Boosting Chest Physiotherapy's Positive Effect with Gamification Technologies

Xiaoyan Zhou

MSc in Software Systems
The University of Bath
September 2021

This dissertation may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation.

Boosting Chest Physiotherapy's Positive Effect with Gamification Technologies

Submitted by: Xiaoyan Zhou

Copyright

Attention is drawn to the fact that copyright of this dissertation rests with its author. The Intellectual Property Rights of the products produced as part of the project belong to the author unless otherwise specified below, in accordance with the University of Bath's policy on intellectual property (see https://www.bath.ac.uk/publications/university-ordinances/attachments/Ordinances_1_October_2020.pdf).
This copy of the dissertation has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the dissertation and no information derived from it may be published without the prior written consent of the author.

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Bachelor of Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Abstract

With the development of the times, a series of congenital diseases have gradually attracted attention. Cystic fibrosis is one of such diseases which could cause breathing problems that threaten life. The patients would have to do chest physiotherapy/breathing exercise daily for clearing their airway. Such treatment processes are normally boring and exhausting so that low adherence are detected among the young patients, which has brought potential risks on the health of the group.

BreatheHero is a project that designed to help the children with lung diseases who require to do the breathing exercise daily for keeping their health with the gamification technology. By gamifying the therapy process, we aims to make the treatment more fun and easier so that the adherence can be improved. Meanwhile, we also want to inspire the young patients with the game and bring them positive effect. The project has been refactored during the research, which has brought the following optimizations: 1) more maintainable system and file structure. 2) more flexibility and instant-playability. Moreover, we produced our novelties in the research: 1) A special PEP input device integrates the game controller and sensors, which better combines the game and the breathing exercise. 2) The game's running platform, web platform is supported which provides more instant-playability. 3) Comparing to the previous researches, a more completed game-play loop is provided and aims to bring young patients longer interest and better game experience. 4) More industrial development life cycles are used for increasing robustness of the project.

During the research and development, we focused on the following research questions: RQ1) How to build game features that can motivate young patients to adhere their breathing exercises? RQ2) How to make the game more sustainable to provide such motivation? RQ3) Does the project actually help the young patients adhere their treatment? For answering RQ1 and RQ2, requirements and features were gathered and analyzed from the related works. And for answering RQ3, a long term field study (last for 7 days and 10 minutes treatment daily) was organised. Due to the influence of covid-19, young patients were not recruited. Alternatively, 5 healthy participants joined in and simulated the therapy and feedback were given. The system was evaluated from four aspects: the therapeutic effect of the system, the game features score ,the system usability (SUS) and the intrinsic motivation inventory (IMI). The therapeutic effect of the system evaluation gave the result that within a 10-minute gameplay daily which last a week, participants simulated on average (14.97 ± 4.76) times breath and (43.29 ± 13.72) times cough daily. The average breath length for each breath was (4.89 ± 1.02) seconds. The daily entertainment average score daily was ($3.57/5 \pm 0.74$). The result proved that, the system could provide happiness at the same time maintaining the breathing exercise. The feature evolution gave the result as the enjoyment level for the whole system as ($81.84/100 \pm 9.56$), SUS score of the whole system as ($63.2/100 \pm 7.29$), the enjoyment IMI score as ($69.6/100 \pm 9.21$) and pressure IMI score as ($52.61/100 \pm 19.04$). The result could give conclusion that 1) The overall entertainment score of the system is high, the system can provide enjoyable game experience. 2) The system is overall easy to use. 3) The system can provide internal positive motivation but the tension level is not enough. The results could help answer RQ3 and become strong evidence for supporting RQ1 and 2. The positive results for BreatheHero also shows that it has great potential for further development.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank my supervisor, Dr. Christof Lutteroth for providing me with guidance, direction and technical assistance in my project. I would like to thank all my study participants for sacrificing their spare time after work to participate in the study.

