all of the data from the scriptable object and apply those to the mechanics of the game, which in this case is showing the dilemma on the board in front of the player and showing the options on the cards. This approach became very helpful in the creation of the project because if there was something missing from the games data, a new property could be added to the dilemma container which could then hold that new data which in turn could then be used in the game. The data inside of a scriptable object can also be viewed in the Unity editor, for example the data from the option port can look like this:

▼ 6f9bf9ff-d4a4-4e61-a13a-07aa409d8284	
Base Node Guid	6f9bf9ff-d4a4-4e61-a13a-07aa409d8284
Port Name	T
Card Image Name	Robot
Resource Up Name	Resource 4
Resource Down Name	Resource 1
Target Node Guid	ad8b9d26-52e3-4294-986d-314129b64a0a

Figure 23: Example of a node link data component inside of a scriptable object

## 6.2.5 Turn System

As mentioned previously the turn system was built using the FSM design pattern. The way this was implemented is by first creating an interface which holds the methods that are necessary for a state to function, these methods are the Enter, Execute and Exit method, these methods make sure that the procedures within a state run smoothly and that they can easily swap states when necessary. The next part is creating the actual State Machine, this script references the interface that was just made, keeps track of the

current state and updates the current state. This is necessary because otherwise there can be no state switching which means that the methods from the interface cannot be executed, that's why the state machine is very important. The next part is to create a script that manages the state machine and performs its actions, this script can also be used to contain data which can then be used inside of the state scripts because they have to reference this manager script before they can work.

When all of the setup has been done, the state scripts can be created, in the case of this game there are two states, the Current Turn State and the Next Turn State. The current turn state is executed when the voting process starts and the next turn state when the voting process has concluded. As an example, when the current turn state executes its Enter method, the dilemma within the game will be updated, which in turn will also update the cards and the resources, this is done as followed:

```
public class CurrentTurnState : IState
{
    // Inits TurnManager class.
    TurnManager owner;
    2 references
    public CurrentTurnState(TurnManager owner) { this.owner = owner; }

    // Method which activates when the state is entered.
    3 references
    public void Enter()
    {
        Debug.Log("ENTERING CURRENT TURN STATE");
        owner.dilemmaManager.UpdateDilemma();
    }
}
```

Figure 24: Example of one of the usages of the Finite State Machine design pattern

By using a turn system like this one the gameplay becomes very smooth and coherent which also applies to the code of the system, this is because the states are very structured which allows for easier debugging when testing the game because errors can be traced back to the right state and the right method in an instant.

## 7. Conclusion

To answer the main question of this thesis: How can alternative future scenarios be used in a multiplayer VR card game for educational purposes? The first step to answering this question was to get the design of the game down, this was done by reading papers and looking at related work. Doing all this research really benefitted the design of the game as there were many sources of inspiration that bolstered the ideas for this specific project. It was also helpful to get a better understanding of how VR works and how immersion can arise from experiencing a VR game. All of these parts together made it possible to create this project and to complete this research.

What made alternative future scenarios so interesting is the way that it can influence the players decision making which is exactly what the game was trying to do, it was meant to challenge the player in making the right decision and in the end cause them to reflect on these decisions so they can improve their decision making in the future. What made the scenarios even more interesting is the way they have been implemented in the game and how they can now be created by other users using the scenario creator in the Unity editor, this will allow the game to be valuable for a longer period of time because new content can be added infinitely.

The multiplayer is something that could've certainly be more fleshed out by adding voice chat for example, in the end the multiplayer connection itself works fine which allows multiple players to play together and when they are together locally to discuss the dilemmas at hand and maybe influence each other's decisions while doing that. Another valuable thing the multiplayer adds, at least locally, is the ability to reflect after the game is finished, this can give the players time to let everything that happened in the game sink in a little bit which maybe causes them to improve their ability to discuss or make decisions.

