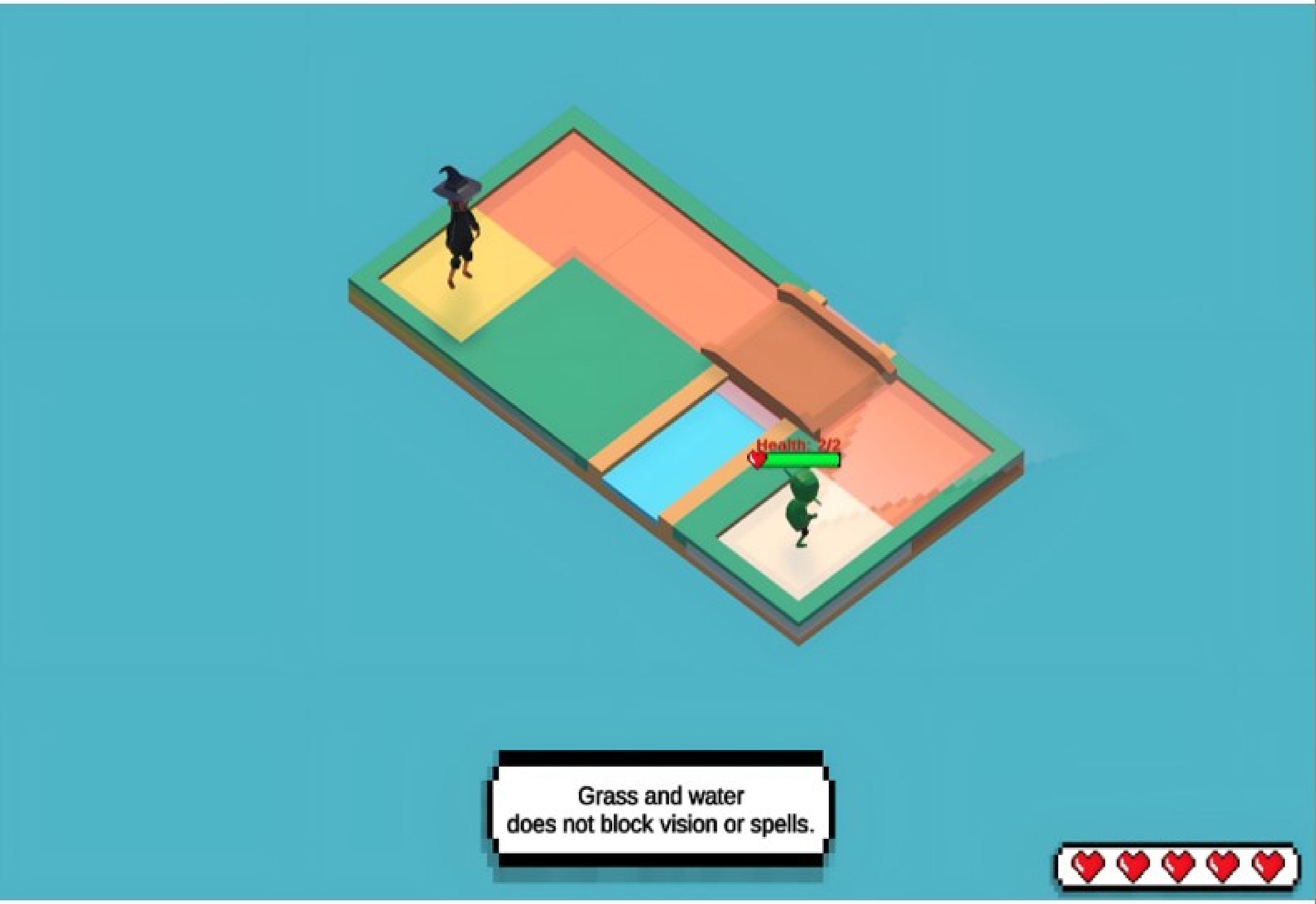
Charlotte Johansen Sebastian Whitehead

Tobias Kiholm Tonko Bossen



BCI Turn Based Strategy game

Making training engaging for CP Rehabilitation

This page is intentionally left empty

AALBORG UNIVERSITY
STUDENT REPORT

Medialogy
Aalborg University
http://www.aau.dk

Title:
BCI Turn Based Strategy Game

Theme:
Patient rehabilitation experience

Project Period:
Feb. - May 2023

Project Group:
Grp. 02

Participant(s):
Charlotte Johansen
Sebastian Whitehead
Tobias Niebuhr Kiholm
Tonko Emil Westerhof Bossen

Supervisor(s):
Bastian Ilsø Hougaard

Page Numbers: 75

Date of Completion:
May 23, 2023

Abstract:

This project describes the process and research performed to create and test two alternative interaction metaphors for motor imagery training in patients with cerebral palsy. Through research of existing solutions, it was found that existing rehabilitation games generally use an "interval" style metaphor, where there is a direct connection between training and action. Where game theory describes a possible metaphor of charging/ battery whereby performing x action one prepares for future gameplay, such as reloading a gun. Due to this a prototype was implemented to test the use of a charging metaphor against the already utilized interval metaphor to see if such a method could affect user performance engagement. Through this testing no definitive difference was found and both methods were determined to have their downsides and merits in different areas.

Preface

This project was created and developed in conjunction with the 2023, 6th semester Medialogy bachelor project at AAU. Work started February 1st and finished May 23th, 2023. The project report consists of two parts, a main report and supplementary material containing all appendices and extra documentation pertaining to the project. The supplementary material will provide extra research and documentation when such was conducted, this information may not be directly relevant to the main project but was conducted throughout the process so was therefore documented. The described implementation was created in unity 2021.3.21f1 and would likely function in newer versions. The repository for the project is publicly available through the link provided. The same is also true for the AV production required for the project hand in.

Links

- GitHub repository - https://github.com/Sebastian-Whitehead/MED6$_{projekt}$
- AV-production - https://youtu.be/WKs0KS8mpzA

Aalborg University, May 23, 2023

T K

Tobias Niebuhr Kiholm
tkihol19@student.aau.dk

Charlotte Johansen
cjohan19@student.aau.dk

Tonko Bossen
tbosse20@student.aau.dk

Sebastian W.

Sebastian Mikael Løwe Whitehead
swhite20@student.aau.dk

Contents

Chapter 1

Introduction

BCI can be used in the rehabilitation of stroke patients [18], and some researchers are now looking into using it for young CP patients as well. As it is now, the program used is not very interactive and it is bland-looking, which can make it hard for the patient to get motivated, if they can not see the reasoning behind the exercise 3.7.1. How can this be improved to further test on CP patients?
This project explores two possible solutions for this. The main focus will be on comparing the two types of pacing between the two solutions. This is done to see if there are any differences in the players' engagement compared to the game flow. The two versions created are very similar and only differs in when the motor imagery events happen, in one it is spread out with single events at a time, and in the other it is in a few groups with multiple events. Both versions were made as a game in Unity.
Lastly, an evaluation was conducted, where the results were processed, analyzed and explained to determine if there was a difference in the two versions, when looking at the players engagement.

1.1 Dictionary

CP Cerebral Palsy

BCI Brain Computer Interface

MI Motor Imagery

Engagement Refers to a player's commitment to the game activity

Chapter 2

Initial Problem Statement

The theme of the project is aiding rehabilitation of children with cerebral palsy, with a brain-computer interface (BCI). The goal is to make some of the rehabilitation process more engaging and motivating for the patient by gamifying the BCI part of the rehabilitation.
Rehabilitation is known to have a tendency to be boring and repetitive, which can lead to demotivation and resulting in a risk of lower engagement from the patient [18]. Gamification has previously been used successfully as a method to increase motivation in patients across different rehabilitation processes, while still being an effective part of the patient's treatment [22] [27].
Therefore this project aims to combine rehabilitation, gamification and BCI in a training session for children with cerebral palsy, with the goal of keeping the patients engaged in the game while doing their exercises.

The initial problem statement is as follows:
"It is possible to create an engaging BCI rehabilitation game for children with cerebral palsy. Hereby making them more motivated to do their exercises."

Chapter 3

Background research

3.1 Cerebral Palsy

Cerebral Palsy (CP) is the most common physical disability in childhood [9]. CP is a permanent physical disability, but it is not unchanging. It affects posture and/or movement along with motor functions. The cause of CP is due to damage to the developing brain, either before, during or after birth [9] [1]. The motor impairments of CP are classified in four categories, an individual can have more than one motor impairment present; (1, spastic) the most common type occurring in 80-90% of CP cases. It causes muscles to appear stiff and tight. (2, dyskinetic) causes the individual to have involuntary, uncontrolled movements. (3, Ataxic) a type of CP that causes trouble with balance and coordination, and (4, hypotonic) which results in floppy muscles, excessive flexibility and issues with their stability. [12]

3.1.1 Gross Motor function classification system

The Gross motor classification system enables categorization of gross motor skills in children with CP. There are five "levels". From level 1, where the child can walk without restrictions, to level 5, where their self mobility is severely limited even when using assistive technology. [23]

3.1.2 Manual ability classification system

The Manual ability classification system (MACS) categorizes how children with CP use their hands to handle objects in their daily lives, eg. eating, getting dressed or drawing. There are five levels, (1) Handles objects easily and with success, (2) Handle most objects with reduced quality or/and speed, (3) Handles objects with difficulty; they need help to prepare, or modified activities, (4) handles a limited selection of easily managed objects in situations that are adapted, and (5) does not handle objects and has severely limited ability to perform simple actions. [10]

3.1.3 Rehabilitation

Rehabilitation through physiotherapy and occupational therapy can play an important role for CP patients, especially for those under 18. This can come in many forms like very specific tasks for the hands, that can also help the patient function on their own, like putting on a jacket by themselves. Other than that they also do some exercises where they have to stand, walk and get up, they also train their muscles as that makes all the other stuff easier, and some of them have equine-assisted therapy. [17] We visited a school to observe a training session with a young patient and talk to the therapist about their training sessions. The therapist talked a lot about how the kids responded better if the exercises were something they could use in their life, like putting on the jacket, she also mentioned that they tried a digital training aid before, and that some of the kids were really motivated by it.

3.2 Rehabilitation motivation

Rehabilitation can be a tedious, monotonous and boring process, which can cause loss of motivation in some people undergoing rehabilitation. This can end up causing the patient to participate less in their rehabilitation. The therapist can only encourage the patient to participate actively. The duration of the participation can be an important factor for the success of the treatment. [18]
A study by Lee and Bae tested 'gamifying' [1] as a part of the participants' rehabilitation to see if it would increase motivation, while having the rehabilitation be just as rewarding. In this study, they assessed the walking and balance ability, and motivation of stroke patients after 4 weeks of rehabilitation. They had a control group and experimental group. The control group underwent walking training and traditional neurodevelopmental treatment (TNT), whereas the experimental group underwent therapy with interactive video games and TNT. [18]

3.2.1 Method

All participants received TNT sessions from the same physical therapist. The video games and walking training were 30-minute sessions, three times a week, in the 4 weeks the experiment ran, followed by TNT.
The video game training consisted of a racing game, where the patient sat on a chair with no back- or armrest. They had a steering wheel and pedals for controls. They also wore a headset to increase focus on the game. [18]

[1] turning a task into something that resembles a game.

3.2.2 Results and discussion

The outcome measure in this study were firstly motivation. Walking and balance abilities were secondary outcomes.
The primary finding of the study was that the experimental group showed a significant increase in motivation when compared to the control group. Balance were increased significantly in both groups, and the values were not significantly different. Walking speed was significantly increased only in the experimental group. [18]
In the study, the video game training along with TNT resulted in greater improvements than walking training and TNT. Both in motivation and walking speed. They argue that game-based rehabilitation programs can make the rehabilitation exercises more motivating and attractive to perform. This can be for rehabilitation where the game is the only treatment, however also as an adjunct treatment. [18]

3.3 State-of-the-art in terms of BCI games

To get a grasp of how BCI already is utilized in regard to research and video games, sources of this kind have been looked into.
The article "Games, gameplay, and BCI: The state of the art" [19] researched published papers revolving around BCI video games in terms of the games genre and, interest in the field, and use of BCI. The different genres mentioned are "Action", "Strategy", "Puzzle", "Simulation", and "Role-playing Games".
The action genres was 49% of the games with games such as "MindBalance" (a balancing game), and an endless 2D spaceship game. The BCI method with the highest accuracy were SSVEP (90%), but also motor imagery (MI) was quite accurate, in some papers. This is probably due to the simplistic design of tasks in the gameplay. 5% of the games were strategy games such as "Chess", or "Mindgame" (the player chooses the quickest path). As strategy games often are turn base, the player has a long time to think of their next move. Role-playing Games is only 4% of the games, with "Alphawow" being one of them. The game is almost as "World of Warcraft", with the player controlling the character with the mouse, and keyboard as usual. Passive BCI changes the player's form from a long range attacking elf to a short range attacking bear, when they become agitated. As the BCI is used passively, there is not required any training before playing the game. By having a normalization function, the user has the ability to change into the elf form even in a stress full context. This also help the BCI system understand the player. As the RPG genre has a lot of elements, it is very rare it utilizes BCI and instead uses hybrid BCI (mix of control and BCI). Simulation games are 30% mainly in virtual environments for training for potentially dangerous situations. Some games have the player control an avatar following predefined paths with the player, using MI to pick the path. By simulating life, one game helped train users to control a

wheelchair. Simulation games had the highest average BCI accuracy, guessed to be due to simple tasks, and the absence of "scores" which could make the players relax more. 12% are puzzle games based on already existing games such as Brain Tetris to let the player have focus on controlling BCI instead of the game's rules. "Brain Tetris" is developed to help ADHD children and in different versions such as pick the correct shape, and rotate and place a given shape.
BCI were often used as traversing in the virtual environment, but rather as to control objects. MI was found to have an average accuracy of 76%, and that fours' sessions yielded more accuracy (however, novice BCI users were often used in these studies). It is suggested, for increased accuracy, the game must have a simplistic control scheme. Additionally, the following is recommended: Keep the game slow, and accept imprecise actions as valid, avoid flashing lights for longer play sessions, be aware of BCI's absents in other game genres, explore multiple gameplay types to let the user learn BCI better, use passive BCI, the use of hybrid BCI in diverse ways, cooperative teamwork, and false positive feedback.

An article written by a team at the University of Southern California described a study conducted to investigate the effects of a BCI with VR on stroke patients with severe motor impairments. The study found that the BCI-VR system improved the patients' motor function and quality of life. The system allowed the patients to control a virtual arm with their brain signals and receive visual feedback from the virtual environment. The results of the study suggest that the BCI-VR system could be a promising tool for stroke rehabilitation. [29]

3.4 Interval and battery Pacing

A well-known type of interval commonly utilized in training is the timed interval. However, in the context of this paper, the term "interval" refers not to the passage of time, but rather to the grouping of repetitions together. The interval game mode strives for a single motor imagery event, while the battery game mode focuses on a smaller number of groups consisting of multiple motor imagery events.

3.5 Interval in rehabilitation games

While researching state-of-the-art rehabilitation games, different patterns of intervals has been found, to exist in said rehabilitation games. A SOTA table was created while researching to get an overview of patterns in rehabilitation games, it can be seen in the supplementary material. 4 different interval levels has been identified, random interval, monotone interval, long intervals, and continuous. Furthermore, a fifth interval has been discovered, however it is not discovered from the existing rehabilitation games, a reward/battery interval.

In this context and going forward, an interval is the exercise and pause structure in the game. Explanation of the different types of interval:

- **Random interval** - The intervals are random, either randomly generated each time or seemingly random made by the creators of the game, an example of a rehabilitation game with random interval is a music game [20]. Random can also be user defined, depending on the user's actions in the game.

- **Monotone interval** - When the intervals occur, they have the same training length and same pause length all the way through the training. One example of such a game used for rehabilitation is a tennis game [5], it has the same rhythm throughout the training, shooting the ball, pause while waiting for the ball to come back.

- **Longer interval** - With longer intervals, the training is done for a longer period of time with longer breaks in between. An example of this is the basketball game, Rumble With Rumble [4], where the player has to have proper posture to gain a better aim. This lets them work/train for a longer period, if they are capable of it.

- **Continuous** - In this type there is no pause during the training, the patient does continuous training with no or very little pause which cannot be justified to be called a pause. Example of this can be a balance game [14] or a posture game, endless runner style [3].

- **Charging/Battery** - The session is separated in a gaming and a training period. The reward of training is to play a game. This can be seen as a battery being charged by training, and used up by playing. An example of this is a painting game, where you have to train getting more paint, when it has been used up [7]. We could only find one game with this structure, which can either be a good research area, or a failed method in terms of training BCI.