Chapter 1

Introduction

1.1 Background

1.1.1 Chest Physiotherapy

Chest physiotherapy or breathing exercises is an airway clearance technique for treating chronic lung diseases (CysticFibrosisFoundation, 2021). The patients with chronic lung diseases would need to do such exercises daily to prevent the worsening of lung situation (BritishLungFoundation, 2021). However, since such exercise is boring and hard, poor adherence is detected especially in among the young patients. For improving the motivation of such treatments, the project introduces a gamified breathing exercises model called BreatheHero.

1.1.2 Gamified Chest Physiotherapy (GCP)

Gamification has been considered as an effective and persuasive incentive mechanism (Blohm and Leimeister, 2013). Such technology has been widely used in the areas of education, sports etc. for motivating people and providing gameful experiences. A research by Hamari, Koivisto and Sarsa (2014) demonstrated the positive impact of gamification on psychological/behavioral outcomes.

GCP could encourage the young patients to spend more time on the exercises than the normal chest physiotherapy (Bingham, Lahiri and Ashikaga, 2012). It is obvious that this technology improves children's compliance with breathing exercises and the data collected during playing the game could be used for subsequent treatment. There has been some previous works on gamification of breathing exercises such as PEP Hero and Globule (Balli, 2018) etc. which are video games with GCP features.

BreatheHero is also a GCP featured game. It takes a respirator game-pad as input to provide game experience during breathing exercises. In the game, the breathing signal will be converted into the game character's actions in the game.

1.1.3 The Limitation of GCP

There are limitations for such technology. Firstly is that children's interest in games is time-sensitive. Eventually they would lose interest in the immutable game, and the game would

lose the effect of strengthening adherence. This would cause conflicts with development since developing such games is expensive and time-consuming.

And as the number of development iterations increases, certain development methods that are separated from users will eventually make the above problems more likely to occur. And this refers us to think about what game features the children really want and how to design the game features to fit as a "breathing" game.

1.1.4 User-based Development Lifecycles

For any software development missions, there are some development models to follow which aim to improve development requirements and improve development efficiency (Davis, Bersoff and Comer, 1988). Some development models are based on one-way requests from customers such as waterfall model (Petersen, Wohlin and Baca, 2009). Such models have the potential risks that cause the final product to deviate from user needs (Balaji and Murugaiyan, 2012), which is the problem mentioned above . There are some other development models based on regular communication with customers such as Agile (Cohen, Lindvall and Costa, 2004). Customers and developers are going to have regular meetings for feedback to improve the products.

Since the task is also software development and the level of fun of the game content and the continued attraction to the young patients are mattered, we aim to return the choice of the game content to them. Inspired by the user-based development lifecycles, the young patients would be invited as the role of customers and continuously put forward new demands and feedback on the game development.

1.1.5 Overview on Gamification & Game Design

A common state in software development is that customers usually have no development knowledge background, which creates a gap between developers and customers (Franken et al., 2015). And at the same time, there is restriction by specific game rules, a better strategy is for developer to collect optional development features for the game based on previous researches and then communicate with the young patients to evaluate the designs. Such processes require the developer has the basic knowledge of game design.

Game design in this project cannot leave the gimification principles (Zichermann and Cunningham, 2011). The game-design elements in the process aim to provide extrinsic motivations to the non-gaming activity (Zichermann and Cunningham, 2011) and in this case is the breathing exercise. The game-design elements can be refined from many game design models in previous researches such as the compulsion loop in gamification (Zichermann and Cunningham, 2011), branching storylines (Rollings, Andrew and Adams, 2006) and boss battles (Lee and Tyler, 2015; Thompson and Clive, 2004) etc.

1.2 Research Questions

The project aims to solve the following research questions (RQ). RQ1: How to build game features that can motivate young patients to adhere their breathing exercises? RQ2: How to make the game more sustainable to provide such motivation? RQ3: Does the project actually

help the young patients adhere their treatment? For all the questions, they would all require a series of evaluation mechanisms to verify the effectiveness.