Lastly after going through the entire research results, the educational purpose of the project should be evident, the focus has been set on improving the players decision making process which is something that should definitely be possible when playing the game, it would have been nice however to have a way of monitoring the individual decisions each player makes so that for example a teacher can look at those results and give feedback if needed.

In the end it is important to know what kind of an impact alternative future scenarios can have on the decision making and the impact immersion can have on a players behaviour, that combined with the value of it being a VR experience should make it a great educational tool for players that want to improve their decision making.

## **8. Future Work**

### **8.1 The Project**

There are a couple of features which could be added to the game in the future if development continues after the research is concluded.

#### **8.1.1 Voice Chat**

Currently the game doesn't have voice chat implemented in the multiplayer, this was something that was fairly high on the priorities list but just wasn't possible because of the lack of time. Voice chat would be essential for playing the game over a long distance with other players because there is no other way of communicating in the game, of course for local multiplayer this doesn't really matter to much because players are probably close enough to communicate by just talking with each other. Communicating is a very important part of the experience when it comes to improving the decision making process, this is because having voice chat available would start discussions between the players about certain dilemmas which could then possibly influence the decisions of every individual player. Voice chat could also be very helpful at the end of the game to reflect with the other players about the game they just played and maybe to decide to play the game another time. This is definitely a feature that would be high on the list when development maybe continues in the future.

#### **8.1.2 Audio & Animations**

Currently the game doesn't have any audio or animations, this is mostly because of the time that was spend to get all of the systems working in perfect order, this made it so that there was pretty much no time left for polishing the player experience by adding audio and animations. This is definitely something that's high on the priority list because it is commonly known that audio and animations can greatly improve the players experience when playing a game. In the case of this game the audio would mostly be used as a form of feedback, for example when a player presses a button a sound should play, the same applies for the animations, for example when a new turn begins it would have been cool to show some pleasing visual feedback to the player to make clear that a new turn is starting. These are all things that could significantly improve the overall experience.

#### **8.1.3 Tool tips**

Currently the scenario creator doesn't have any tutorials or anything of the sort, even though the process of creating a scenario using this system is quite easy, it would still be nice to add some tool tips to each of the tools the graph window provides, this only makes the use of the system smoother which is especially true for new users of the scenario creator. This is definitely something that will be looked into if the project's development continues because it will only make the system even better.

### **8.2 The Thesis**

There are a couple of things that could have been more fleshed out in the thesis but because of the lack of time and the current situation in the world this just wasn't possible.

## 8.2.1 Feedback

What this thesis lacks for the most part is some testing of the scenario creator, this wasn't really possible because the creation of this system was completed very late in the project cycle. Having other people testing the system would have probably made the system even better because of the valuable feedback you can receive from these testing sessions. Another part of the project that could require some more testing is the game itself, this was also very hard to do because of the current situation in the world and because it is still quite difficult to find people who have a VR headset available at home and have the time to download, install and test the project on their own headset. Having these testing results would have given the whole research some more confirmation on it being a success or not.