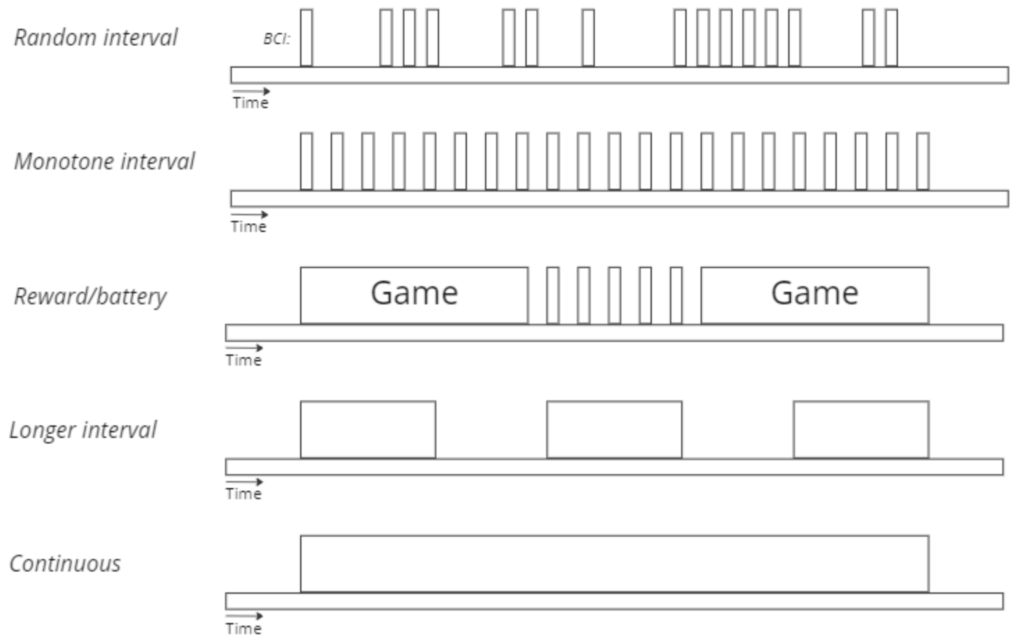


Figure 3.1: Different observed intervals

3.5.1 Task order

When handling tasks in a video game, there are multiple ways of handling the order and the degree of freedom of the tasks. They are defined as "manual", "random", "freedom", and "choice". They are illustrated in Figure 3.2. "Manual" and "random" are in terms of the program picks how the order of the tasks are proceeded. When the sequence is "manual", it is picked beforehand and by "random" it is selected by random to give diversity. The "freedom" and "choice" order, is when the tasks order is picked by the user. In "freedom" the user can always choose which task to take and go back later, but in "choice" they can only pick one task, and can not try again.

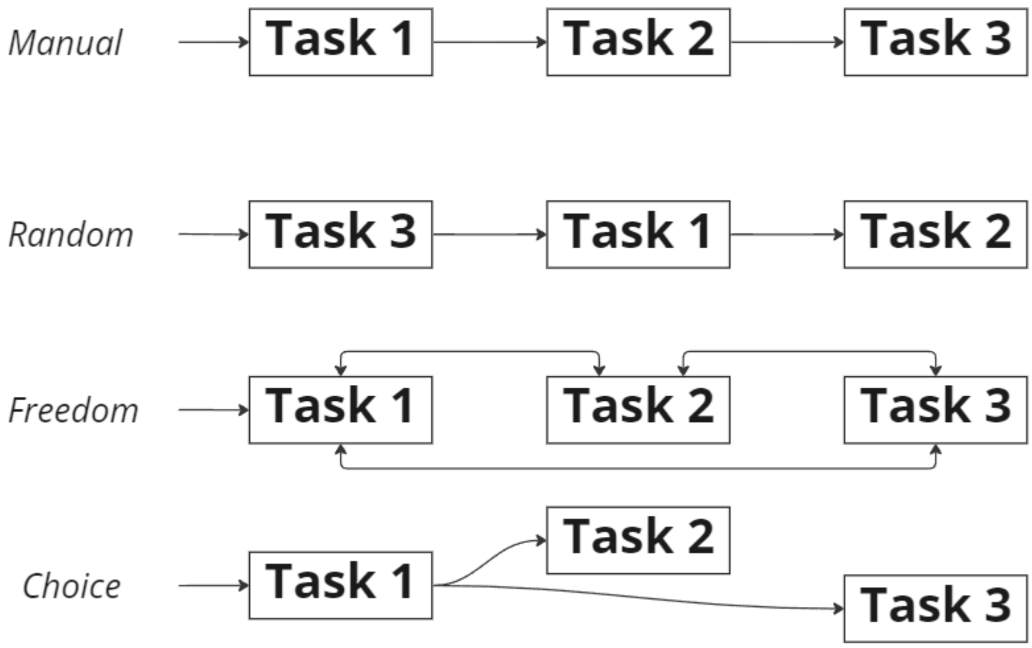


Figure 3.2: Different ways of handling task orders

Due to this difference, the idea of comparing the interval metaphor (commonly used in rehabilitations games) and the battery metaphor (derived from game theory) came forth. Attempting to see whether a distinct difference between user performance or user engagement would arise. Hereby also attempting to conclude which of the two metaphors preformed best overall and for what reasons.

3.6 Player engagement

A sense of engagement is different from player to player, but there are some elements that generally affects the engagement of the players. These elements are challenge, play, premise, character and story.

3.6.1 Challenge

Challenge is about handling the conflict of the game. This has to be balanced, or it might lead to frustration if it's too hard, or lead to no sense of achievement if it is too simple. A challenge is a task that is satisfying to complete, this makes a challenge very individualized and depends on the abilities of the player. A challenge is also said to be an activity that requires some kind of skill with clear goals and feedback. [11]

3.6.2 Play

Play is a more abstract concept. Salen and Zimmerman defined play as "free movement within a more rigid structure"[28]. Play can be many things like aggressive, serious, playful or an outlet for fantasy. Designing a game with some degree of freedom and free play is key for engaging the Player. Play can be thought of as freedom or movement within a structure, it can help us learn skills, socialize, solve problems, relax and have fun. [11]

3.6.3 Premise

Premise is what gives context to the formal elements in a game. This can help contextualize the world the players find themselves in and help them see what choices they have within the game, and by doing so it can help to get the player emotionally invested in the game. Many games have drama in them to create engagement from the player. One of the most basic drama elements is the premise. The premise establishes the action of the game with a setting or a metaphor for players to understand. [11]

3.6.4 Characters

Characters are the elements used to tell the story, but can also serve as a player's way into the world. The story is told through the characters. No matter how complex the story and character is written, there are four questions to ask when making a character. These questions are to make sure that the characters' presence is thought out. The questions are (1) What does the character want? (2) What does the character need? (3) What does the audience/player hope?(4) What does the audience/player fear? [11]

3.6.5 Story

Lastly is the story, which is used correctly can help get the player emotionally invested. In many games, the story is often a backstory, which is also the case in this game. Since it is meant to be replayed multiple times during a patient's treatment, we have limited the amount of story in the game itself.[11]

Not all games contains all these elements, but some form of them, and they all play into how much engagement the player feels towards the game. [11]

3.7 UNICS

To better understand how the CP patients gets rehabilitation now, and how a digital solution could work for them, a UNICS was performed. This was done by visiting a school and observing a rehabilitation session and later visiting a hospital, and observing multiple sessions on multiple age groups.

3.7.1 School visit

Two group members attended a physical training session with a MAC level 3 cerebral palsy patient around the age of 13 - 14 years old. The extent of the training session was around 45 minutes two times a week. The training session included a therapist talking and helping the patient with practising independence orientated tasks. An example task was the patient having to put on a button-up jacket. In this case, the therapist's assistance role was helping the patient control their involuntary movements. This help was described as above average for this participant, likely due to the presence of the three observers. This task would then likely either be repeated or another task would be undertaken. For example, practice cutting ones own nails, or moving and working safely in a kitchen. These task however were described as patient specific.
It is also important to note that the patient seen during this visit was deemed outside our chosen macs window, being a macs 3 rather than the previously specified mac 1 or 2.

Figure 3.3: Interview and observation of therapy

School Therapist Interview

Through the questions posed to the therapist present, it was very apparent that any accommodations given to a participant were very patient specific and could vary wildly. Due to this, a universally applicable training program would be impossible, but the task of deciding when a given program is applicable would and should be left down to the given therapist. It was discovered that the exercise had to be fun or make sense to the patient for them to be motivated.

Chapter 4

Final problem statement

The pacing is important in the game to keep the player engaged, by not making it too difficult or too easy. Furthermore, it is important to keep in mind that BCI interactions takes time, both the rest and the motor imagery event. Additionally, the meta goal of the video game being developed is to perform the most amount of BCI repetitions as possible, but without decreasing the player's engagement level. Since there were limited research in previous charging based rehabilitation games, it was decided to explore this area. In order to explore the area, a charging based structure and an interval structure are compared.

Based on this, the final problem statement is as follows:
"How does 'charging/battery'- and 'interval'-based pacing compare, in terms of the player's engagement, in a video game made for upper limb motor imagery training in children with cerebral palsy?"

4.1 Success criteria

To test this problem statement, the following success criteria have been listed:

1. The user should on average have a likert scale above neutral in terms of engagement and fun, with the best pacing implemented.

If the users are not motivated, they might not have any reason to use our program instead of their regular physiotherapy exercises. In a study by Reeve, J. and Lee, W.[25] they found that higher engagement from the students led to higher levels of motivation throughout the year.

2. Each BCI metaphor should present a distinct grouping/structure of BCI events

This requirement exists to ensure difference between the two presented metaphors. With this required difference, a user should be prevented from achieving identical or structurally similar groupings of BCI events. Without a distinct difference, any measurements during the evaluation process would first have to verify a significant difference before any dataset could be considered valid and useable.

3. The user should get at least 15 BCI repetitions during one game session of 20 min, excluding setup procedure.

Through an interview with a physio therapist, session length was deemed to be roughly 45 minutes. Structured with 5 - 10 minutes of session introduction. 10 - 15 minutes of system setup and calibration, leaving at minimum 20 minutes to achieve a certain number of BCI events, in this case the defined 15 repetitions. Optimally, this number will considerably higher, but a basic minimum guarantees session consistency.

Chapter 5

Design

This chapter covers the design process of creating a rehabilitation came that can adapt two different pacing versions, without changing too much of the core gameplay, in order to compare the two. The first thing to cover is the requirements of the game which will be used to have a better chance at achieving the research goal.

5.1 Game requirements

The following are the required aspects of developing the rehabilitation game we aim to create. All the requirements are developed by thinking of the abilities and limitations the target group has and are separated in the categories "movement", "gameplay" and "graphics".

5.1.1 Target group

General information about the user or the target group, and the training session.

- MACS level 1 - 2
- 16 - 18 years old
- understand task, as 50% of the target group has intellectual difficulties [9].
- Assistance to set up training session, as the user will be accompanied by a therapist.

5.1.2 Movement

The users' ability to move by themselves and placement of their body doing the session.

- No, or limited arm movements, as moving can be difficult for some users.

- Sitting down, as the main focus, should be training.
- One arm laying on the table (Mads), as it is required in terms of the training.

5.1.3 Gameplay

Overall design of the gameplay in terms of genre, game system, and game mechanics.

- Automatic or manual changing of difficulty, to engage the user.
- Slow paced and grid based, as the user can have a difficulty moving at a certain speed to follow the game.
- Endless and re-playable, to keep the player engaged after playing multiple training sessions.
- Approximately 30 minutes training session, as this is what the therapist we talked to, recommended.

5.1.4 Graphics and technology

Visual requirements of the game.

- No flashing lights and effects, as 25% of the target group has epilepsy [9].
- Simplistic, to avoid a lot of effects for the user to perceive.
- Traditional gaming experience, as VR can make users motion sick, the therapist talks with the patient doing the training, and to avoid equipping more than the BCI helmet.
- Touch screen platform, as the target group is more familiar with tablets, and can have some trouble with mouse and keyboard.

5.1.5 Interval/Battery

As this project aims to compare two different pacings, the game should be adaptable to both pacings without any big gameplay changes.

5.2 Brainstorm

To begin the design process, we brainstormed everything game related, from themes and genres to more specific game ideas. Many ideas were quickly discarded after the brainstorm.

Different genres of games we came up with included:

- Platformer
- Puzzle
- Racer
- Arcade style
- Rail shooter
- Sport
- Turn based/tactical

The game ideas we liked the most for the target group and were discussed further were:

- Magic based - Here the player would have some kind of magic ability to overcome the obstacles in the game
- Colouring - Could either be a colouring book or drawing program
- Bridge building game - make bridges that people in the game would have to traverse
- Virtual pet - Tamagochi inspired, the thought was that children especially might want to come back to something cute like a pet.
- Muscle visuals - Visualize the arm and muscles that the player needs to think about when doing the BCI exercise.
- Spider-Man, spiderweb shooter - Same idea as muscle, but with the added effect of the player being a "superhero".
- Football - Football game where the player would do BCI in certain football scenarios like penalty or corner kicks.
- Supply crate - The player would be in control of a plane and had to send supply crates down to people in need.
- Riding - Game where the player would do the BCI interaction when the horse is supposed to jump.
- Grid based - The player moves in a grid in turns overcoming the enemies.

Some of these ideas came from thinking about pre-existing games that are popular among kids around our target group, like Minecraft, Fortnite, Spider-Man, just to name a few. These examples are all unrealistic games with some fantasy or superhuman elements to them.

5.2.1 Choosing a game

After the brainstorm we discussed the different ideas, and came up with small scenarios as to what would work best with the design requirements mentioned in section 5.1 as well as the idea we saw the most potential to be fun and re-playable with the BCI.
The game idea that will be worked on and developed further from this point is a Grid based game, which is the turn based/tactical genre. The reason for this is that the game can be quite slow paced without it feeling forced because as long as it is the player's turn, the enemies will not engage or advance further. Therefore, the player has time to perform MI events without being unnecessarily stressed.

5.3 Design development of the game

During the design of the game, the target group is being kept in mind, for example in case of their limitations and their age, in hopes to pique their interest and not discourage them.
Since the problem statement mentions a comparison of two types of pacings, the game is designed with both battery and user defined interval in mind. Many of the design decisions are the same, even though the sequencing is not. If at some point something does not apply to both types, it will be mentioned.

5.3.1 Elements to engage the player

It is important to think about engaging the player, which is important for the problem statement laid out, that is all about engaging the player. When talking about engaging elements there are different tools with which we can do that; talk about challenge, play, premise, character, story and world building, how they are used in this game is explained in this section. [11] What each element does is explained further in section 3.6.

Challenge

For our game, the base challenge is to get past the enemies that are in the player's way, by incapacitating them. There are multiple ways to do this, depending on the play style, you could for example try to engage all enemies at once, or try to take them on one by one.

Play

The structure in this game is turn based, and the play is rule based, you can only do the allowed actions inside a certain area. However, the player is free to explore however they want, inside the rules applied in the game.

Premise

In this game, the premise is: The game is set in a fantasy world with goblins and mages. They are clearly enemies, as our witch has to defeat them all before they defeat her.

Character

In our case, the answers to these questions based on the main character are: (1) The character wants to defeat the enemies before they defeat her. (2) The character needs to be safe, to do that she needs to incapacitate her enemies before they get to her. (3) The audience hopes that she overcomes her obstacles and escapes without any harm and too much hardship, so that she can live. (4) The audience fears that she is incapacitated and they lose the game.

Story

The backstory is that the protagonist (playable character) has been using magic carelessly and has been discovered helping someone with her magic. She now has to run away to a new place in order to live in peace again. The story in the game itself mostly focuses on that she is a good person, and do not wish to harm the goblins pursuing her, however when they spot her, they immediately attack her. This will not communicate very clearly in the game as it is now. This is because it is not the focus of the research question, and there is a limited time-frame for the project, but it is an idea to implement if it is to be used in a real rehabilitation context.

5.3.2 Formal Elements

Formal elements are what makes the structure of the game. When talking about formal elements we mean, the players, objectives, procedures, rules, resources, conflict, boundaries, and outcome. All these are the essence of a game. [11]

Players

When playing we can do things we could never do in real life, in this game it would be magic. There is only one player in this game. The player takes on the role of the witch that has to defeat the enemies, and hopefully they will develop some kind of attachment to her and her story while they play.

Objectives

Objectives are what the player has to do in the game in order to complete it or progress. The best objectives are challenging but achievable, which makes it almost impossible to create a game for everyone. [11]
In this game the main objective is an "escape" objective, the player has to escape from the people pursuing them and get to safety. There is also a sub objective, which is finding a solution to the level, the player has to think a little tactical in order to complete the levels without much difficulty. An example could be avoiding getting detected by all enemies at once, such that the fight is a lot harder.

Procedure

The procedure is basically asking who does what, where, when and how?
The player can traverse the map, confront enemies, and exit once all enemies have been incapacitated. When confronting enemies, the player can attack and move, the enemies can do the same each round. In the battery version of the game, the player can also do the "charge" action and fill their battery. In the interval version the MI interaction will determine if they hit the enemy or not, while in the battery version the charge action is the MI interaction, and it will determine if they get more mana or not.

Rules

The rules define the game and allowable actions by the player and game environment. Some of the rules differ a little from the battery to the interval to facilitate the exercise. These rules are the core of the game, and no specific numbers or values has been determined yet, only rough ideas, and they will be play tested, first with paper prototypes, then whatever works will be developed for a digital prototype.

- Player turn - The game is turn based. Each turn, the player gets 2 actions, they can walk and attack. In the battery version, they can choose to walk and attack or spend their turn charging their energy.
- In movement - The playable character can move a defined number of squares each round. The exact number will be determined during play testing of paper prototype and digital prototype.
- Attack - The attack phase is slightly different depending on the type of pacing.

  - Battery - The player picks an enemy in their line of sight to use to attack. The attack cost is some of their charged energy.
  - Interval - The player picks an enemy, again in their line of sight. Here they will have to do a BCI exercise in order to see if they hit the enemy or not. A successful BCI attempt will result in a hit.

- To lose hit points (HP) - The player loses 1 HP for each time the enemy hits them.

- Charge and energy (only in battery version) - During their turn the player can choose to use their action to charge their energy in order to prepare for the battle ahead. The charge will be the BCI attempt in the battery version. For each time they do a successful BCI attempt, they will gain some energy, until their energy meter is full. This energy is used to cast magic.

- Completion of level - To complete the level, the player has to incapacitate all enemies in the level, this is to make sure the player does a sufficient amount of BCI exercises.

- Enemy Rules - The enemy has different rules than the player, which will be described here.
  - Movement outside battle - The enemies move in a simple pattern when not engaging in battle with the player.
  - Movement in battle - When the enemy either spots or is being attacked by the player, they will approach the player in the shortest route in order to get to the player.
  - Attack - The enemy can only hit the player from a melee range, meaning they have to stand at a square next to the player in order to land their hit.