- **RQ1: How to build game features that can motivate young patients to adhere their breathing exercises?**

  The research question is about human-computer interaction and game design and aims to improve the adherence of chest physiotherapy for young patients directly. The chest physiotherapy requires patients' will and self-management ability but this is normally very difficult for children (Taddeo, Egedy and Frappier, 2008). For improving the adherence for young patients, extra efforts for making the treatments more enjoyable and motivating are required. There has been several studies focusing on such area such as the Bubble PEP (WestSuffolkNHSFoundationTrust, 2019). The project BreatheHero uses the similar idea and combines it with video games. The game controller is designed into a PEP device with sensor system. In this case, RQ1 would mainly focus on **(1) How to make the game features fit such control system? (2) How to make the game features more fun?** For answering the question 1, user researches would be considered for collecting ideal features and user trial surveys would be considered for collecting user feed-backs and answering question 2. The two processes will be cycled to maximize the optimization of the game.

- **RQ2: How to make the game more sustainable to provide such motivation?**

  A big challenge for the project is that how to provide such motivation for the children as long as possible. The effect of repeated level design of this issue is unsatisfactory. Young patients are likely to get tired of the game soon. For solving the problem, a story line could be necessary. As making the player as one part of the game story, it would make the game appeal to player for longer (Thompson, 2021). In game design area, storytelling systems are usually used in role-playing games and there are many story telling models can be referenced such as linear, nonlinear or interactive models (Rollings, Andrew and Adams, 2006). For all the models, they usually have one main line and several branches. The main story line would assist the player push forward the level play and branches would courage the player to explore the game world more. At the same time, a story telling system normally appears with its morality system. A story line with a good morality system could have positive impacts on players in a longer term(Casas-Roma et al., 2019), which is exactly needed for the project. By applying such models, I aim to create a relatively complete game world view and story lines for providing players longer term interests. And at the same time, the accompanying morality system could make the patients better understand their diseases and have a more positive attitude to face it. The final effect of this design will require user research to prove.

  Another aspect to address this problem is the compulsion loop in gamification (Zichermann and Cunningham, 2011) mentioned above. Such loop creates challenges for the player and once the player completes the challenges, he/she would be rewarded. The principle of this model is that in this process, players continue to gain a sense of pleasure and the sense of pleasure generates a desire to play (Zichermann and Cunningham, 2011).

  Boss battles also have a long history in game design area to make the game more attractive (Lee and Tyler, 2015). Minotti (2017) has compared the boss battles in

game *Cuphead* (a game designed as a collection of boss fights) as "uniquely beautiful and worthwhile challenges". Various bosses cooperate with different boss battle music can enhance the players' experience. And appropriate boss difficulty and failure will increase the player's ultimate sense of accomplishment when they finally overcome the challenges (Minotti, 2017; Rollings, Andrew and Adams, 2006). The boss battle could be an important part between the compulsion loop (Zichermann and Cunningham, 2011) and reward system even for all the game elements in the entire game.

There are also various of game design elements such as achievements, collection system and shop system etc (Zichermann and Cunningham, 2011). that can help provide players long term interest to address the problem. For the project breathehero, I aim to develop a game play circle as complete as possible and provide more lasting interest through the interaction between game elements.

- **RQ3: Does the project actually help the young patients adhere their treatment?**

  After the implementation of the features, a field study would be required for testing the final effect. Pariticipants would have the time to experience the game and give feedback and suggesstions for further development. The survey results collected would help to answer the RQ3. During the field study, the following hypothesis would be made and tested:

  Hypothesis 1: The proposed videogame can provide the user long term interests and positive effects with its completed play circle.

  Hypothesis 2: The proposed videogame can help the player make the breath exercise easier.

  Hypothesis 3: The proposed videogame is easy to play so that it has the universality for patients in all ages.

  For each hypothesis, there would also be several sub-hypothesises such as the story telling system has helped most for bringing players positive effect etc. to locate the positive game designs for providing experience for further development.

1.3 Novelty

The novelty of the project BreatheHero could be concluded into four aspects: game control features, game's flexibility, game content and development lifecycles.