## References

- Brown, E., & Cairns, P. (2004, April). A grounded investigation of game immersion. In *CHI'04 extended abstracts on Human factors in computing systems* (pp. 1297-1300). Retrieved May 11, 2020, from:
- Cardoso-Leite, P., & Bavelier, D. (2014). Video game play, attention, and learning: how to shape the development of attention and influence learning?. *Current opinion in neurology*, 27(2), 185-191. Retrieved May 15, 2020, from: <http://pdfs.semanticscholar.org/129c/138f47f8ee9e0a3c4ec1e695f05021abaacd.pdf>
- Cho, B. H., Ku, J., Jang, D. P., Kim, S., Lee, Y. H., Kim, I. Y., ... & Kim, S. I. (2002). The effect of virtual reality cognitive training for attention enhancement. *CyberPsychology & Behavior*, 5(2), 129-137. Retrieved July 20, from: <https://www.liebertpub.com/doi/abs/10.1089/109493102753770516>
- Codemodeon. (2018). Cybersecurity VR Card Game. *Codemodeon, Turk Telekom*. Retrieved April 15, 2020, from: <https://codemodeon.com/case-studies/cybersecurity-vr-card-game/>
- Daggasoft. (2017). Manastorm: Champions of G'nar. *Daggasoft*. Retrieved April 15, 2020, from: <https://daggasoft.com/#/games/manastorm>
- Elmezeny, A., Edenhofer, N., & Wimmer, J. (2018). Immersive storytelling in 360-degree videos: An analysis of interplay between narrative and technical immersion. Retrieved July 26, 2020, from: <https://opus.bibliothek.uni-augsburg.de/opus4/frontdoor/deliver/index/docId/53564/file/7298-30747-3-PB.pdf>
- Exit Games (2020). Photon Engine. Retrieved July 28, from: <https://www.photonengine.com/>
- Freina, L., & Ott, M. (2015, April). A literature review on immersive virtual reality in education: state of the art and perspectives. In *The International Scientific Conference eLearning and Software for Education* (Vol. 1, No. 133, pp. 10-1007). Retrieved March 31, 2020, from: <https://progesis.itd.cnr.it/download/eLSE%202015%20Freina%20Ott%20Paper.pdf>
- Germillion, A. S. (2019). Colors and emotions: how colors make you feel. *99designs*. Retrieved June 10, 2020, from: <https://en.99designs.nl/blog/tips/how-color-impacts-emotions-and-behaviors/>
- Git Extensions (2008). Git Extensions. Retrieved July 28, from: <http://gitextensions.github.io/>
- High Voltage. (2016). Dragon Front. *High Voltage*. Retrieved April 15, 2020, from: <https://www.high-voltage.com/our-games-cat/2016/#games896>
- James, W. (2007). *The principles of psychology* (Vol. 1). Cosimo, Inc.. Retrieved May 15, 2020, from: <http://psychclassics.yorku.ca/James/Principles/prin11.htm>
- Kordaki, M., & Gousiou, A. (2017). Digital card games in education: A ten year systematic review. *Computers & Education*, 109, 122-161. Retrieved April 21, 2020, from: [https://www.researchgate.net/profile/Maria\\_Kordaki/publication/314091894\\_Digital\\_card\\_games\\_in\\_education\\_A\\_ten\\_year\\_systematic\\_review/links/5a47c439aca272d2945fd007/Digital-card-games-in-education-A-ten-year-systematic-review.pdf](https://www.researchgate.net/profile/Maria_Kordaki/publication/314091894_Digital_card_games_in_education_A_ten_year_systematic_review/links/5a47c439aca272d2945fd007/Digital-card-games-in-education-A-ten-year-systematic-review.pdf)

Microsoft. (2014) Microsoft Visual Studio. Retrieved July 28, from: <https://visualstudio.microsoft.com/>

Naef, M., Staadt, O., & Gross, M. (2002, November). Spatialized audio rendering for immersive virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology* (pp. 65-72). Retrieved May 15, 2020, from: <https://dl.acm.org/doi/pdf/10.1145/585740.585752>

Oculus VR. (2019, May) Oculus Quest. Retrieved July 28, from: [https://www.oculus.com/quest/?locale=en\\_EN](https://www.oculus.com/quest/?locale=en_EN)

Patrick, E., Cosgrove, D., Slavkovic, A., Rode, J. A., Verratti, T., & Chiselko, G. (2000, April). Using a large projection screen as an alternative to head-mounted displays for virtual environments. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (pp. 478-485). Retrieved May 11, 2020, from: <https://dl.acm.org/doi/pdf/10.1145/332040.332479>

Silva, L., Hach, H., & Baroni, G. (2019). The King's Dilemma. *Horrible Guild*. Retrieved May 4, 2020, from: <https://boardgamegeek.com/boardgame/245655/kings-dilemma>