Resources

Resources are assets the player can use to accomplish their goals. By definition, the resources must have utility and scarcity in the game system. [11]
In this game, the player has health, which when drops to zero causes the player to lose the level. In the battery version, they also have the rechargeable energy. Other than that, there are not any more resources in the core game to keep track of. It has been discussed whether or not to add a point system, so the player can compete with themselves for their next session. The point system could be based on time, turns, or successful BCI attempts to mention a few ideas. However, we need more feedback from therapist and the target group to know if this would be encouraging or the opposite.

Conflict

The conflict emerges when a player tries to accomplish the goal set by the game within its rules. The conflict is a mix of the rules, procedures and situations in the game that prevents the player from completing objectives and goal directly. [11] In this game the conflict is the enemies the player has to defeat, without losing all their HP, in order to exit the level, thus completing it.

Outcome

The outcome of the game must have uncertainty in it, in order to hold the player's attention. There are many ways to structure the final outcome of the game. In a game like the one we are designing here, the outcome will be either win or lose for the player. Either a win or loss will also result in different outcomes for the playable character, is she captured or does she get to live a peaceful life when she escapes?

5.3.3 Interaction with system

When discussing system interaction, one is referring to the information provided to a player and how it is structured, what control the player is afforded over the system as well as what feedback is provided based on the two aforementioned parts.

Information Structure

The player will always be provided with their current resource levels, this included their current energy (mana) level, their remaining hearts as well as move able distance. As well as when applicable MI and rest prompts (shown while regenerating mana), enemy field of view while navigating the level together with enemy health while in combat. There is also a level of information that could be interpreted from observation, this includes enemy patrol routes.
This structure of information is intended to provide the player with their own personal resources to manage, as well as relevant state information while regaining said resources. Information pertaining to combat or avoidance there of is provided prior to and during combat in order to reduce visual noise and streamline the flow of information to the player

Controls

Out of combat
During the player's turn, they are afforded a series of information depending on the game state. When not in combat, the player is afforded movement up to a given

distance together with the option to attack a non-altered enemy, or regenerate their mana. All these controls should be facilitated through touch screen input and be unrestricted when not within combat.

During combat
During combat, the player is given the options to move, attack or regenerate energy. Hereby tasking allowing for resource management in terms of health vs available mana compared to the enemies own health.

Feedback

Player feedback within the game is provided through persistent interface elements describing the player's own resource (health and energy.) Appearing UI elements showing an enemies' health, as well as prompts when regenerating mana. There is also potential for toggle able walking range indication for enemies when tapping on a given enemy, allowing for potential planning of future turns.
Beyond this, shader like elements showing the enemies field of view give a better overview of where an enemy can see, i.e. potential areas to avoid. Together, with combat and movement feedback being provided through animations illustrating the consequences of a given chosen action or counter action. When selecting a move, a highlight able area showing where the player can move as well as making the potential touch screen also provide feedback as to what is possible given the game state.

5.4 Gameplay

The gameplay consists of different actions each turn on a grid. The player can move around the map, recharge mana, and attack enemies. The following section describes the different aspects of the gameplay in regard to such actions, interaction, view, BCI interaction, and "hidden" elements that change the gameplay, but are not visual for the user.

5.4.1 Game view

When designing the video game, a lot of different types of possible views were discussed, to get the most appealing and appropriate one. As the game would be a tactical turn based strategy game, 1st and 3rd person, would not be very appealing. The main two types of views focused on were 2D side-view and 3D top-down-side view. 2D would make it easier to implement in regard to maps, and graphics as it is only 2 dimensions. It would also make it easier to control the player's possibilities, but can also limit the player's feel of freedom. With 3D, the maps would be more

diverse, and could give the player more possibilities. It could add more difficulty for the player to understand their relation with the enemies, compared to 2D, but can be easier to understand their movement.
In general, both views have pros and cons. 3D is more dynamic, and 2D is easier and has more control. Some of the problems depends on how well the UI design is done. With these thoughts, 3D is picked, as 2D will be too linear gameplay, as it can be difficult to let the player have different possibilities, without implementing a lot of features.

5.4.2 Action diagrams

This diagram shows the different possible actions the player can perform on their turn, and the different variables they may need to be aware of. This is also made to make the implementation of the game easier, to have common variable names. Different actions can trigger other actions and impact different variables collectively. This also makes it easier to balance the game.

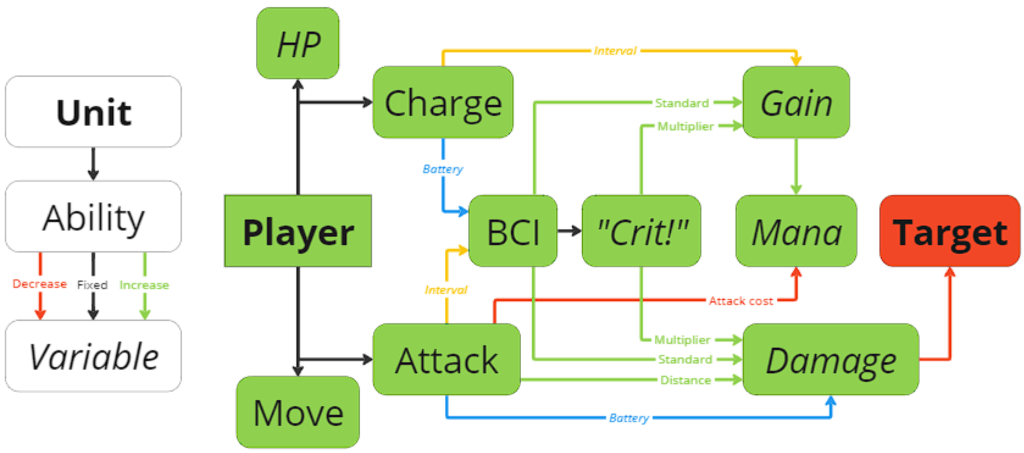


Figure 5.1: Action and enemy interaction diagram

To visualize the steps of doing different actions, a storyboard was created. With simple symbols and interface, the actions "Movement" and "Attacking" are shown. Each story board starts at the beginning of a turn and ends after a complete interaction. The different actions can be mixed as the user likes and in different situations.

Movement phase

Movement is done by clicking on the desired square, the user wants the player to walk towards. The area the user can pick is visualized with a big outline. After clicking, the player moves towards the clicked square with the shortest route, avoiding obstacles.

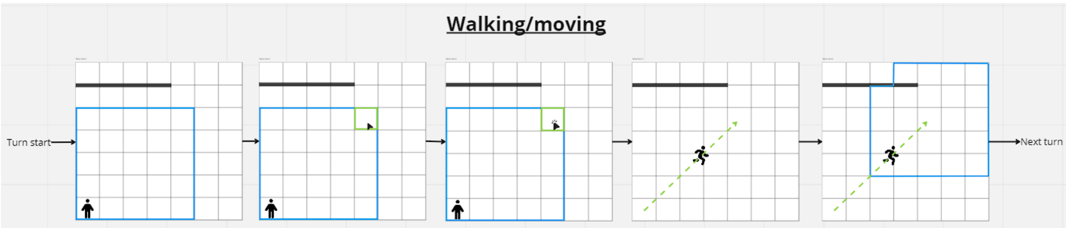


Figure 5.2: Storyboard of movement phase

Attacking phase

The attacking phase starts when a player is spotted by an enemy or engaging a fight themselves. The enemy has a visualized area, where they will spot the player. The character that starts the fight, has the first turn. The player starts shooting when the enemy is in the player's line of sight, while the enemy start running towards the main character to hit them when within melee range. When the interaction is done, one of the characters would normally be dead, and the game can continue.

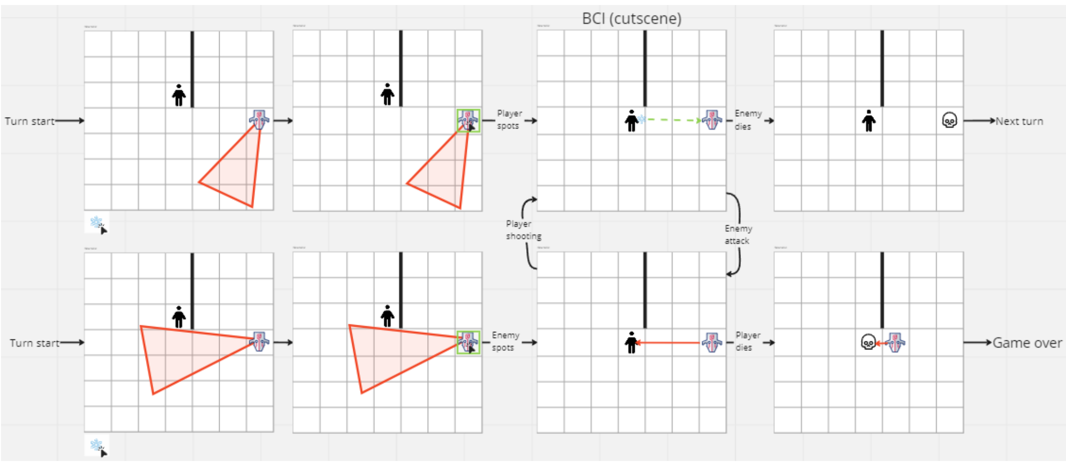


Figure 5.3: Storyboard of attack phase

Action order and game flow

The illustration underneath is a representation of the different actions a player would take doing a standard map. Each block acts as a single turn. The upper blocks are when the player does motor imagery. The light red ones are fails, and the light green ones are success. The blue is when the player charges their mana, brown is when the player moves the player around the map, yellow is when the player attacks an enemy, and red is when the player is attacked by an enemy. The upper diagram is how the MI interaction would look when having an "interval" pacing, and the bottom one is when a "battery"-pacing is being used. The next section (5.4.3) talks more about the different pacing. As we can see here, the MI prompts are closer to each other and grouped more in battery, compared to interval, which is more singular and spread out.

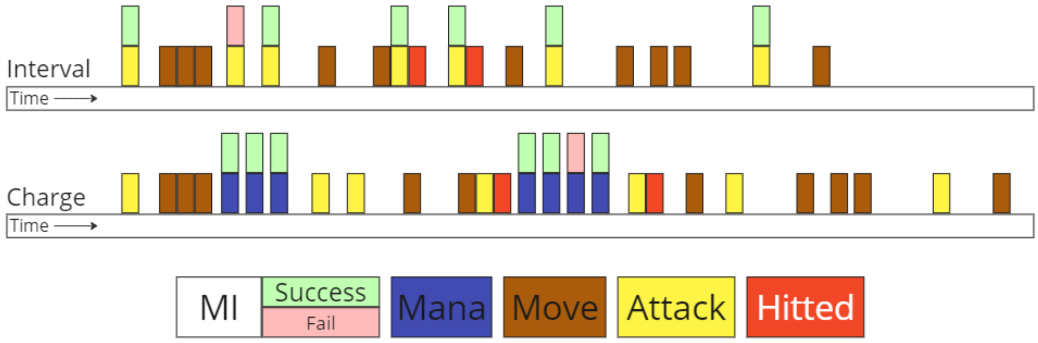


Figure 5.4: Desired game flow of the two versions

5.4.3 BCI integration in terms of "battery" and "interval"

The integration of BCI depends on the selected type of pacing discussed in Section 3.5. As mentioned in the problem statement, this project is looking into the two types "battery", and "interval". "Interval" seems more fun and interesting, as it is more action based, but "battery" has potential to let the user focus more on the BCI repetitions and stay calm.

Battery works in the context of the game as "mana" being charged for the player to be able to use their magic, to attack enemies. The player can recharge their mana as much as they want to by using one turn. The amount of mana an attack costs, how much mana the player can contain, and how much mana each recharge gain, will change the amount of BCI and impact the attack speed and pacing of the game.

Interval is invited in terms of each attack. Attacking depends on how well the BCI was executed. For both types the damaged dealt, enemies health, and hit chance, will impact how many times the BCI will have to be executed. By tweaking these parameters, the game can control how many BCI repetitions the player will have

to perform, in regard to the enemies.
Other ways of manipulating the amount of BCI repetitions is by increasing or decreasing some of the following features. The goal is to have the player do the most amount of BCI repetitions, without being stressed, or losing interest in the game.

- Amount of MI events to perform an action.
- Amount of enemies in total in map.
- What types of actions will require BCI.
- Random attack of enemies out of nowhere.
- Turn countdown timer to force the player to take their turn.

5.5 Prototype

5.5.1 Paper prototype

As mentioned earlier, two different pacings were looked into, "battery" and "interval". To explore these in the context of the project, a paper prototype was used. The prototype was made using materials from board games. It used cutouts as enemies, a minifigure as the player and the whiteboard grid was used to make the level. To simulate the BCI succeeding and failing, a die was thrown and if you got a 1 or 2 it was a fail, while 3 through 6 was a success. A couple of tests were conducted within our group, testing the different customizable aspects of the game, like health and the mana cost of attacks. When some acceptable settings were found for both pacings a test was conducted on a participant from outside the group. We used to think aloud method, so we would know what the participant thought of both pacings. The participant liked the "interval" pacing a little more than the "battery", he thought it interrupted the flow of the game that you had to stand still and recharge. He proposed that it might give a better flow if you got more mana per charge.

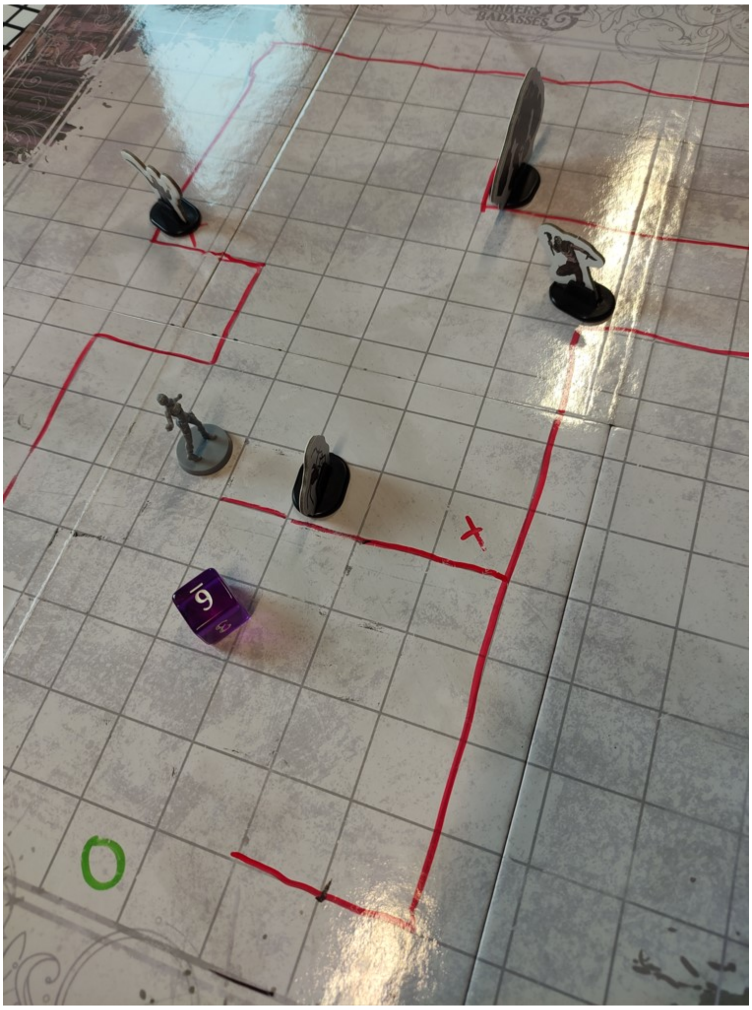


Figure 5.5: Combat using the paper prototype

5.5.2 Digital prototype - tutorial and level design

Both the tutorial and Level 1 is designed with sketches first, before implementing it. The sketches can be seen as Figure 5.7. The arrows show the direction and path of the enemies, and the mouse shows the interaction the player is supposed to perform. This helped to easier manipulate, and better understand the aspects before making it. The tutorial is designed with different areas, each having a purpose to teach the player a single element at a time. The elements they must learn about are "movement", "attacking", "charging", and "tile interaction". First area, the player can only move. Second area, the player has to kill an enemy. Third area is only when in battery mode, as it will have them charge enough to kill another enemy. Fourth area makes the player understand grass, water and bridge tiles. Fifth area teaches about remaining tiles such as stone, crystal and trees. Fourth and fifth area will also help the user to understand the enemies' behaviour.

Level 1 is made with focus of short areas, so the player does not have an advantage, as they have ranged attack, while the enemies only have melee attack. The level is separated in forest, and mountain areas, to get a better map understanding. The enemies patrol to advance most of the map, but still lets the player have some "quiet" area, for them to charge their mana. The map has a main path being a circle going through the whole map, with some dead ends and smaller circles connected to this. The main path is guarded by two enemies, that the player has to encounter to get around the whole map. The level may change slightly when implementing, after play testing it, and a better understanding of how the enemies and core gameplay is optimized with the level design.