In the game control features, BreatheHero's PEP integrated device contains somatosensory sensor and sound sensor. The somatosensory sensor allows user to control the character's movement by tilting the controller, which removes the need of the keyboard and mouse during the game play, the young patients would be able to focus on their breathing exercise. The sound sensor allows to monitoring the user's cough during the game play, which is inevitable during the breathing exercise. When coughing is detected, the game would be paused automatically. This makes the treatment process much smoother. Besides, the controller device was also designed to be very portable (see Figure 1.1). It can be easily connected to most devices with USB 3.0, which makes the patients possible to do the breathing exercise anytime. All the novel features make the game control more in line with the concept of human-computer interaction,

more convenient to use and enhance the therapeutic effect.

Figure 1.1: The PEP Device Designed for BreatheHero

For the game's flexibility, the project has been refactored with Godot game engine. Compared to the original project with Unity, Godot engine allows easily to export the project to web version and push the project online. Assistance with the portable controller design, we aim to explore the possibilities of home therapy. Relative to other GCP projects such as Globule and PEP Hero (Balli, 2018) which require more fixed treatment equipments at hospitals, BreatheHero has made more explorations on the flexibility and convenience of treatment.

In the game content, compared to the other researches mentioned above, BreatheHero has a more completed game play cycle. The cycle is based on the compulsion loop (Zichermann and Cunningham, 2011) as core model and extends its lifecycle with reward system, storylines and morality system and boss battles. By implementing such model, BreatheHero not only want to support the patient in treatment, but also hope to have a long-term positive impact on the patient's psychology.

In the development life cycles, the young patients would be treated as the participants of the project and directly participate in development (with surveys and emails). This aims to return the choice of game features to the young patients and improve the efficiency of development. By joining in the development, we also hope they can have a better understanding of the project and a more sense of accomplishment after the development is completed. Participation in the study would be completely voluntary. We will respect the privacy of all participants. Participants have the rights to quit the study anytime without any explanations.

All the novelties aim to make the game design closer to the young patient group, increase the game's fun and enhance the therapeutic effect of the game.

1.4 Summary of Results

The BreatheHero is defined as a GCP game and in the project, it has been refactored to adapt to the new research environment. The features for the game has been collected by analysing previous works and gathering ideas from the public. By applying agile development

life cycle, the prototype with all required components was developed in time. Due to the effect of covid-19, only healthy participants involved in the field study instead of young patients with lung diseases. However, by simulating the breathing exercise, valuable test results and feedback were collected. The field study has showed that at the current stage, BreatheHero has good potential for enhancing the players' motivation for breathing exercise and for further development.

1.5 Dissertation Structure

The dissertation would be organised as following: Chapter 2 would discuss the related researches. The knowledge on cystic fibrosis, chest physiotherapy and breathing exercise would be introduced. Gamification principles and previous GCP instances would also be discussed. Starting from Chapter 3, the content would about the design and development of the system. Chapter 3 would gathering and introducing our requirements by analysing previous researches and Chapter 4 would introduce the design of our prototypes. Chapter 5 is the implementing chapter and the development process would be organised as Sprints, which can provide a clear trace of the development process. Chapter 6 would introduce the mechanism for field study and followed by Chapter 7, the results evaluation. In Chapter 8 and 9, the outcome of the project will be discussed and summarized.

Chapter 2

Literature and Technology Survey

There are previous related works for the project in the area of chest physiotherapy, gamification, development lifecycles and video-games for chest physiotherapy. In the section, we would start with the introduction of cystic fibrosis and chest physiotherapy and followed with GCP instances. This is for us to have a better understanding of breathing exercises and generate our game features from previous works. Subsequently, we would highlight and analysis the gamification principles. Assistance with game features generated from previous works and the gamification principles, we attempted to resolve the RQ1 and RQ2 from a comprehensive, well-founded and instructive perspective. Finally, we would introduce some development life cycle models, which would provide guidance for our development and design plans.