Situation Lab, Candy, S., & Watson, J. (2015). The Thing From The Future. *Situation Lab, School of Design at Carnegie Mellon University, University of Southern California School of Cinematic Arts*. Retrieved April 16, 2020, from: <http://situationlab.org/project/the-thing-from-the-future/>

Slater, M. (2018). Immersion and the illusion of presence in virtual reality. *British Journal of Psychology*, 109(3), 431-433. Retrieved July 25, 2020, from: <http://publicationslist.org/data/melslater/ref-344/bjop.12305.pdf>

Unity. (2005) Unity Game Engine. Retrieved July 28, from: <https://unity.com/>

Unity. (2020) Unity Graph View API. Retrieved June 10, from: <https://docs.unity3d.com/ScriptReference/Experimental.GraphView.GraphView.html>

Vertigo Games (2019). Skyworld: Kingdom Brawl. *Vertigo Games*. Retrieved April 15, 2020, from: <http://vertigo-games.com/portfolio/skyworld-kingdom-brawl/>

Warmelink, H., Mayer, I., Weber, J., Heijligers, B., Haggis, M., Peters, E., & Louwerse, M. (2017, October). AMELIO: Evaluating the team-building potential of a mixed reality escape room game. In *Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play* (pp. 111-123). Retrieved July 25, 2020, from: <https://dl.acm.org/doi/abs/10.1145/3130859.3131436>

Wikipedia. (2020-a, July) C Sharp (programming language). Retrieved July 28, from: [https://en.wikipedia.org/wiki/C\\_Sharp\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))

Wikipedia. (2020-b, July) JavaScript. Retrieved July 28, from: <https://en.wikipedia.org/wiki/JavaScript>

## Appendix A – Feedback session with teachers

Notes from April 24, 2020

### Feedback Sonia Palha

When is the decision making process optimal, and when is not optimal?

What insights must this game give?

Are you assuming based on theory about decision making?

What is the role of VR?

Dennis:

Right now it's mostly because of the visual aspect of VR.

Maybe focus on creating emotions.

If I have to make a decision, I wonder, what role do I have?

Play as a team where everyone in the team has different roles, these roles can complement each other, for example a leader, technician, innovator, director, economist, criticus

Social interaction with dialog would make the game more interesting.

Maybe only in specific phases.

Would players maybe have the option to make their own options for each dilemma, say them or type them in.

Open-ended or not.

Possibility to change previously made decisions.

Choose three things or think of own 4<sup>th</sup> option.

Personal characteristics.

Literature tip that could maybe be interesting:

Dillenbourg, P., Baker, M. J., Blaye, A., & O'Malley, C. (1995). The evolution of research on collaborative learning. *Learning in Humans and Machine: Towards an interdisciplinary learning science.*, 189-211.



## Feedback Charlie ter Horst

Take a look at the game 'The King's Dilemma'

For a theme you could look at the following:

Developments in Science, for example atom energy (atom bombs, nuclear energy)

Privacy and social media, for example the value of data to Facebook etc. or China's social point system.

For Civics: Different roles, for example prime minister, minister of economics, etc. Why do politicians promise more than they can live up to?

To what extend is it actually a card game?

Anders: Right now it's more like a multiple-choice voting system, wrapped in a visual wrapping of cards.

Charlie: If it's going to be an actual card game the game will quickly become more and more complex, for example balancing issues.

It would be nice if the game would offer multiple themes to choose from, teacher picks one out of five themes to play the game with.

Anders: VR as emotional machinery, for example optimistic and pessimistic.

Charlie: Background and music would be easy to use for something like this.

Do you want to conduct research with this software, for example to what extend do emotional rational decisions prevent this?

Anders: Idea for VR graphics, newspapers from the future? That is an easier way to implement visual feedback than making complicated graphics.

Charlie: What do we win?

Dennis: Players maybe have different goals.

Possible learning goals:

Short term goals are in the way of long term goals.