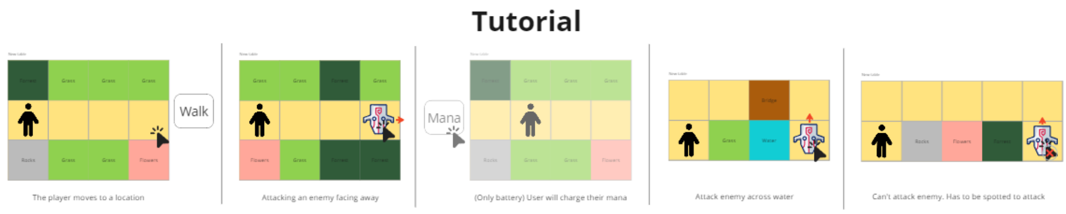


Figure 5.6: Tutorial design

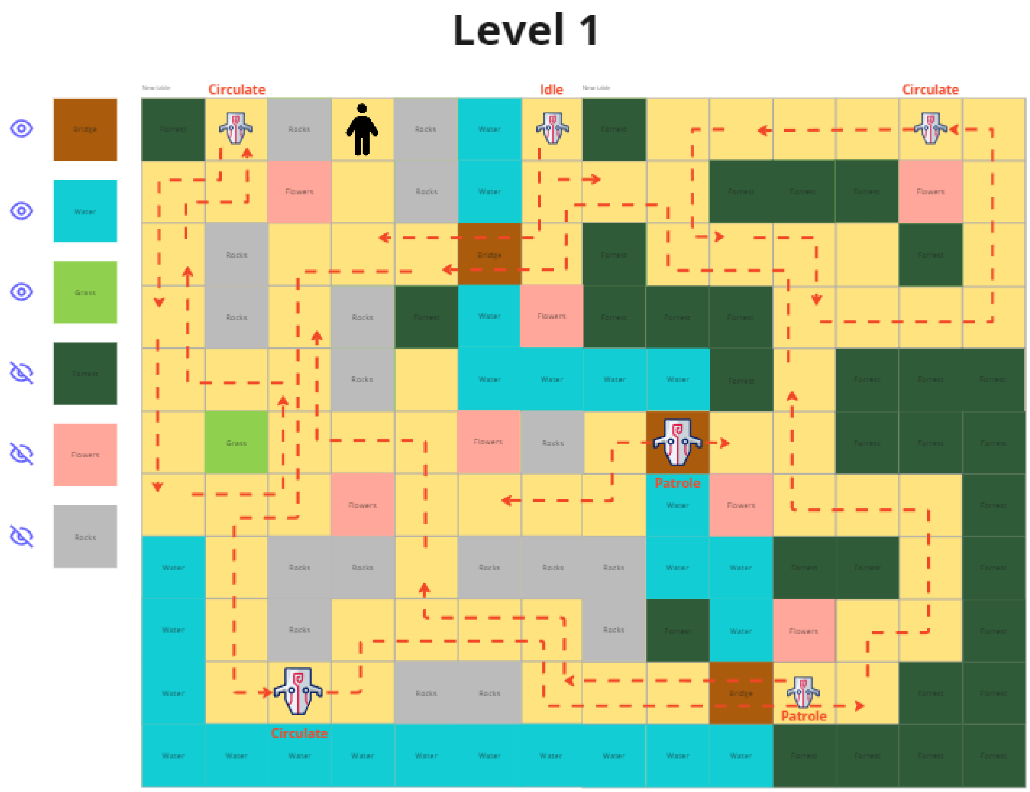


Figure 5.7: Level design

5.5.3 Pilot test

A pilot test was conducted to ensure the game made sense and worked as wanted. This test showed that the map was too small, and therefore the map was made bigger. When the map got bigger more goblins were also added, and to ensure the game is not too difficult, some of the goblins had their health reduced.

Chapter 6

Implementation

The implementation of the product has been developed in the video game software https://unity.com/. The product utilizes premade 3D world interaction such as "BoxColliders", vector calculation, audio-, animation managers, etc. The scripts are written in C and are made as Object Orientated Programming (OOP). The "gameObjects" in the game are made to prefabs, to reuse them in multiple levels and tutorial.

6.1 Program overview

The system is split into different categories, there are managers, movement, UI, and units scripts.
The manager scripts are in charge of managing the overall game, it has audio manager, game manager (checks if a player is dead and restarts the level), turn manager, game mode managers, and a win condition manager.
The movement scrips are responsible for all the movement in the game, there is one script called TacticsMove that has most of the code for movement, since enemies and player need to access much of the same information in order to move. The other script is the tile functionality, which finds neighbours, checks if a tile is walkable, and resets the tile.
The UI scripts are making the infographic text, the BCI slider and the shaking, basically it creates all the UI elements that are not on top of an enemy or the player.
The last category is units, in this category all the scripts that control anything with either the player or the enemy directly is implemented. It has enemy functionality, how they make their decisions and patrol paths. It is also in this category all the animations and health/mana UI are implemented. The player's resource management (health and mana) are also created in this category.

6.2 Map and Grid

The map is a grid and the grid is made so that each square consists of a cube 3D gameObject measuring (1, 0.1, 1), each square will be referenced to as a tile from this point on. The tiles which the player is able to navigate to has a script attached to them which adds the functionality needed in order to make the movement scripts work. The script attached to the tiles is in charge of:

- Changing the colour of tiles that are selectable, the target and the current tile. so the player has the information needed in order to navigate in the map, see figure 6.1 for an example.
- Resetting the tiles to their original state after a movement has occurred.
- Identify the neighbours of the tiles.
- Checking if a tile is traversable.

The last three items are needed for the two searching algorithms, breadth first search and A*, used for the player and enemy movement respectively.

Figure 6.1: Demonstration of the different tile colours. The blue is the one that is selected by the player, red are the ones that are selectable, and yellow is the one the character is currently standing on.

6.3 Turns

As the game is turn-based, a script is made controlling which character's turn it current is. An overall boolean control if it is the player or enemies turn. A number of variables controls the interaction of a character's turn. When a character's turn starts, it activates the character, and a wait boolean. The character will then take an action, and execute it. The wait boolean will wait to pass the turn, until an animation of the character has finished. When all variables are correct, the turn will be passed on to the next characters. The enemies will have their turn separately, and pass the turn to each other. If an enemy is dead, the turn will simply continue to the next one. When they have all been iterated, the turn will pass over to the player.

6.4 Character Abilities

Each character has the ability to walk around the map, spot their target, attack them, and the player can charge mana. The enemies are following a decision tree, that tells them what action to perform next turn, while the player can decide from them self if they would like to move, attack, or charge if the game version is "battery" mode.

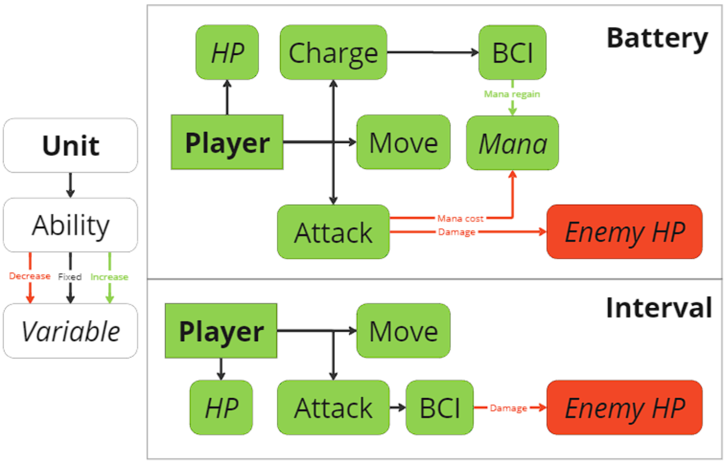


Figure 6.2: Players possible interactions in the two versions

6.4.1 Movement

There are two types of movement implemented, player movement and enemy movement. They will be explained separately, even when they have some things in common.

Player Movement

The player character moves when the player clicks on an available tile and presses the accept button. This functionality is achieved mostly with a breadth first search (BFS) algorithm, a move function, and some mouse click checks.
BFS is an uninformed searching algorithm, it does not take the location of the goal into account, it ignores where it is going until it finds the goal and reports success. BFS is implemented with a first-in, first-out (FIFO) queue. [24] In the game, the BFS is in charge of deciding which tiles the player can click on and move to, i.e. it makes a tile selectable based on the distance the player character is allowed to move and adds them to a list.
The BFS code can be seen on code snippet 6.1.

```
public void BFS() {
    ComputeAdjency(null);
    GetcurrentTile();

    Queue<Tile> BFSsearch = new Queue<Tile>();

    //add currenttile to queue
    BFSsearch.Enqueue(currentTile);
    currentTile.visited = true; //not wanna come back to this.
    //currentTile.parentTile = null; //Find it later when backtracking.

    while (BFSsearch.Count > 0){ //Continue until empty
        Tile t = BFSsearch.Dequeue(); //process one tile, pop off the front.
        selectableTiles.Add(t);
        t.selectable = true;

        if (t.distance < moveRange){
            foreach(Tile tile in t.adjacentList){ //Anything adjacent to it, will set the original tile as parent.
                if (!tile.visited){
                    tile.parentTile = t;
                    tile.visited = true;
                    tile.distance = 1 + t.distance; //can add 1 everytime in an arch away from the start tile
                    BFSsearch.Enqueue(tile); //add it to the queue.
                }
            }
        }
    }
}

```

Kode 6.1: BFS algorithm implemented in C#

For the BFS function to work, some information is needed, we extract information about; the adjacent tiles, the current tile and the target tile. This information is extracted from the first two functions calls on line 2 and 3 in the BFS code snippet (6.1). The following will be a walkthrough of how this BFS code works, and all line numbers are references to code snippet 6.1.

After the two function calls, a new empty queue is created on line 5, called *BFSsearch*, it is used to keep track of the tiles that have not been visited yet during the search.

Next, the current tile is added to the *BFSsearch* queue, and it is marked as visited because we do not want to search that tile. The rest of the code is inside a while loop, that will run as long as there are tiles in the BFSsearch queue.

On line 13 the first tile in the *BFSsearch* queue is dequeued and assigned to a variable we call t. Then it adds the current tile to a list of all the selectable tiles, and marks it selectable, with the variable *selectable* set to true.

Line 17 starts and 'if statements', that check if the distance of the current tile from the tile we started at is less than the maximum movement range.

The next line starts a for each loop that loops through each of the tile that is adjacent to the current tile, which in this code is a list called *adjacentList*. If these tiles have not been visited, yet it does four things, (1) the parent of the adjacent tile is set to the current tile (2) it marks them as visited, (3) the distance attribute adds one to the current tile's distance and (4) adds them to the *BFSsearch* queue.

The move function is moving the player character from one tile to the next. Each step along the path it finds is a tile. The movement is simply adding velocity multiplied with time to the character's position.

Enemy AI Movement

The enemy moves automatically when it is the given enemy's turn. In section 6.4.3 it is explained how the enemies move when they have not spotted the player. When an enemy spots the player it will go into "attack" mode, this is where the AI searching algorithm will begin.

It is an A* searching algorithm designed for finding the shortest path in the grid to the target tile. An **A*** algorithm is characterized by a heuristic function that estimates the cost in this case distance, from the current tile to the target tile. It also does a cost evaluation, where it evaluates the cost with two factors, the G cost (in this context the actual cost from parent tile to current tile) and the H cost (estimated cost from current tile to the target tile) based on the heuristic function. It also has an f cost, which is a combination of the G and H cost, the f cost is the priority of a tile to be selected for the next expansion of the search. The beginning of the A* Star function can be seen on code snippet 6.2. The function takes one parameter, a tile object that will represent the target tile.

Then two lists are initialized, an open and closed list, by common practice, the

open list contains all tiles that have not been processed, and the closed list contains all the tiles that haven been processed, it starts by adding the starting tile (currentTile) to the open list, this is the tile where the path finding starts.
Line 8 and 9 calculates the heuristic cost. This is the way it decides which way is the shortest to the target. This is calculated with the Euclidean distance between the target tile and the current tile position. Then we set the F value of the current tile as the same value as the heuristic cost.

```
1 protected bool FindPath(Tile target){ //enemy astar
2     ComputeAdjency(target);
3     GetcurrentTile();
4     //Two lists, open and closed list.
5
6     List<Tile> openList = new List<Tile>(); //any tile that has not
       been processed
7     List<Tile> closedList = new List<Tile>(); //every tile that has
       been processed, not done until the target tile is in this list.
8
9     openList.Add(currentTile);
10    currentTile.heuristicCost = Vector3.Distance
11    (currentTile.transform.position, target.transform.position);
12    currentTile.f = currentTile.heuristicCost;
13
```

Kode 6.2: A* algorithm implemented in C#

The main loop of the code can be seen in code snippet 6.3. It starts at line 3, which states that as long as there are tiles in the open list (tiles yet to be processed) it continues.
Line 4 assigns the tile from the open list with the lowest f value to the variable "t". It finds the lowest f with the function LowestF, which returns the tile with the greatest potential to get where we want it to go. Then we add that t tile to the closed list because it has been processed.
Line 7 is making sure that the enemy will not stand on the same line as the player object. If the t tile is the same as the target, it will run the FindLastTile function and then call the MoveTo (makes the enemy move). The FindLastTile(Tile t) function is responsible for finding the tile that is one step BEFORE the target tile in the path it has calculated. It is doing so by checking if the target tile is occupied by the player, and if it is not, then it will push it into a stack. Then it traverses the parent tiles from target tile to start tile until it reaches start or max movement is reached.
Line 13 starts a for each loop that processes all the neighbours, assuming the target tile has not been reached yet. Each tile has three possible cases, which is checked with the if, else if and else statements in the code. The three cases are (1) The tile is already in the closed list, then it should not do anything. (2) It found a way to

a tile that is already in the list, so it found more than one path to a tile, then we have to calculate the cost and compare the temporary g cost with the previously found g cost. The temporary g cost is the sum of gCost of tile t and the distance between tile t and the neighbour tile. If the new path it finds is faster, the parent tile, gCost and f values are updated to be the fastest path. (3) If the tile is not in the closed or open list, then it has not been encountered before, it sets the parent tile to t, and calculates the gCost, the heuristic cost and the f value. The last thing it does is adding the tile to the open list for later processing.
This loop continues until the open list empty, and all possible paths have been explored, or until the target tile is found. If the open list is empty before it has found the target until there is no valid path to the target.

```
//loop the open list. if we hit 0, without getting to the tile, we have no path
while (openList.Count > 0){
    Tile t = LowestF(openList);
    closedList.Add(t); //add to closed list, we have found the closest route to this t.

    if (t == target){ //cannot step on target, because there is a unit, stop algorithm 1 node before end
        AStarTargetTile = FindLastTile(t);
        MoveTo(AStarTargetTile);
        return true;
    }

    foreach (Tile tile in t.adjacentList){
        if(closedList.Contains(tile)){
            //do nothing
        }
        else if (openList.Contains(tile)){
            //compare g scores
            float temporaryG = t.gCost + Vector3.Distance(tile.transform.position, t.transform.position);

            if (temporaryG < tile.gCost){ //if this is the case, then it is faster
                tile.parentTile = t;
                tile.gCost = temporaryG;
                tile.f = tile.gCost + tile.heuristicCost;
            }
        }

        else{ //add the new node, calculate cost
            tile.parentTile = t;

            tile.gCost = t.gCost + Vector3.Distance(
            tile.transform.position, t.transform.position);
            tile.heuristicCost = Vector3.Distance(
```

```
                tile.transform.position, target.transform.position);
                tile.f = tile.gCost + tile.heuristicCost;

                openList.Add(tile);
            }
        }
}
//problem - what to do if there is no path, not implemented.
Debug.Log("Path not found");
return false;
}

```

Kode 6.3: A* algorithm implemented in C#

6.4.2 Spotting

All characters are able to "see". Multiple rays detect targets in front of the character. Enemies "looks" at all times, but the player-character only does when the player, tries to attack an enemy.
The method implemented to do this, is seen underneath called *Eyes()* in the code snippet 6.4. The static variables used for the method are "inc" the frequency of rays at a minimum of two, "FOV" the field-of-view and how wide the rays spread out in degrees, and "distance" the distance of the rays and how long the character can see. Other variables are "targetTag" the tag the rays are looking for, and "attackTarget" the variables set as spotted target.
The code makes a for-loop with the variable "angle" increment from *FOV* with an increase of *inc* in each step. For each iteration, a zero vector is made added with the angle from the character's forward vector with the distance as scalar. The location of the character, with the y-value set to the half height, is set to the start position of the ray. If a ray is hitting a gameObject with the correct tag, the gameObject will be set as the target, and the method will stop. If no target is found, the target variables will remain as "null". To have a full view all around the character, the FOV is set to 180, as the range of angles is going from both the negative and positive reach of FOV.

```
[Range(2, 15)] public int inc = 5;  // Increment and rate of
    raycast
[Range(1, 180)] public int FOV = 25;  // Field of view [1-180]
[Range(0f, 100f)] public float distance = 10f;  // Raycast distance
public string targetTag;  // Tag unit will target and attack
protected Unit attackTarget;  // Target to attack

protected void Eyes() {
    inc = Mathf.Max(2, inc); // Minimize increments
    RaycastHit hit;
```

```
    for (int angle = -FOV; angle <= FOV; angle += inc) {
        Vector3 targetPos = new Vector3(0, 0, 0);  // Initialize a zero-vector
        // Get angle from for-loop and object forward direction
        targetPos += Quaternion.AngleAxis(angle, Vector3.up) * transform.forward * distance;
        Vector3 projection = transform.position;  // Unit position
        projection.y += gameObject.GetComponent<Collider>().bounds.extents.y / 2;
        Debug.DrawRay(projection, targetPos, viewColor);  // Visualize raycast
        if (!Physics.Raycast(projection, targetPos, out hit, distance)) continue;
        if (hit.transform.tag != targetTag) continue;  // Only matching 'target tag'
        SetTarget(hit.transform);  // Set target to hit game object
        return;  // Break loop
    }
}

```

Kode 6.4: The "eyes" of the characters to spot each others

6.4.3 Enemy behavior

The enemies are following a behaviour path to tell them what to do when they start their turn. At the start of each enemy's turn, They make a decision from a decision tree, what to do next. They decide from the current state they are in. The state can change over time on their own turn, the player's turn, or when they finish their turn. Figure 6.3 shows a representation of the decision tree and its behaviours. The stippled lines are state change, which are waiting for previous state or turn to complete. Flow lines are continuing as normal. The ":any:" tile, is executable from any other state. At each state, if the enemy can not find a path to where they would like to go, they will try to search the area. If there is still no path found, they will go idle.