2.1 Cystic Fibrosis, Chest Physiotherapy & Breathing Exercise

For gamificating the breathing exercise and applying suitable game design of GCP games, a better understanding of the mechanism of chest physiotherapy and breathing exercise is required.

2.1.1 Cystic Fibrosis

In various lung-related diseases, this research field usually focuses on the gamification treatment of Cystic Fibrosis (CF). The genetic disorder feature (Knowles and Durie, 2002) gives CF the opportunity to accompany human society to this day, and threaten human health worldwide, even for babies and adolescents. CF is caused by the two loss-of-function mutations in the CFTR protein (Knowles and Durie, 2002). Normally, our lungs' epithelial cells would produce a layer of slippery mucus, which guarantees the expectoration of airways' bacteria. This physiological mechanism is to ensure the cleanliness of the respiratory tract to avoid inflammation etc. The main pathological manifestation of CF is that the mucus becomes thicker and more viscous, the mucus and bacteria would accumulate in the airway. Such accumulation would lead to the obstruction, infection and inflammation of the airway (Balli, 2018).

The disease has estimated affected between 70,000 and 100,000 people worldwide and although there are ongoing researches focusing on treating CF, there is no cure for CF right now. The

symptoms of CF will accompany the patient for life (Worldwide and CysticFibrosis, 2018). Since CF is a genetic disease, it is usually diagnosed in youth. In UK, 1 in every 2500 babies born with CF (CysticFibrosisTrust, 2021) and over 4,000 children is affected by CF (Research, 27-9-2016). For the children born with CF, rigorous daily treatment would be with them for life to keep them healthy and alive (Worldwide and CysticFibrosis, 2018).

According to the pathological causes of CF, preventing the deterioration of lung disease has become a vital part of the treatment (Eakin et al., 2011). Airway clearance therapy (Volsko, 2013) has been proved as an effective way to alleviate the deterioration of respiratory complications and treat CF. Specific methods include positive expiratory pressure (PEP) (Elkins, Jones and van der Schans, 2006) etc.

2.1.2 Chest Physiotherapy

The direct cause of CF mortality and deterioration is that the pulmonary secretions (mucus, bacteria etc.) that should be discharged naturally with cough accumulate in the lungs, and excessive mucus accumulation can cause lung infection and inflammation. Repeated infections and inflammations would eventually lead to more serious lung complications (hemoptysis, pneumothorax etc.). If there symptoms continue to worsen, they may lead to death (Flume et al., 2010). As mentioned, chest physiotherapy is widely used for people with CF to clear mucus from the airways to prevent the occurrence of complications of respiratory diseases and the effectiveness of its treatment has also been proven in various related researches (Hofmeyr, Webber and Hodson, 1986; van der Schans, Prasad and Main, 2000).

Some chest physiotherapy technologies such as PEP have a very simple but effective principle. The intervention of a PEP technology is that the PEP device (see Figure 2.1) incorporates forced expiratory manoeuvres to generate higher pressures behind mucus and lead to coughing, which aims to help the patients cough out the mucus (Elkins, Jones and van der Schans, 2006). Although, there is no evidence proves that PEP has better therapeutic effects than other chest physiotherapy methods (such as postural drainage with percussion, active cycle of breathing techniques etc.), its therapeutic effects is still worthy of recognition (Elkins, Jones and van der Schans, 2006; McIlwaine, Button and Nevitt, 2019).

Figure 2.1: The PEP Device by Acapella Company
Cite from https://www.physio-pedia.com/File:Acapella.jpg

2.1.3 Breathing Exercise

The breathing exercise is the most direct pratical operation of PEP treatment, which is also the aspects where we need to know the details most for gamification such process. A breathing exercise with PEP is a forced breathing process and the patients would usually need to do

it several times. With the help of such process and the pressure provided by the device, the mucus in the lung can be pushed to the mouth of the airway and finally excreted with cough. A research by Darbee et al. (Darbee et al., 2004) showed that patients receive such training continously and correctly can significantly improve lung volumes and expiratory flow in the treatment cycle (10 - 12 months) to relieve and prevent lung complications.