Things for own good versus common global issues

Insight in consequences of decisions

How decisions are being made in certain settings

Finding out what interests other people have

By guessing what everyone's agenda / goal was

Who benefits?

Anders: Do you think live dialog would be good for the game?

Charlie: Yes, live dialog would be excellent for this kind of game.

Could maybe be possible without live dialog, but then you will probably have to do more with game economies.

I would for sure keep that social aspect in VR.

Prime-Minister wants to be reelected and because of that wants to score charisma points.

How do we make it fun?

Don't go to big on the moralistic aspects.

Show consequences of decisions without being to moralistic.

Results of the game, win / lose conditions:

Country failure

Global failure

Country win

Global win

Anders:

Quality of the game depends mostly on the design, scenarios and decisions.

Can we make it less depended on that?

Or do we have to make it easier to create, test and edit scenarios?

Charlie:

With two choices per dilemma and five dilemmas per game.

Only at the end of the game.

35 options.

Maybe make a microeconomy with a couple of variables, for example as in Civilization.

Formulas of the game economy.

Effect of raising basic wage.

How would you use this in an educational settings?

If a teacher needs to use it to create scenarios then you are forced to keep things simple.

Make a decision between having an amazing game or having an easy to edit game.

For example five decision making moments in the timeline, every event has two possible outcomes, players have their own agenda, after that end scores on KPI's. Then it is relatively easy to create a new story.

Economy = 0 -> No one has money

Economy = 10 ->

Ending text as a result, depends on:

Your population was healthy for X%. Delivered something on privacy.

What is the added value of VR?

Visuals.

Working in three dimensions.

For example do something with three axis's, like Power, Resources and Ethics

Can of course also be done with three separate 2D graphs

## Appendix B – Testing results

### Testing Goals

- Check if there are any bugs in the game.
- Check if the gameplay loops are clear for the player without explanation.
- The goal of the game is clear without explanation.
- Check if the player has any feedback about the game (Things that need to be added, need to be done differently, etc.).

### Test Setup

The test was done groups of team, one person used the Oculus Quest and one person used the PC (There was unfortunately only one Oculus Quest available to perform the tests). The players played through the game once and while playing had to think out loud and discuss with each other about the dilemma that was shown. When they were done they could reflect on the game and decide to play again to get another ending if they wanted, if not they could fill in a questionnaire.

There have been five unique tests during the development of the project (Testing was difficult because of COVID and government rules).

### Feedback while playing

- First time VR users had some difficulty with getting used to the controls, maybe add some sort of tutorial? Adding controls text is not enough for newcomers
- Lack of visual and auditive feedback makes the game feel bland.
- Very rarely the cards would move a little after resetting the positions of the cards.
- Pointer of the VR controller would sometimes clip through the UI which made it impossible to press any buttons.
- The playing area feels a little empty, could use some more interesting objects.
- The text fields on the board work a little clunky when trying to scroll through them.
- The descriptions on the card are hard to read when they lay on the table, you have to pick them up if you want to read them.

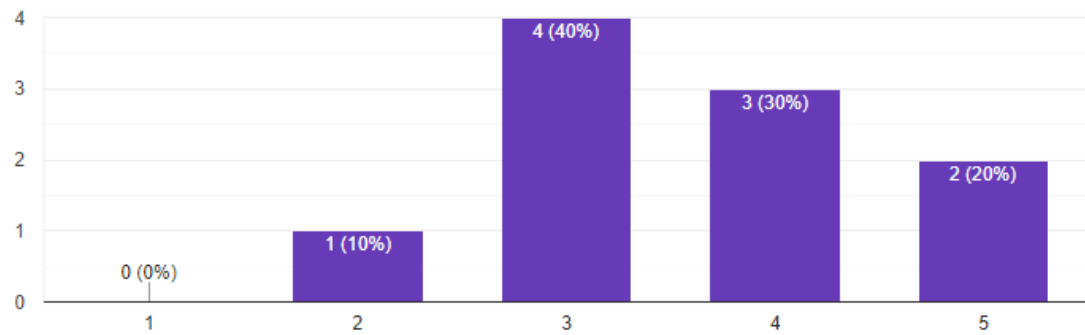
### Feedback from survey

On a scale of 1 to 5, 1 means completely disagree, 5 means completely agree.