Each enemy can start in the "Patrol" or "Idle" state. The patrol state has the enemy follow their patrol points. They are made manually in advance, and are followed linear or circular. The idle state will simply have the enemy stand still and skip their turn. The enemies will go into another state depending on if they are being damaged, or has spotted the player. If at any moment the enemy spots the player, they will go into the "Attack" state. This will have them go towards the player, if the distance is greater than their walk distance, or if their walking distance will reach the player, they will walk up to them, and attack them. When the enemies turn start again next round, if they can not see the player any more, they will investigate at the last scene location. If they spot them again, they will keep attacking them.

After investigating, they will search the area, by walking to a random location in a given distance. When they still have not spotted the player, they will return to their patrol path, and wait until they spot the player or get damaged. When the player damage them, they will go into the "Damaged" state, which will have them look towards the location they have been attacked from. When they start their turn, they will go investigate the location they got hit from, and follow the decision tree.

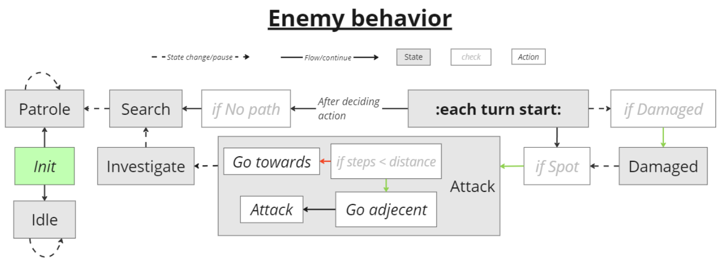


Figure 6.3: Chart of the enemies behavior

6.4.4 Attacking and health

As mentioned in Section 6.4.3 - Enemy Behaviour, the enemies simply attack by walking up to the player, and attacking them, when they are inside attack range. As the player performs ranged attack, an attack will be executed even from a distance. The ranged attack can however not be executed through objects. It works by utilizing the "Spot" method mentioned before, when the user clicks upon an enemy. This will have the player-character look towards the target, and try to spot it. If no target is fetched, the attacking will be aborted. At successful attacks, the player-character will launch a projectile towards the targeted enemy. At impact with the enemy, they will take damage according to the projectiles damage variable. All characters has a script managing their health. The health points will decrease when they are being attacked. By having a separate method, animations and sound effects can also be called here. When the health points are less than zero, the character will be marked as dead, and can no longer be interacted with, with regard to turn, tiles, and attacking.

6.4.5 Charging mana

When the game mode is set to "battery"-mode, the player has to be aware of their amount of mana they have. The program will only allow attack, when the player has a certain amount of mana points. Each attack will decrease the amount of mana points. To recharge mana, the player has to click on the "Charge mana" button. This will call the BCI prompt. To unsure the mode is following the battery model, the player is forced to perform a certain amount of successful motor imagery attempts.

After recharging mana, the player's turn will end, but can now attack enemies freely.

6.4.6 Confirm button

As the users may have a difficulty with their movements, a confirm button is implemented, to have the user confirm their action after picked it. After a tile has been picked for movement or en enemy has been selected for attacking. A problem with the interface was that, when confirming the action, the mouse ray-cast would detect tiles or enemies behind it, and pick those instead, just before confirming. To avoid this, a "BoxCollider" was added to the button to stop the mouse click ray-cast, form hitting other objects in the scene. The same was done with the "Charge mana"-button.

6.5 Models and Animations

6.5.1 Characters and Map

The character models used in our project are all made in Blender [8]. The characters are made using box modelling, the girl was made with inspiration from a tutorial, and the goblins are made with our idea of how goblins look, in mind. The map is made out of many different square tiles. All the tiles are from an asset bundle on itch.io [16], and they are public domain and can be used for commercial use as well as educational use.

6.5.2 Animations

Unity's build in animation trees are used to control the animations of both the player and the goblins.

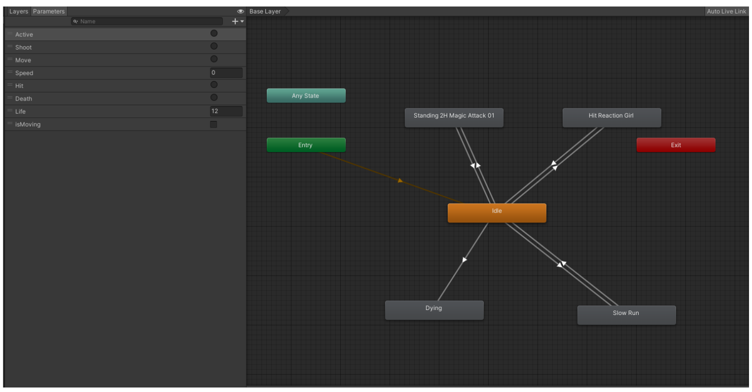


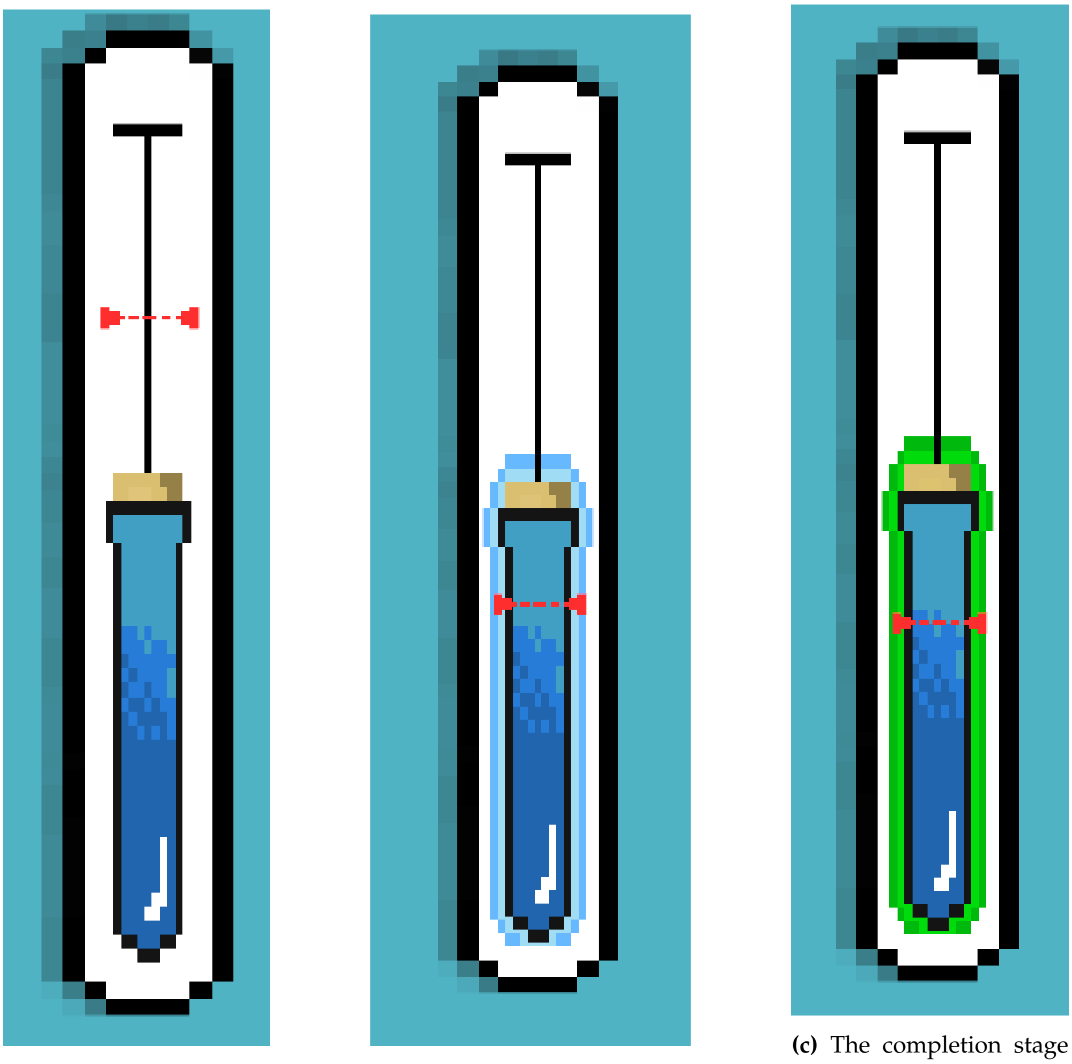
Figure 6.4: Chart of the enemies behavior

As seen in figure 6.4 we use a mix of trigger and float variables to control the different animations. The triggers are very straight forward, for example if the shoot trigger is called, the 2 handed shooting animation will play and then return to the idle animation. The speed float is used to control the running animation, this is done by accessing the TacticsMove script where we actually move the model, and get the velocity of the player. After that we set the speed to the absolute value of the velocity, so it works on both the positive and negative directions on the x and z axis, and then says that if speed is above 0.1, the running animation has to play on a loop until the velocity and therefore also the speed gets back to zero. This way the animation is running as long as the model is moving and does not go back to idle before the model is standing still.To trigger the shooting animations, a check is performed, to check if the game is in the Battery or the Interval mode. If the game is in Battery, the shooting animation is performed upon clicking the confirm button, but in Interval the shooting animation is performed after the BCI check. To control the hit reaction on the player, we check if there is a change in the player's health. If a change is detected, the program then checks if the health is zero or above zero. If the health is above zero, the hit reaction animation will run once, and it will play the hit sound at the same time. If the health is zero the death animation will run once, but it won't go back to the idle animation, instead the player will just stay down and won't be able to move or anything. The goblin animations work a little different. Because the player shoots a magic projectile, an OnCollisionEnter check was used, to trigger their hit and death animation. Again, a health check was used to see if the hit or the death animation should trigger. The running animation works in exactly the same way as the players, because the goblins also use the TacticsMove script to move. All the above-mentioned animation have been made with the help of the website Mixamo [6].

6.6 Motor imagery and BCI

6.6.1 In-game motor imagery prompting and visualization

As implemented, the BCI interface is built around the unity slider UI element, with textures being created using the GIMP open-source image editor [26]. These textures are created to be visually coherent with the rest of the UI elements using a simplistic 8bit pixel art style. During a BCI prompt event, the slider progresses through three states. **Rest** 6.5a the first portion of the slider indicates to the user to relax. **Prompt** 6.5b the sliding bar enters the second half of the slider, its traversal speed is halved, and a neutral bar highlight element becomes visible. This second stage asks the user to recall a predefined motor imagery sensation. The third and final stage, **Completion** 6.5c, is triggered when either the input manager game element registers a motor imagery event during the previous **prompt** stage 6.5, or when the sliding element reaches its end. The completion stage has two options of Success or fail, with each of their own respective response depending on the selected game mode. When in battery mode, a failure triggers the BCI prompt to vanish, reset, then reappear, requiring 4 successful prompts before completing. When in interval mode, both successful and failing completion stages trigger a turn end.



(a) The rest stage **(b)** The prompt stage **(c)** The completion stage (successful variant)

Figure 6.5: The three stages of the BCI prompt

```
...

        if (Slider.value > 0.99)
        {
            Fail();
        }
    }
}

public void FilterInputSuccess()
{
    if (Slider.value >= 0.418f && StartBciPrompt == true)
    {
        Success();
    }
}

...

```

Kode 6.5: BCI sucess filtering function

The timing and speed of the UI prompt is defined by a public bciPromptDuration in combination with a prompt speed used to alter the slider speed during the "Prompt" stage. So, a duration of 7 with a BCI prompt speed of 0.5 will produce a total duration of roughly 10.5 seconds. The Code below 6.6 runs every update and implements the above slider movement. The slider itself changes position based on a value from 0 to 1. Between each BCI event the reset function is called which resets all necessary variables ready for the next BCI event call.

```
...

    time -= Time.deltaTime * currentSpeed;
    Slider.value = 1 - (time / BciPromptDuration);

if (Slider.value >= 0.418f && StartBciPrompt == true) {
    Highlight.enabled = true;
    currentSpeed = promptSpeed;

...

```

Kode 6.6: Slider progression code

6.6.2 Software simulation

Within the BCI slider scripts update function, a wizard of oz implementation exists, allowing for the conducting of a test without the functioning BCI hardware. It functions by at program start choosing a random success chance between 65 - 85%. This is the average range for a functional BCI to detect a motor imagery event according to previous cited testing citeBCIsuccessRate. Every time the BCI prompt is triggered, a random potential success position is chosen together with rolling against the aforementioned percentage to determine whether or not the user will succeed during this BCI prompt. This has the effect of creating the most accurate artificial representation of actual BCI hardware. When the program determined the user will succeed, the "force success" function of the provided BCISim script is called. This was coupled to the existing BCI framework as to not limit functionality of functioning BCI hardware in the future. [21]

```
void Start() {
    ...
    successChance = Random.Range(successChanceLimits[0], successChanceLimits[1]);
}

void Update(){
    ...

    if (wizardOfOz) {
        if (!timeChosen) { //Prevent Repition
            if (SuccessRoll()) { // Roll for success
                // Choose sucess point
                startPoint = Random.Range(0.2f, 0.78f);
                willSucceed = true;
            }
            timeChosen = true;
        }

        if (!forced && willSucceed && Slider.value >= startPoint) {
            ForceSuccess();      //Call the forced sucess function
            Invoke(nameof(ForceSuccess), 3.5f); // Release the force sucess function
            forced = true;       // Prevent repition
        }
    }
    ...
}

```

Kode 6.7: Wizard of oz implementation

Chapter 7

Evaluation

This chapter explains the evaluation process, the setup and other relevant information regarding the evaluation.

7.1 Methodology

The aim of the evaluation is to test if, and how charging and interval based pacing compares, in terms of a player's engagement in the prepared game made for upper limb motor imagery training, for children with cerebral palsy. The experiment design is within-subjects, each participant will try both the charging and interval based version of the game. The order the participants will test the versions will be changed between each participant, this is an effort to reduce practice and bias effects.
After they have tested each version of the game, participants will be asked to answer a short questionnaire regarding their experience with the version they tried, mostly concerning their perceived pacing and engagement. After this, they will be asked to answer a questionnaire regarding their experience of the game as a whole. See the questionnaires in the supplementary material. Then lastly they will be asked a few questions in a semi structured interview.
The test will be Wizard of Oz, since the BCI input to the program will be fabricated by the program. It is made to give the user a success rate between 65 - 85% because a study with a similar target group by Jochumsen et al. estimated the motor imagery success rate to be $\sim$75%, which we add and subtract 10% to make it seem more natural [13].
The participants in the evaluation will be healthy, able-bodied adults for ethical reasons, and they are able to give their full consent themselves.

7.2 Setup

The setup has a focus on simulation and hiding the fact that the BCI is not being used, from the participant. Additionally, the user has to sit with the laptop, as it has touch screen and therefore must contain the program running on it. By connecting an external monitor, keyboard and mouse to the laptop, a tech facilitator can control the laptop from across the user. This will allow hiding potential information and not have to grab the laptop away from the user. A wireless controller will allow the tech facilitator to force a simulation of the BCI and overwrite it, without the user noticing. The main facilitator will be by the user's side to explain the process and be at their assistance.The setup, can be seen as Figure 7.1.

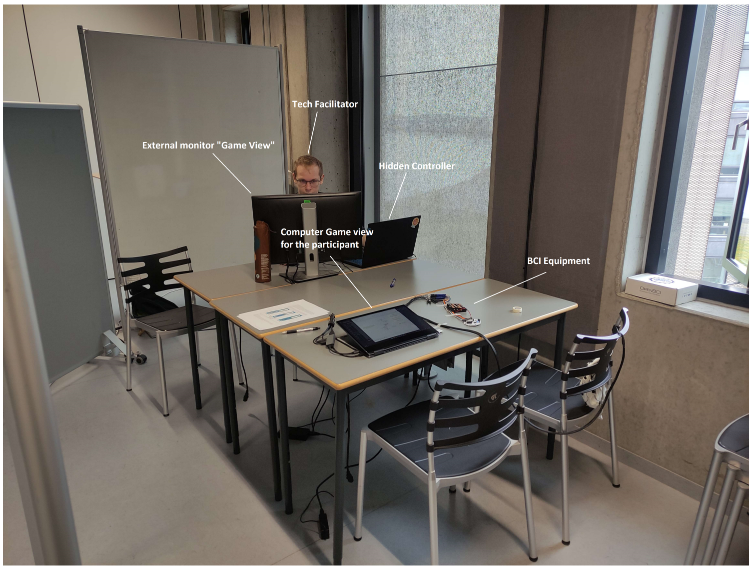


Figure 7.1: Setup of evaluation of testing

7.3 Procedure

This section covers the procedure step by step.

- The participant will be welcomed, and the facilitator will explain the purpose of the project and the goal for the test.
- The participant will be asked to read and sign the consent form.
- The participant will go through the configuration with the BCI.
- They will then play through 3 very short tutorial levels which will show how the game mechanics work.

- Then they will play the main level with the first sequencing - e.g. interval.
- When they are finished, they will answer the first questionnaire.
- The participant will then play the same levels again with the other sequencing method, which in this case would be battery.
- When the level is cleared they are asked to answer the second questionnaire which will be identical to the first.
- The participant will then answer a questionnaire about the gameplay in general, (see section 7.5.1)
- Then the short interview will be conducted.
- After this, the participant will be thanked for their time, the truth about the BCI will be revealed, and the test is finished.