A PEP breathing exercise cycle can be completed by repeating the following instructions (Physiopedia, 2021):

- Sit up straightly, slightly up the chin and put elbows on a table if necessary. The pose would help you open your throat and breathe smoothly.
- Take a deep breath and hold it for 2 to 3 second. While holding the breath, put the mouthpiece in the mouth or cover the mask over the mouth and nose.
- Breathe out for 4 to 6 seconds steadily (or as long as you could) and try not to cough yet.
- Repeat the steps above and then take a deep breath in through nose and cough the mucus out.
- Remove the mouthpiece or the mask. Do the coughs 2 to 3 times and take a deep breath. Use the stomach muscles to breathe 3 times quickly and forcefully. And then cough hardly to expel the mucus. Do not swallow the mucus.
- Have a rest for 1 to 2 minutes if necessary and such cycle would last 10-20 minutes until all the sputum is cleared.

The gamification would happen during the steps above. The game design would need to fit the process and fully consider human-computer interaction issues. The PEP device's grip and sudden conditions such as cough have been taken into consideration. The somatosensory design is used to help patients better maintain the posture during training to achieve better entertainment treatment effects and the sound sensors help patients pause the game when they cough. More features and improvements would be considered based on the instructions.

The intensity of PEP breathing exercises depends on the needs of the patient. If the disease becomes worse and more mucus is produced, higher intensity PEP breathing training is required. The intensity of PEP breathing training will be determined by the duration and frequency of the exercise. Gamification can make higher-intensity PEP breathing exercises easier, so as to help patients entertain the treatment.

2.2 GCP's Instance and Applicability and Effectiveness

The section would introduce the related work on GCP and its instance and applicability. GCP is developed for improving the adherence of chest physiotherapy especially in the group of young patients. We would start by analysing the current status of young patients' adherence of chest physiotherapy.

2.2.1 Adherence to Chest Physiotherapy and Improvements

Low treatment adherence for chronic conditions has been well recognised in the society and a study has shown that, approximately 50% of chronic patients have low adherence to their treatment (Myers, 2009). This phenomenon is especially obvious in the chest physiotherapy. People with CF would require to do the chest physiotherapy daily, however a lot of studies have shown a low adherence to CP (Myers and Horn, 2006). Ensuring adherence would require a high degree of understanding and recognition of chest physiotherapy and a high sense of self-management. Due to insufficient social experience of children, the problem of low adherence is particularly serious among young patients (Williams et al., 2007). The immediate reasons for the children's low adherence to treatment include feeling restrictive and unfair, boring, low enjoyment, social stigma and complex treatment program etc. (Prasad and Cerny, 2002; Williams et al., 2007)

For improvement of the low adherence for young patients, the following aspects could be considered. Firstly, not only the patients but their families should be made aware of the treatment's positive effect; secondly, social support and healthcare support are always necessary; thirdly, increase the variety and interest of treatment activities would help children accept their treatment easier (Prasad and Cerny, 2002). Aimed at the boring and lack of enjoyment of the treatment, gamificate the process could be a solution. In such solutions, video games have been combined with the breathing exercise. Gamification principles and game elements are applied to the treatment. There some previous works provide demo or products on such video games which would be analysed in the following session.

2.2.2 GCP and Its Instances

Gamification means applying game design and game elements in non-game context and GCP is the product for such principle. Compare to the other methods aiming to improve adherence of breathing exercise, GCP can provide children with longer interest. Related works has shown the potential of GCP and in this session we would discuss the game design and effect with GCP instances in previous researches.

2.2.2.1 Globule, Ange-Gardien, PEP Hero, Heritage, Bloïd and Les Aventures du Briand in Balli's Research

Balli (2018) evaluated six GCP videogames in his research. The project started with two game design students and the Sainte-Justine Hospital's CF pediatric team (Montreal, Canada) and used a participatory design approach. The prototype of the games were developed during the project. Ten participants (children aged 7-11 with CF) were invited to play 2 or 3 games in the research. Before the game, the children would answer a questionnaire on their play preferences and design preferences and after the game, they would answer another questionnaire to evaluate the games. All games used the same hardware which is an input breathing device connected to the computer (see Figure 2.2). The games' gameplay mechanism is through the player's continuous breathing exercises (exhale and inhale) with the input device.