The controls of the game were clear.



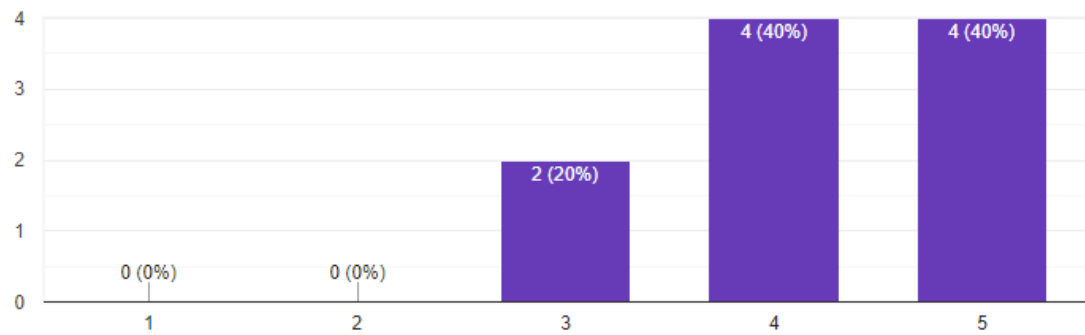
10 antwoorden



Creating a room / Joining a room was easy to do.



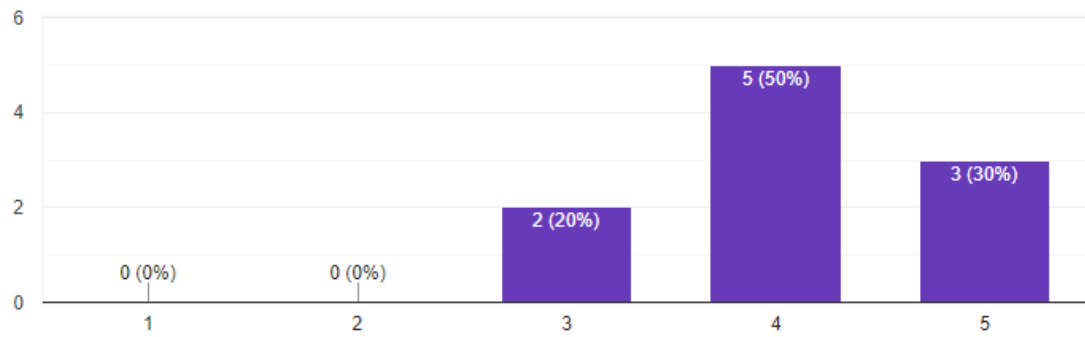
10 antwoorden



The UI (text) was easy to read.



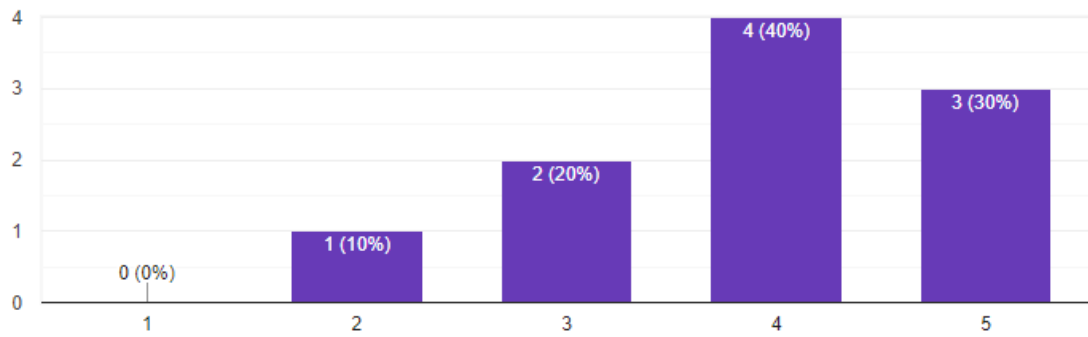
10 antwoorden



Interacting with the UI (buttons, etc.) felt intuitive.