7.4 Logged data

This data should tell us if they make more or sufficient MI attempts per playthrough of the game. The data can also be used to discover any errors in the game. And if they succeed or fail equal amount of times in doing MI attempts in either of the types. Measurements, besides the metadata:

- **Events:**
  - Player attacking
  - Player Moving
  - Motor imagery success
  - Motor imagery fail
  - End player turn
  - Start player turn
  - Start goblin turns
  - Win condition - test complete/end game
  - End goblin turn
  - Goblin death
  - End tutorial
  - Player hurt (take damage)
  - Regenerate mana

- Player death
  - Scene start
- **Others:**
  - Goblin name
  - Health points
  - Damage taken
  - Mana points
  - Game mode
  - Number of times players has made MI attempts
  - Number of successful MI attempts
  - Number of failed MI attempts
  - Whether the player attacked, attack counter
  - Whether the player charged
  - Whether the player takes damage
  - How many times the player moves

7.5 End of test

7.5.1 Questionnaire

The questionnaire is split into two parts, one for each of the game versions and one for the overall gameplay experience. The full questionnaire can be seen in the supplementary material.
The game specific version questionnaire is made by the group to get the specific information needed about the different experiences with the two versions.
The questionnaire for the overall gameplay is built from the two questionnaires below. The questions are hand-picked to fit the goal of the evaluation. The first questionnaire is the user engagement scale, in a much shorter version [2]. The scale has 4 different categories, the categories are as follows:

- FA = Focused attention, feeling absorbed in the interaction and losing track of time.
- PU = Perceived usability, negative affect experienced as a result of the interaction and the degree of control and effort expended.
- AE = Aesthetic appeal, the attractiveness and visual appeal of the interface.
- RW = Reward factor.

The other questionnaire that we got some questions from is the Guess-18 scale, which is a shorter version of the Game User Experience Satisfaction scale [15].

7.5.2 Interview

The interview consists of the last 8 questions in the supplementary material, it is meant to let the participants elaborate on their experiences and thoughts with the game as a whole and the motor imagery event pacing in the different versions. They are asked about their thoughts on both types and what they liked and disliked about them, and their opinion on the flow of the game with each sequencing type respectively.

Chapter 8

Results

The following chapter consists of the results of 10 participants performing the tests over the span of two continuous days. Most participants are Medialogy students from 4. - 8. semester in the age from 20 to 25 years old. Participants plays in average approximately 11 hours of video games a week. They mostly played were "Action-adventure" (70%), "Shooters" and "Role-playing" (60%), and "Survival Horror" (50%).
The results show the comparison of the two versions in regard to MI perception and prompts, overall gameplay, pacing, metaphor, and preference.

8.1 Interval and battery results

This shows the comparison of the two pacings up against each other.

8.1.1 Statistical test for engagement variables

This section will cover the statistical tests conducted and how they were made. The goal of these tests is to see if there is any statistical significant difference in the players' engagement in the two versions, these questions are inspired by some of the question of the GUESS-18 [15] and the user engagement scale [2].
Each condition is tested to check if the data is parametric or not, with Shapiro-Wilk test and Mauchly's test of sphericity, and then an appropriate statistical test is conducted.

Metaphor opinion

The Shapiro-Wilk test was conducted for the data of what the participants thought about each metaphor. For the interval data, the test's statistic value was $W = 0.859$, with a p-value of 0.073, which indicates normal distribution. It is worth noting that

the p-value is close to the significance level of 0.05, which could mean that there is a small departure from normal distribution. For the battery data, the test statistic value was W = 0.868, with a corresponding p-value of 0.094, so we conclude that the data shows evidence of being normally distributed.
A Mauchly's test of sphericity was conducted, the test statistic value is W = 0.893, which a p-value of 0.635, meaning we assume equal variance of the difference between the two conditions, and we proceed with the dependent t-test. The dependent t-test result suggests that there is no statistical significant difference between the participant's most liked metaphor (t = 0.76822, df = 9, p-value = 0.462).

Performing motor imagery was frustrating

The Shapiro-Wilk test yielded a test statistic value of W = 0.727, with a corresponding p-value of $p < 0.05$ for interval version, so the null hypothesis of normality is rejected, and a statistic value of 0.905 with a corresponding p-value of p = 0.248 for the battery version. Therefore, a Wilcoxon signed-rank test was conducted, the results indicate that there is no statistical significant difference between the two conditions regarding their frustration on performing MI in either game version (V = 5.4, p-value = 0.498).

Engagement

When testing the answers to the question "Performing motor imagery broke my engagement" the Shapiro-Wilk test for the interval version resulted in a test statistic value of W = 0.873 in a p-value of 0.108 indicating normal distribution. For the battery version, the Shapiro-Wilk test yielded a statistic value of W = 0.932 and p = 0.466, also indicating normal distribution. Since the evaluation was within subjects a Mauchly's test of sphericity was conducted, it yielded in statistic value of W = 0.959 with a corresponding p-value of p = 0.847, which means we accept the null hypothesis, that the variances of differences are equal. Based on this, we do a parametric test, so we do a dependent t-test. The results indicate that there is no statistical significant difference between the engagement level in the two conditions (t = -1.358, df = 9, p = 0.209).

8.1.2 Logged Game Data

As mentioned in section 7.4, the program logs data while a participant is playing the game. This data is used to find the number of attempts, successful attempts, and attempts per minute.

Success Rate and attempts

Even if the test is a "Wizard of Oz" test, successful attempts are still logged for two reasons (1) we ask the participants how many motor imagery events they feel they succeeded and how many they feel they failed, and (2) to show the Wizard of Oz simulation implementation is working correctly.
The mean success rate for interval is 76.06% and for battery the mean is 76.27%, which means that the "Wizard of Oz" implementation worked as expected for the test, and we got the functionality needed for the test.
The attempt and success means can be seen on figure 8.1 along with the success rate for a quick overview.

Condition	Attempt mean	Success mean	Success rate
Battery	11.8	9	76.27%
Interval	11.7	8.9	76.06%

Figure 8.1: Overview of success rate and attempts in the two conditions

Repetition per minute

Motor imagery repetitions per minute is calculated as the mean in both conditions. The mean for the battery version is 2.38 MI events per minute. The mean for the interval version is 2.54 MI events per minute. On average, this sums up to at least 35.7 MI events per 15 minutes for the battery version, and 38.1 MI events per 15 minutes for the interval version. This means that we exclude one more variable, and thus make sure that the participants' opinions of either version is not affected by the number of MI events. Furthermore, one of our success criteria depends on number of repetitions during one game session.

8.1.3 Game flow

As previously mentioned in the design chapter at Section 5.4.2, a prompt for the desired flow of gameplay in the two versions were discussed. Here is a representation of the game flow of participant 0's two trails. The colours of the different tasks is the same as previous, with the addition of purple showing the enemies turn and black being enemies death.
The two graphs corresponds with the desired game flow, that the interval version is singular and more spread out compared to the battery version, which is more grouped in larger groups. This shows that the two versions had the desired type of pacing, which makes them valid.

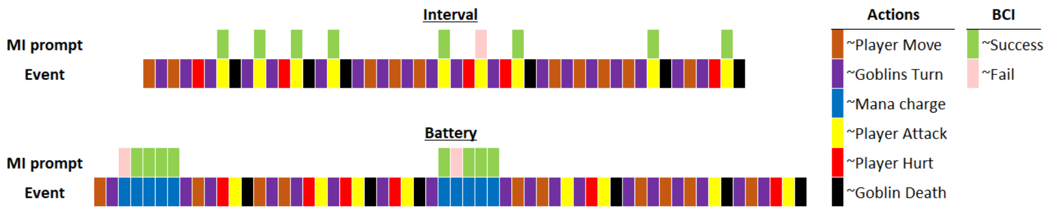


Figure 8.2: Game flow of P0

8.1.4 Overall preferred metaphor and pacing with explanation

Figure 8.3 shows the participants answers to their preference regarding the pacing, metaphor (battery or interval), and how much they liked each version in general. The results are based on questionnaire answers, regarding the pacing and metaphor they answered on a scale from 1 - 7, where 1 is interval and 7 is battery. How much they liked each version in general is a simple 1 - 5 scale.
"Preferred pacing" refers to them answering which version they liked the most, so they had to choose between them, on the figure a tendency of learning towards a preference for the interval version can be seen (M = 2.90, SD = 1.52).
When asked about their preferred metaphor, people again leaned towards interval, meaning more people preferred to do motor imagery events when attacking rather than when charging mana (M = 2.80, SD = 1.55).
When asked about how much they liked each metaphor separately on the 1 - 5 scale, from "Dislike" - "Like". The answers to this question favours the interval version again, with the mean for the battery version being 3.60 (SD = 1.43), and the mean for the interval version being 4.0 (SD = 1.05).

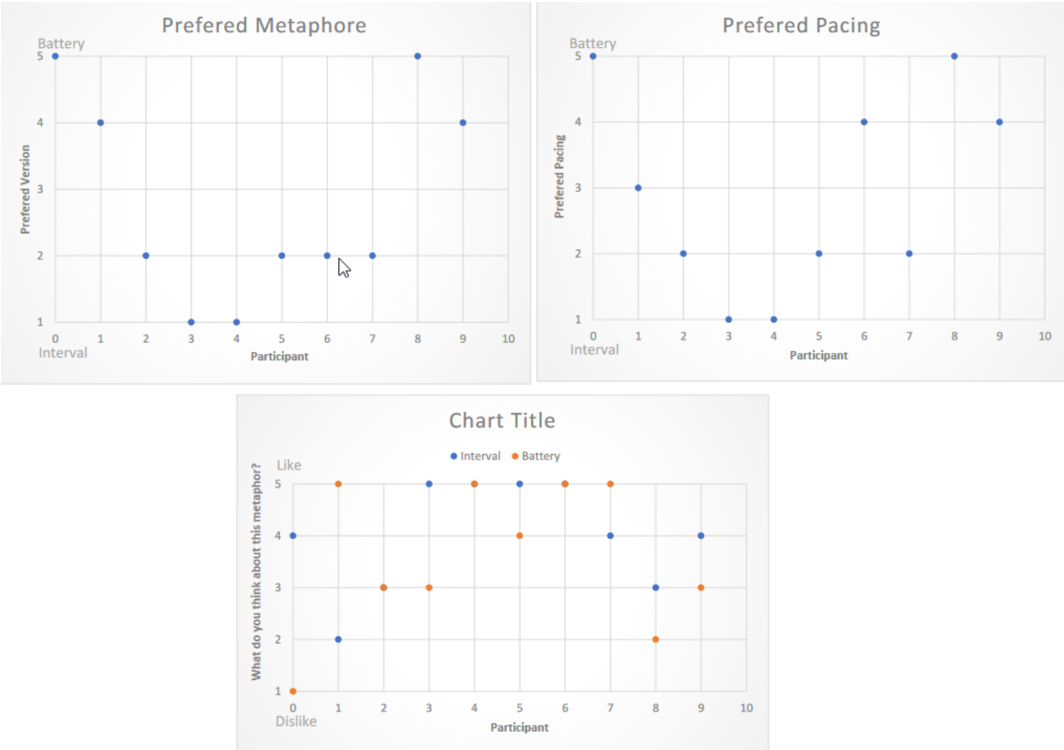


Figure 8.3: Preference of the pacing and metaphors

8.1.5 Version specific questionnaire results compared

The comparable answers from the two different versions of the game. An overview of all means and standard deviations can be seen on figure 8.4.

Variable	Interval	Battery
MI frustration	4.3 (0.95)	3.7 (1.77)
MI broke engagement	2.6 (1.17)	3.8 (1.61)
Could concentrate on MI	5.1 (1.19)	5.3 (1.06)
Timing made sense	5.1 (1.29)	4.5 (1.27)
Felt natural in this context	4.6 (1.58)	4.1 (2.02)
Found triggers logical	5.2 (1.03)	5.1 (0.99)
Though of metaphor	4.0 (1.05)	3.6 (1.43)

Figure 8.4: Mean and standard deviation of each version specific variable in the questionnaire

The first question was, "Performing motor imagery was frustrating". As can be seen on figure 8.5, the battery answers are very spread out, some did not think that it was frustrating at all, while some though it was very frustrating, the most frequent answer is "Somewhat agree" 30% of the participants answered this. While looking at interval, there is a density of answers in neutral and somewhat agree. More people found it frustrating, but not to a strong degree, 50% of the participants answered "somewhat agree".

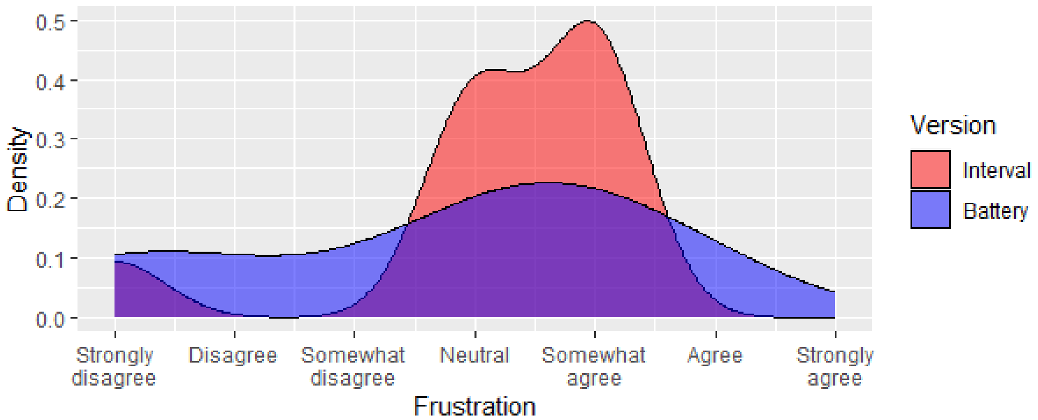


Figure 8.5: The users' frustration performing MI

The next question was, "Performing motor imagery broke my engagement". These answers show that the participants are more likely to feel that their engagement is broken when they tried the battery version of the game (see figure engagement). 50% of the participants answered "Somewhat disagree" considering the interval version, while again battery answers are spread somewhat out, with the most dense answer being "neutral" with 30%.

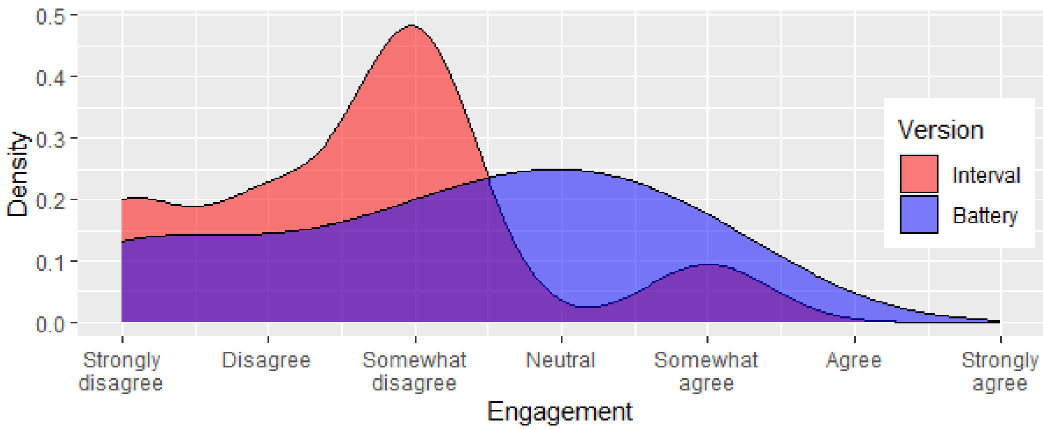


Figure 8.6: The users' engagement performing MI

The third question was, "I could concentrate while performing motor imagery", the curve on figure 8.7 shows that the participants generally could focus on performing the MI events, however a few more people found it easier to focus in the battery version, only 10% answered below "Somewhat agree", while in interval had 20% answer below the "Somewhat agree" point.

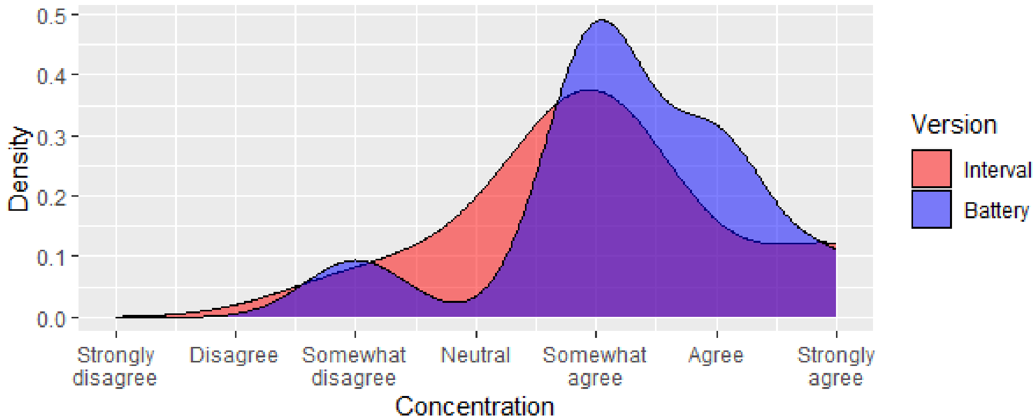


Figure 8.7: The users' concentration performing MI

The results from the fourth question, "The timing of the motor imagery made sense" showed that more people found the timing to make sense to them in the interval version than in the battery version. Most people answered "Somewhat agree" or "Agree" in the interval version. See figure 8.8.

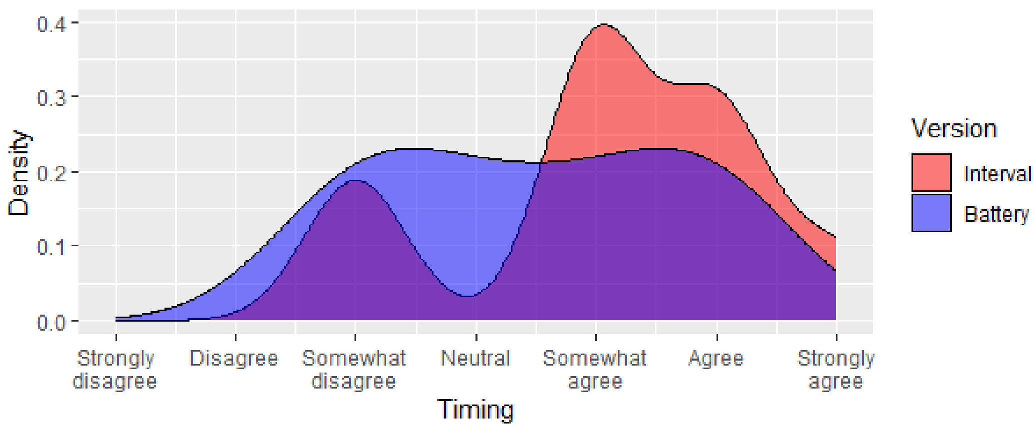


Figure 8.8: How well the user thought the timing of the metaphor made sense

The fifth question they got was "It felt natural to perform motor imagery in this context". This is evenly spread out both in the interval and battery version, however there is a slight increase in the interval compared to the battery version.