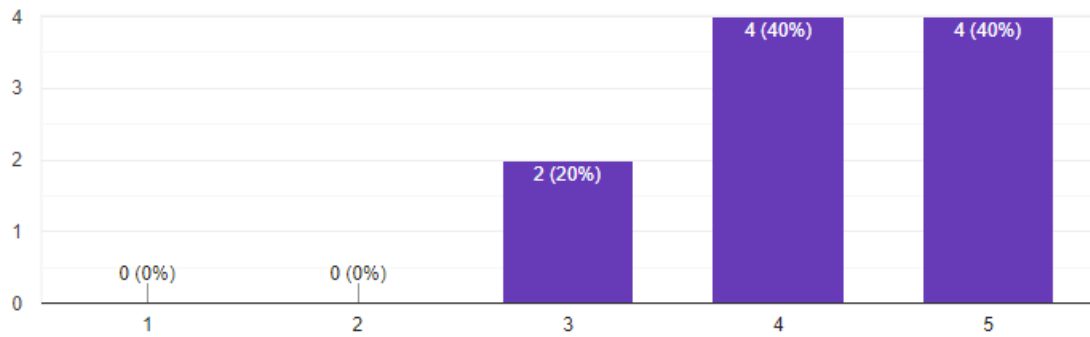


10 antwoorden



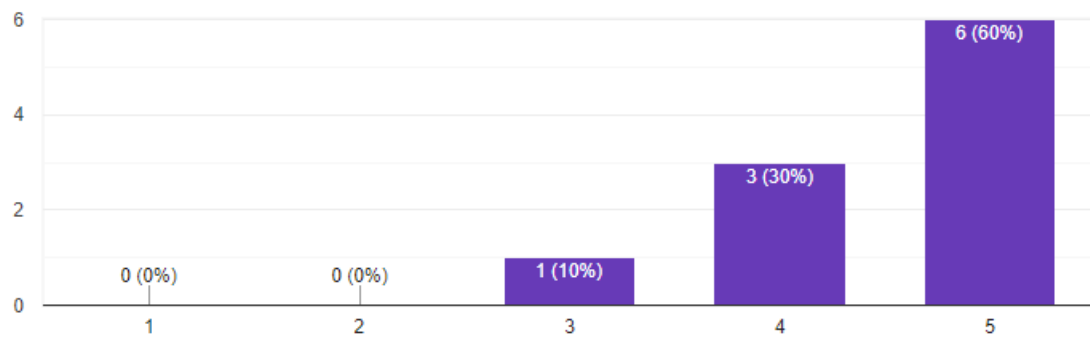
The objective of the game was immediately clear.

10 antwoorden



Picking up the cards and placing them felt intuitive.

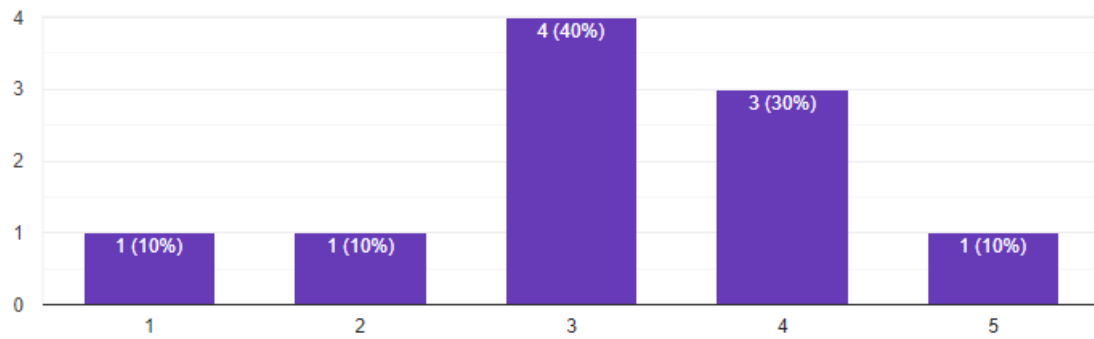
10 antwoorden



The communication with other players influenced my decision making.