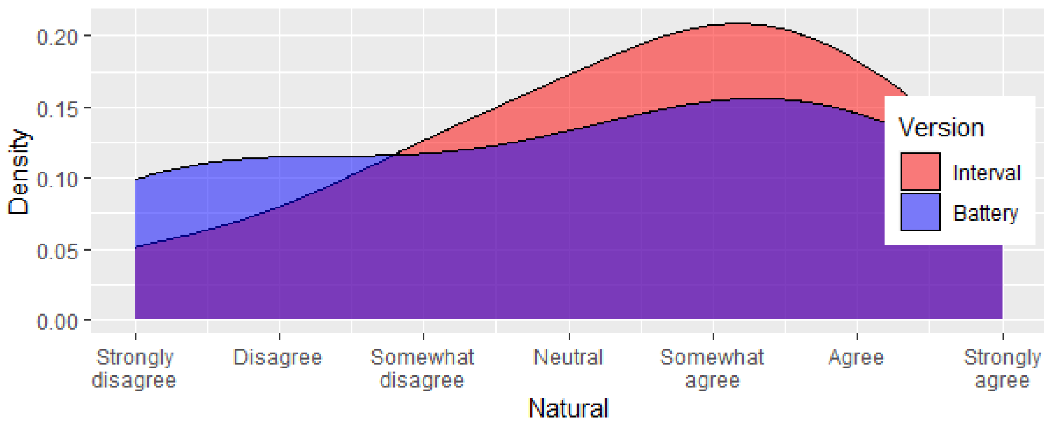


Figure 8.9: How well the user thought the metaphor felt natural

The last question "I found the trigger for motor imagery logical" had all participants answer above neutral.

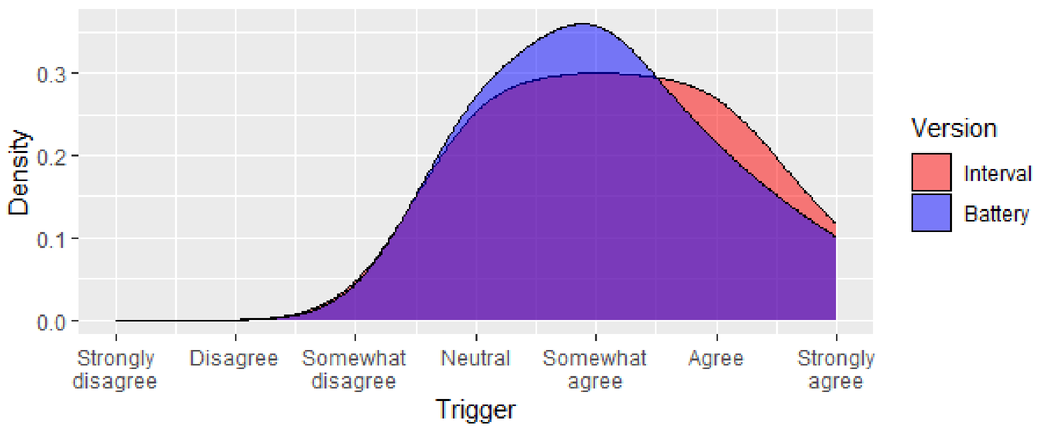


Figure 8.10: How well the user thought the trigger of the metaphor made sense

8.1.6 MI perception difference

On 8.11 we can see how many motor imagery attempts the participants thought they had, compared to how many they really had. As we can see on the blue boxes (battery) interquartile range on the success difference is between 3,5 and -3, the upper whisker is 7, the lower whisker is -5, the median is 0 and the mean is 0,5. In the fail differences the interquartile range is between 3,25 and -1,25, the upper whisker is 5, the lower whisker is -2, the median is 0,5 and the mean is 0,2. There is also a single outlier at -10. Then, if we look at the orange boxes (interval) we see that the interquartile range is between 4,25 and 1, the upper whisker is 5, the lower whisker is -2, the median is 2, the mean is 2,7, and it had a single outlier at 10, in success differences. In the fail differences the interquartile range is between 4,25 and 0 upper whisker is 7, the lower whisker is not applicable, the median is 1 and the mean is 2.

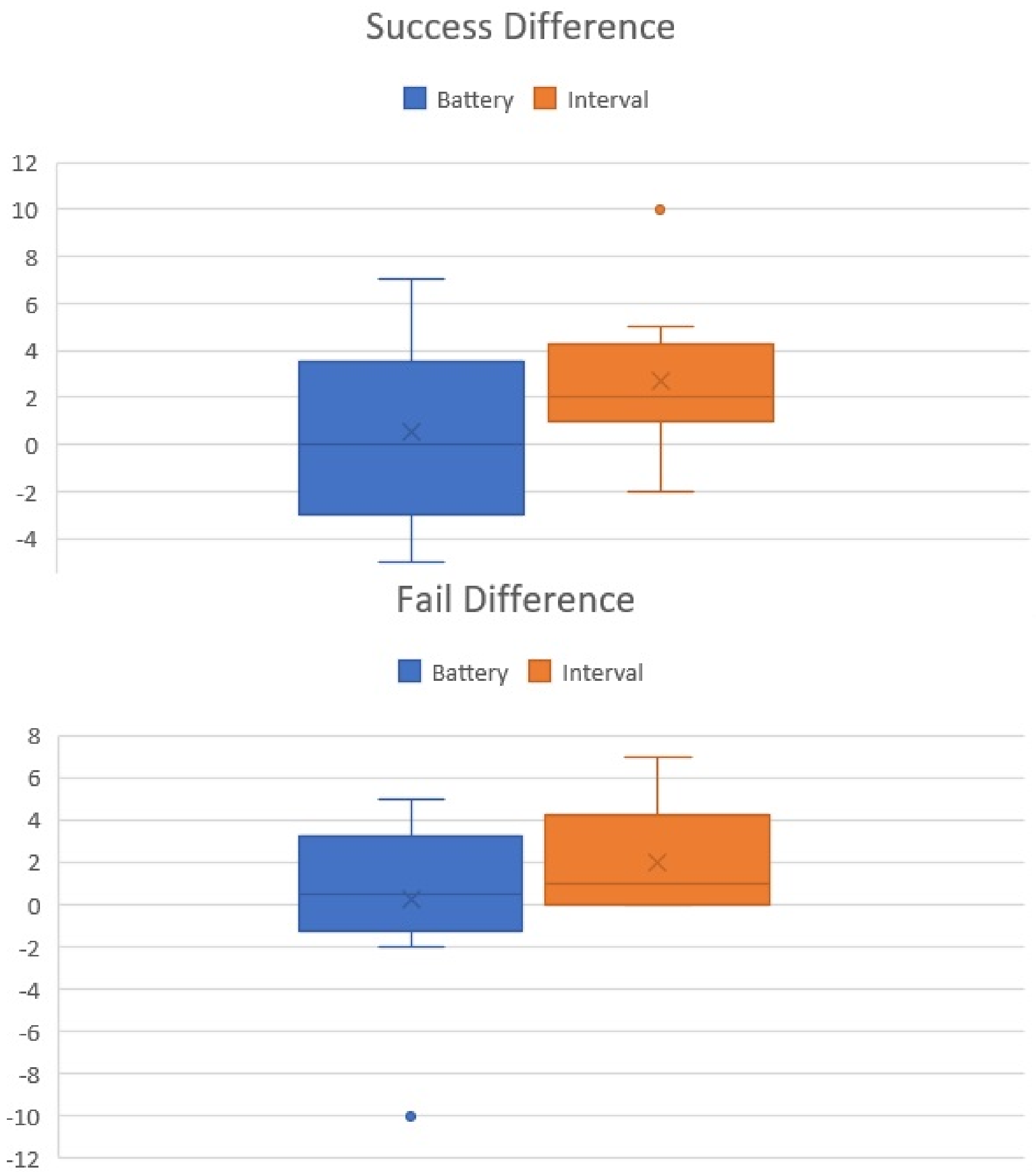


Figure 8.11: User's perception of their MI prompts in comparison to their actual performance

8.2 Qualitative results

Six of our ten participants liked the interval game mode more, one of them specifically said that the pacing felt better because you did not have to start and stop all the time like you did in the battery game mode.
Most of them had positive feedback about the interval mode. It was mentioned by

multiple participants that the metaphor of making a fireball in your hand worked better than the charging metaphor. Some of them mentioned a sense of stress in the interval mode, which as counted as both a pro and a con.
Almost every participant who had negative feedback about interval, was that they thought it was stressful at times, and thought that it might negatively impact their performance when performing motor imagery.
Many of them also had positive feedback about the battery mode. One of them specifically said he thought the management part was really cool, and others liked that it was calm when they had to perform the motor imagery.
They also had some negative feedback about the battery mode. Some of them thought it had a weird rhythm, when you had to hide and charge your mana, and then did not have to do any motor imagery for a while. This also led to the lack of punishment if they failed the motor imagery when charging, which some of them thought was a little boring.
When asked what they generally thought about both modes, most of them mentioned that they liked the pacing of the interval mode. They felt like they were in the midst of battle, which could make you feel a little pressure or stress, though that is exactly what some of the others thought was bad about it. Most of them thought the pacing in the battery mode was a little too slow. One of them thought it was a little weird, that it was slow and boring while you charged, but when you were fully charged it was fast and fun.
When asked what mode was easiest when performing motor imagery, half of the participants said interval was easier and again mentioned that the fireball metaphor made more sense to them. 30% of the participants thought battery was easier because you have to do multiple in a row, and because the enemies were not pressuring them. The last two participants had no preference.
They were evenly spread out when asked about which mode they thought they did the best in.
They had some ideas about how a game could be made that used both methods together. Multiple participants said they could imagine using one hand for attacking and then the other hand or even your leg to charge your mana. One of them said they would like an iteration of the game where you had to do motor imagery to charge mana, and then you could try for a special attack which was high risk high reward, with motor imagery as well.
Most of them felt like they were in control most of the time, and only a few of them did not feel like they were in control. One of them said it was random when he felt in control and when he did not, but when he did and succeeded he felt excited. No participants were aware that the test was a Wizard of Oz test before the facilitator told them.

8.3 Over all gameplay results

The overall gameplay results will be used to measure the gameplay experience that the participants had. This information will be used to determine if it is worthwhile to possibly continue working on this game in the future, as one of the goals is to create rehabilitation that is more enjoyable for the patient. In the 20 game trails, the game had 2 times when it had a big enough bug, for the game to loop in on itself. With the implementation of a "force turn skip" and by helping the participant with what to do. The game continued after a couple of seconds, without any other flaws. There were not at any point that the game crashed enough for the evaluation to be forced to be aborted. The figure (8.12 shows the results of the last part of the questionnaire and is made to get an overview of the general opinion of the game.

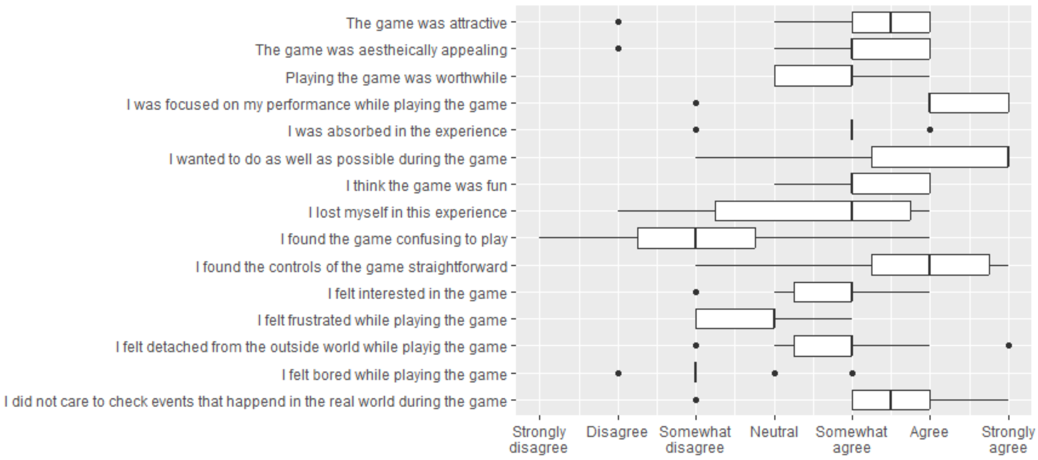


Figure 8.12: Results from overall gameplay

The mean and standard deviation for each of the variables can be seen on figure 8.13. The variables can be split into six different categories covering the game as a whole, the categories are adapted from the questionnaire they came from: (1) Focused attention, (2) Perceived usability, (3) Aesthetics, (4) Reward factor, (5) Play engrossment, (6) Enjoyment, and (7) personal gratification. The scale they answered has been altered to go from 1 (strongly disagree) - 7 (strongly agree). When reviewing this data, it is important to note that with only 10 participants, one outlier can mean a lot for the way the mean is represented, therefore it will be necessary to consider the median and outliers in some cases.

Focused attention has two questions, (1) "I was absorbed in the experience" (M

= 3.2, SD = 1.03) and (2) "I lost myself in the experience" (M = 4.5, SD = 1.43). The questions are about feeling absorbed in the game and losing track of time [2]. The mean for this category was 3.85, indicating a neutral response tendency among the participants in their absorption in the game across the category. However, the question "I was absorbed in the experience" has one big outlier which can be seen on figure 8.12, which affects the mean in the category as a whole, since the median of this question is shown as "Somewhat agree" meaning a score of 5.

Perceived usability category has two questions, (1) "I felt frustrated while playing the game" (M = 3.7, SD = 0.67) and (2) "I found the game confusing to play" (M = 3.2, SD = 1.48). The questions aim to measure the negative effect experienced as a result of the interaction, and degree of control and effort expended [2]. The mean for this category is 3.45, indicating a response tendency between "Somewhat disagree" and "Neutral" answers. The median for the category is "Somewhat disagree", which translates to a score of 3. This indicates that the participants generally did not find the game confusing or frustrating to play.

The category for Aesthetics also has two questions, (1) "The game was aesthetically appealing" (M = 5.0, SD = 1.25) and (2) "The game was attractive" (M = 5.1, SD = 1.29). The mean for the category as a whole is 5.05, indicating a tendency to response "Somewhat agree", when asked about their opinion of the aesthetics of the game.

The reward factor category has two questions, (1) "Playing the game was worthwhile" (M = 4.8, SD = 0.79) and (2) "I felt interested in the game" (M = 4.7, SD = 0.82). The mean for this category is 4.75 indicating a response tendency close to "Somewhat agree". This shows that the participants found themselves to be somewhat invested in the game experience and that it mattered to play the game.

Play engrossment has two questions, (1) "Did not care to check events that happen in the real world" (M = 5.1, SD = 1.59) and (2) "Felt detached from outside world while playing" (M = 4.9, SD = 1.10). This category is measuring the degree to which players are engaged in the game, it reflects how the game succeeds in capturing and keeping players' interest. The mean for the category is 5.0, this indicates a tendency to respond in either "Somewhat agree" or "Agree", meaning a score of 5 or 6. These results indicate that the gameplay is somewhat reliable at engaging and maintaining the participants' interest.

The enjoyment category contains these to questions (1) "I think the game was fun" (M = 5.2, SD = 0.79) and (2) "I felt bored while playing the game" (M = 3.1, SD = (0.88). The mean of the category is 4.15. However, it can be seen on figure

8.12 and the mean of the questions individually that participants generally thought the game was fun, as the mean is 5.2 while the median is 4, which is "Agree" and "Somewhat agree" respectively. In the question that asked if they felt bored, the tendency was to answer "Somewhat disagree" which is also the median, and translates to a score of 3.

The personal gratification category can be described as measuring sense of satisfaction that the participant experience, and it is derived from wanting to fulfil the goals. The questions in this category are (1) "I was focused on my performance while playing" (M = 6.1, SD = 1.20) and (2) "I wanted to do as well as possible during the game" (M = 6.0, SD = 1.49), the mean for the category is 6.05. This indicates a response tendency in the "Agree" category. This points towards the participants being invested enough in the game so that they wanted to do well, and cared about their results. Either in BCI interactions or in terms of normal gameplay.

Variable	Both versions
I lost myself in this experience	4.5 (1.43)
I was absorbed in the experience	3.2 (1.03)
I felt frustrated while playing the game	3.7 (0.67)
I found the game confusing to play	3.2 (1.48)
The game was aesthetically appealing	5 (1.25)
Game was attractive	5.1 (1.29)
Playing the game was worthwhile	4.8 (0.79)
I felt interested in the game	4.7 (0.82)
Found controls of the game straighforward	5.8 (1.23)
Did not care to check events that happen in the real world	5.1 (1.59)
Felt detached from outside world while playing	4.9 (1.10)
I think the game was fun	5.2 (0.79)
I felt bored while playing the game	3.1 (0.88)
I was focused on my performance while playing	6.1 (1.20)
I wanted to to as well as possible during the game	6.0 (1.49)

Figure 8.13: Mean results from overall gameplay

Chapter 9

Discussion

This chapter discusses the gathered results and what they mean for the problem formulation and possibly further research. It also covers general discussion about the project and evaluation as a whole.

9.1 Evaluation results

This section covers the results and what they mean for the game and the problem formulation.

9.1.1 Logged Data Result

The logged data regarding the MI events for the success rate and number of attempts and successes, was mostly meant to show if the participant did better with either of the versions of the game, however since the BCI cap that were made available to us did not give consistent data flow, and we decided to not use it, this goal was discarded.
The data, however, can still be used to see if the simulation tool we made for the realistic success rate and worked as it was meant to. Since the mean success rate, mean for interval and battery are 76.06% and 76.27% respectively it is confirmed that it worked as intended, because it is within our accepted range of 65% - 85%. As mentioned in section 6.6.2, the simulation tool is chance based, and the success rate ranges from 57% - 100% in both conditions. 100% however is unrealistic for BCI input and this is also considered when looking at the preference of the game versions.
The logged data for measuring repetitions per minute are used to help determine if the patient would get enough training from the game, so that it would even be useful in the future with further development and added features.

9.1.2 Version specific

Based on the previously reported data in section 8.4, the tendencies observed in the questionnaire results, indicate that participants reported higher levels of frustration for the interval version compared to the battery version. Additionally, engagement levels were lower for the interval version.
In terms of specific aspects, participants found the timing of events in the interval version to make more sense compared to the battery version. They also perceived the Interval version to feel more natural in the given context. It is worth noting that both versions were perceived to have logical triggers for the MI prompt.
However, it is important to exercise caution when drawing definitive conclusions due to the small sample size of 10 participants. The data may not be statistically significant. Nevertheless, within this limited sample, the tendencies suggest that the interval version of the game may have drawbacks in terms of frustration and engagement, but excels in areas such as timing, naturalness, and trigger logic.
To gain a more comprehensive understanding of the strengths and weaknesses of the interval and battery versions, further analysis with a larger and more relevant participant group would be beneficial. Additionally, it is important to note that the majority of the participants indicated a higher level of weekly game-time, so it would be safe to assume that these results reflect that of those already familiar with game logic and norms that this game utilizes.

Engagement Statistics

The comparable data that could help determine the engagement separately in the two versions showed no statistical significant difference (see section 8.1.1). Which means that when looking solely at engagement as a factor, there is no difference in the two versions. However, as mentioned above and in the interview answers (see section 8.2) there are other factors that might make a user prefer one over the other.

9.1.3 Metaphor preference

If we look at 8.9 we can see that the participants agreed slightly more with the interval mode's metaphor being natural. If we look at the interview results in 8.2 this is further supported, since multiple participants mentioned the fireball metaphor and said it was easier to imagine a fireball forming in your hand. It is also shown that 50% preferred the interval mode when they had to do motor imagery, this is one of the questions where some of them specifically mentioned the fireball metaphor being more natural than the charging metaphor, which 30% preferred.

9.1.4 Interview discussion

As mentioned in 8.2 when asked which version they thought they did the best in, the participant's answers were very evenly spread out. This might be due to the fact that we simulated the BCI, and that their successes and failures were random. If they had used a real BCI it might have looked different, and they might have favoured one version over the other. It was mentioned multiple times that the "fireball in hand" metaphor was more logical and made performing motor imagery easier, but again this might be because of the randomness of our simulation. Some participants mentioned the kind of stress in the interval mode as being both a pro and a con. The fact that some thought of it as a pro, can be due to the reason that many of the participants play video games on a regular basis, and therefore are used to the stress during video games, and therefore liked it. This could potentially be different with people who are not used to playing games.
In general most of the interview answers reflect the findings in the quantitative data, since they had positive and negative things about each version, therefore there were no clear preference across all the participants, except for the fireball metaphor being liked by many.

9.1.5 BCI simulation

As previously discussed in the implementation chapter, a Wizard of Oz implementation was required in order to conduct a realistic testing as possible in the given time frame. As cited in [13] the average success rate for a BCI participant on average falls between 65% and 85%. During the testing, the participant's success rate artificially set within this span at random. While this was the intent with the program, this should be verified. By comparing the average number of success and fails, an average success rate can be calculated. When calculated for each version of the program, both have an average success rate of roughly 76%. This can be seen in figure 8.1.

9.1.6 Overall questions

The overall questions indicated that most of the categories and questions resulted in a score above neutral in most of the positive questions. The only positive question to have a mean below 4 is "I was absorbed in the experience". The other questions that has a mean below 4 are the questions we want to have low scores. They are "I felt bored while playing the game", "I found the game confusion to play", and "I felt frustrated while playing the game".
Based on this, we argue that the game in general has an acceptable level of engagement score on average for the participants that have tried this. We believe that the game has potential and this is a very early stage of a full rehabilitation video game,

in section 9.4.1 a list of improvements are presented that are believed to be able to improve the engagement score further.

9.2 Evaluation method discussion

After verifying that artificial success rate lies within the average span. The actual procedure and tone of The Wizard of Oz test must be evaluated on its merits. This includes its ethical aspects, its validity/accuracy as a testing method, as well as how it could potentially have effected the results.
As part of the ethical considerations of conducting a Wizard of Oz test, the disclosure of the false nature of the test must be provided. Their reaction did in no case show any negative feedback to this. Following the disclosure, nearly all participants expressed a lack of previous suspicion. This can be validated through their belief/perception, however false, that they had control of the system at least 50% of the time if not higher. This fact was also expressed through interview questions pertaining to version preference, where multiple participants explained a perception/feeling of practice bettering their ability to perform motor imagery events.
Due to these responses, it is believed that the validity and accuracy of results relating to system perception and user agency can be deemed valid and provide similar insights despite the lack of functioning BCI hardware. This falsehood does however prevent quantitative objective measurements of what effect the utilized metaphor could potentially have on the user's ability to recall the specified motor imagery. This, for example, could have potentially shown that when using the battery metaphor, the applied stress could negatively affect performance. A trend that multiple participants perceived and expressed concern over when discussing the pros and cons during the concluding interview process.
In order to achieve the highest realism during testing, the participants were put through a pseudo calibration procedure. Similar in structure and content to that which would actually calibrate the BCI software. Different only in the number of calibration repetitions, normally 30 sets of think and rest prompts reduced to only 5. Done only as to not waste the participant's time when the calibration quality would not affect the following test. In order to further sell the functionality of the hardware the participants would be asked to bite down or blink (procedures which would normally be used to measure proper connectivity and signal quality), following the preliminary explanation that the hardware, being quite sensitive, could easily be effected by these movements. This should contribute to the reliability and validity of the test.
In terms of the procedures effect, the obvious direct result of The Wizard of Oz testing prevented the objective measurement of user performance during play. Beyond this the feeling of frustration and lack of correlation while however accurate to a real BCI setup could potentially reflect negatively on a given version as a

user's belief that they are correctly preforming motor imagery without feedback validating that the system correctly detected said thought, or vice versa, without the user attempting to preform motor imagery the system provides positive feedback stating that a thought was detected.
A thought about the preferred metaphor indicates that, the participants were quite often biased towards the trail, where they had the most successful MI events. This can give a hint towards an optimized version of the evaluation. As the BCI results are simulated, it is possible for us to manipulate it, as to fit the evaluation as optimal as possible. By having the two trails of gameplay, when trying both versions, have the same success rate, this bias will hopefully be eliminated. The timing and order of success and fails, will be different, for them not to get suspicious of the BCI being simulated. This could potentially show a more honest preference in metaphor and version.
In conclusion, it is believed that the data obtained during The Wizard of Oz testing to provide equal quality to that which would have been obtained through actual measurement. Despite its lack of objective measurements, which would otherwise have been obtainable and measurable, such as the success rate of the MI events.

9.3 Success Criteria

With the results in mind, we look at the success criteria and asses if they are fulfilled or not.

9.3.1 The user should on average have a likert scale above neutral in terms of engagement and fun, with the best pacing implemented.

There was no statistically significant difference in which metaphor the participants liked the most, therefore we look at both versions to assess these criteria. As seen on figure 8.4 when asked if the motor imagery broke their engagement, the mean for both versions scored below 4 (neutral), which in this case is preferred since the question is about broken engagement. Therefore, this success criterion is fulfilled.

9.3.2 Each BCI metaphor should present a distinct grouping/structure of BCI events

The game flow of both versions can be seen on figure 8.2, here it is shown how the motor imagery and gameplay events are spread out in both versions. From this figure we can conclude that the success criterion is fulfilled as interval is spread out as singular motor imagery events, with smaller gaps in between, and battery is in two groups consisting of multiple motor imagery events, with larger gaps in between.

9.3.3 The user should get at least 15 BCI repetitions during one game session of 20 min, excluding setup procedure

In 8.1.2 repetitions per minute is discussed. The mean for the battery version is 2.38 repetitions per minute and 2.54 for interval. This means that it would be about 47.6 and 50,8 for battery and interval respectively. Therefore, this success criterion is fulfilled as well.

9.4 Evaluation, future development and improvement

The evaluation in this report can and should be seen as a bigger pilot test, so that in the future, a similar evaluation structure might be used with a connected BCI. During the data processing and evaluation process, we found some points that could be improved for a bigger evaluation in the future.
The first point is to have more of the questions from the user engagement scale [2] and GUESS-18 [15] questionnaire to be asked in the version specific questionnaire in order to get more data regarding engagement in each version and not just in the game as a whole.
The second is that the evaluation would probably get more relevant data if it was tested on the target group, or someone closer to the target group in either age or disability.

9.4.1 Future game improvement

The game itself can also be improved in order to make it a more enjoyable experience for the players. Due to the time frame of the project, these did not get implemented, but are noted for possible future work.

- Mini-map will help the player to know where the enemies are on the map, that they have left to attack. This will make the game less about searching the map. The map is however currently not that big, for this to be a major issue.
- Additional feedback, specially in regard to the status of the MI prompt.
- Have enemies ignore the player's position, when finding a path to their location, to have them walk towards the player.

Chapter 10

Conclusion

The aim in this report was to compare two different versions of a BCI rehabilitation game for individuals with cerebral palsy, with both versions having a different pacing. The goal was to identify if there is any difference in engagement levels in the different versions, which hopefully would lead to more motivation and enjoyment in the patient's rehabilitation training.
Revisiting the final problem statement laid out in chapter 4.
"How does 'charging/battery'- and 'interval'-based pacing compare, in terms of the player's engagement, in a video game made for upper limb motor imagery training in children with cerebral palsy?"

The success criteria were established to help answer this question, in section 9.3 each of the criteria has been discussed in order to assess whether they are fulfilled or not. All the success criteria are deemed to be fulfilled, therefore based on this we argue that we can answer the problem statement with a high degree of certainty.
Based on the findings, there is no statistical significant difference to be found in engagement levels in the charging or interval based versions. Any difference that may have been measured are likely due to chance or other factors, such as they had more successful MI events in one over the other. Based on this, we conclude that each version is being equally engaging for a patient, and it is purely up to the individual's preferences. However, as mentioned in section 9.4 a more in depth engagement questionnaire could have been answered after each version in order to have more data to compare the two versions, would be desired.

Bibliography

[1] 2007. A report: the definition and classification of cerebral palsy April 2006. *Developmental Medicine & Child Neurology* 49, s109 (2007), 8–14. DOI:http://dx.doi.org/https://doi.org/10.1111/j.1469-8749.2007.tb12610.x

[2] 2018. A practical approach to measuring user engagement with the refined user engagement scale (UES) and new UES short form. *International journal of human-computer studies* 112 (2018), 28–39.

[3] 2023a. Dragon Escape. (2023). https://www.viribusvr.com/product/suit-system/

[4] 2023b. Rumble With Rumble. (2023). https://www.viribusvr.com/product/suit-system

[5] 2023. VR Stretching Games and Exercises for Children with Cerebral Palsy. (March 2023). https://www.cognihab.com/blog/vr-games-exercises-for-cerebral-palsy [Online; accessed 8. Mar. 2023].

[6] Adobe. 2023. Mixamo. (2023). Retrieved May 9th 2023 from https://www.mixamo.com/#/

[7] Alexandranbsp; Maria Bartas, Mike Long Heilskov Nguyen, Moaaznbsp; Mohamed Jamal Allahham, and Weronika Katarzyna Korbutt. 2019. 1–89 pages.

[8] Blender. 2023. Blender. (2023). Retrieved May 9th 2023 from https://www.blender.org/

[9] World Cerebral Palsy Day. 2020. WHAT IS CEREBRAL PALSY? (2020). Retrieved Feb 22, 2023 from https://worldcpday.org/portfolio-item/slide-decks/

[10] Ann-Christin Eliasson, Lena Krumlinde-Sundholm, Birgit Rösblad, Eva Beckung, Marianne Arner, Ann-Marie Öhrvall, and Peter Rosenbaum. 2006. The Manual Ability Classification System (MACS) for children with cerebral palsy:

scale development and evidence of validity and reliability. *Developmental Medicine & Child Neurology* 48, 7 (2006), 549–554. DOI:http://dx.doi.org/https://doi.org/10.1111/j.1469-8749.2006.tb01313.x

[11] Tracy Fullerton. 2019. *Game Deisgn Workshop, A playcentric approach to creating innovative games*. Taylor Francis Group, LLC.

[12] Australian Cerebral Palsy Register Group. 2023. Report of the Australian Cerebral Palsy Register, birth years 1995-2016. (2023). https://cpregister.com/wp-content/uploads/2023/01/2023-ACPR-Report.pdf

[13] Mads Jochumsen, Muhammad Shafique, Ali Hassan, and Imran Khan Niazi. 2018. Movement intention detection in adolescents with cerebral palsy from single-trial EEG. (2018).

[14] Oleh Kachmar, Anna Kushnir, Bohdana Fedchyshyn, Julián Cristiano, John O'Flaherty, Kjetil Helland, Gordon Johnson, and Domenec Puig. 2021. Personalized balance games for children with cerebral palsy: A pilot study. *Journal of pediatric rehabilitation medicine* 14, 2 (2021), 237–245.

[15] Joseph R. Keebler, William J. Shelstad, Dustin C. Smith, Barbara S. Chaparro, and Mikki H. Phan. 2020. Validation of the GUESS-18: A Short Version of the Game User Experience Satisfaction Scale (GUESS). *Journal of Usability Studies* 6 (2020), 49–62. Issue 1.

[16] Kenney. 2022. Kenneys Game Assets 1. (2022). Retrieved May 9th 2023 from https://kenney.itch.io/kenney-game-assets-1

[17] Kim Kristensen. 2021. Fysioterapi og ergoterapi til børn med cerebral parese. *Sundhed.dk* (2021).

[18] Daegyun Lee and Youngsook Bae. 2022. Interactive Videogame Improved Rehabilitation Motivation and Walking Speed in Chronic Stroke Patients: A Dual-Center Controlled Trial. *Games for health* 11, 4 (2022), 268–274.

[19] David Marshall, Damien Coyle, Shane Wilson, and Michael Callaghan. 2013. Games, Gameplay, and BCI: The State of the Art. *IEEE Transactions on Computational Intelligence and AI in Games* 5, 2 (2013), 82–99. DOI:http://dx.doi.org/10.1109/TCIAIG.2013.2263555

[20] Fabiana Paula Almeida Martins, Thaís Massetti, Tania Brusque Crocetta, Priscila Bianchi Lopes, Amanda Abreu da Silva, Eliketylen Fernandes Figueiredo, Luiz Carlos de Abreu, Talita Dias da Silva, and Carlos Bandeira de Mello Monteiro. 2019. Analysis of motor performance in individuals with cerebral palsy using a non-immersive virtual reality task – a

pilot study. *Neuropsychiatr. Dis. Treat.* 15 (Feb. 2019), 417–428. DOI:http://dx.doi.org/10.2147/NDT.S184510

[21] med material. 2023. HandStrengthener. https://github.com/med-material/HandStrengthener. (2023). Accessed: 2023-05-22.

[22] Omar Mubin, Fady Alnajjar, Nalini Jishtu, Belal Alsinglawi, and Abdullah Al Mahmud. 2019. Exoskeletons with virtual reality, augmented reality, and gamification for stroke patients' rehabilitation: Systematic review. *JMIR rehabilitation and assistive technologies* 21, 9 (2019), e12010–e12010.

[23] Robert Palisano, Peter Rosenbaum, Stephen Walter, Dianne Russell, Ellen Wood, and Barbara Galuppi. 1997. Development and reliability of a system to classify gross motor function in children with cerebral palsy. *Developmental Medicine & Child Neurology* 39, 4 (1997), 214–223. DOI:http://dx.doi.org/https://doi.org/10.1111/j.1469-8749.1997.tb07414.x

[24] David L. Poole and Alan K. Mackworth. 2017. *Artificial Intelligence: Foundations of Computational Agents* (2nd ed.). Cambridge University Press, USA.

[25] Johnmarshall Reeve and Woogul. Lee. 2014. Students' classroom engagement produces longitudinal changes in classroom motivation. *Journal of Educational Psychology* 106 (2014), 527–540.

[26] Neumann Sven. 2023. GIMP. (2023). https://www.gimp.org/

[27] LaTasha R. Swanson and David M. Whittinghill. 2015. Intrinsic or Extrinsic? Using Videogames to Motivate Stroke Survivors: A Systematic Review. *Games for health* 4, 3 (2015), 253–258.

[28] Katie S. Tekinbas and Eric Zimmerman. 2003. *Rules of Play* (1st ed.). The MIT Press, USA.

[29] Athanasios Vourvopoulos, Octavio Marin Pardo, Stéphanie Lefebvre, Meghan Neureither, David Saldana, Esther Jahng, and Sook-Lei Liew. 2019. Effects of a Brain-Computer Interface With Virtual Reality (VR) Neurofeedback: A Pilot Study in Chronic Stroke Patients. *Frontiers in human neuroscience* 13 (Jun 2019), 210. DOI:http://dx.doi.org/10.3389/fnhum.2019.00210