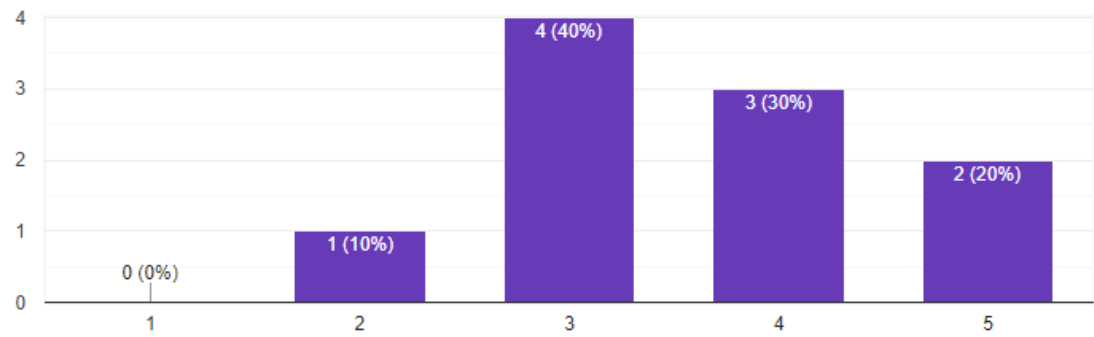
10 antwoorden



The scenario and dilemmas made the decision making challenging.



10 antwoorden

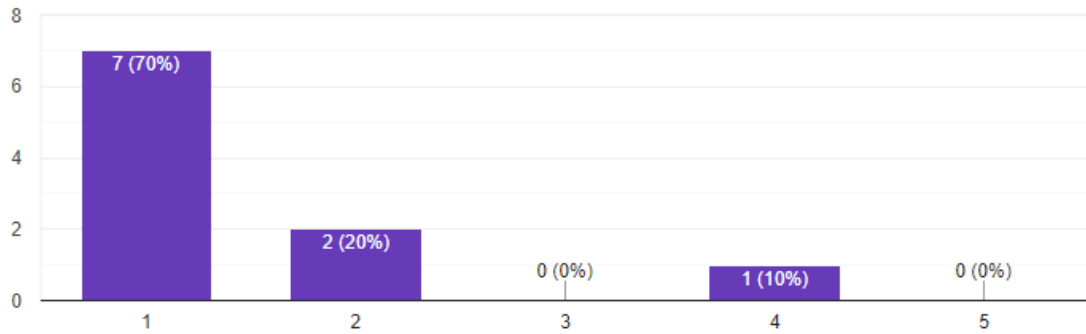




I encountered problems while playing the game (Bugs, network issues, etc.)



10 antwoorden



Extra feedback concerning the game (Additions, changes, etc.)

8 antwoorden

Some animations and sounds would have been a nice addition

I don't know if it was the network or the game but there were some very rare stutters during the game.

The cards would sometimes act a little wonky, don't know if that's my fault or the games fault though.

Lack of animations and sound lessen the experience  
The game ends a little abruptly

The buttons didn't register a couple of times which made it difficult to proceed

It was that the controls were explained to me otherwise I would have had no idea what to do. A visual tutorial would have been nice.

The room feels a little bland, it would be nice to have some more objects inside of the room, also animations and audio.

The text was a little to grainy in my opinion, I had to move my head pretty close to read it