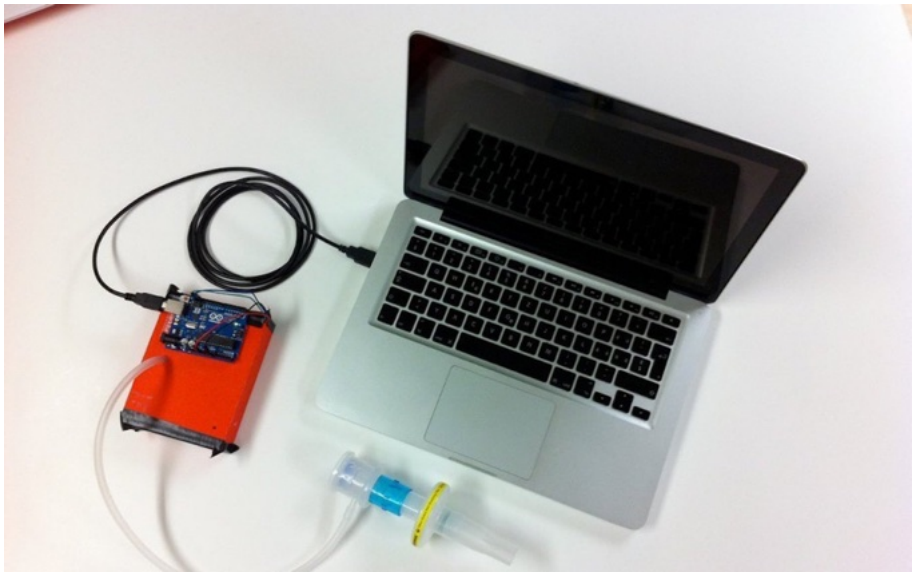


Figure 2.2: The Input Device in Balli's Research (Balli, 2018)

2.2.2.2 Globule

Globule (Balli, 2018) (see Figure 2.3) is a puzzle game with GCP features, however, the breathing exercise and gameplay are divided into two parts. For each game level, the game has the following 3 phases: 1) children would need to do the breathing exercise with the therapeutic pattern; the children would get game tips (an arrow pointing north, south, east or west) every-time finishing a correct and successful breath; 2) children have a break to cough out the mucus; 3) The children finish the puzzle with the arrows that lead them to the goal. Multiple game scenes are provided to children, they can choose from: the desert, the jungle and futuristic universe to start their games.

Figure 2.3: The Game Globule: The Breathing Exercise to Collect Arrows (Left) and Puzzles (Right) (Balli, 2018)

Eight out of ten children gave the positive feedback. Generally speaking, the game feedback is fun and easy to learn. The principle of the game is a simple reward mechanism. Players complete the puzzle and get rewards. Because the completed puzzle can be shared among

players, this increases the social nature of the game. Overall, this game has positively affected the children. However, the fatal flaw of this game is that breathing exercises and game mechanics are separated, which adds more time requirements to the original treatment. This is a huge burden for the long daily treatment.

2.2.2.3 Ange-Gardien

Ange-Gardien (Balli, 2018) (see Figure 2.4) is a strategy tower defense game, the game combines breathing exercises with games. Children would need to build towers to defend the attack from the devils and protect a little guardian angel. When the demon attacks, the children activate the tower and perform breathing exercises to help the tower attack. And when the breathing exercise is done, the attack would stop for children to do the expectorations. The children can build new towers with the points they gained in each attacks. The visuals of the game has been changed due to the development ideas in the figure but the mechanism is reused.

Because of the delay of the development, only two participants tested the game. Positive feedback were received, the game is easy to play and fun. The children feel much easier to do the exercise when playing the game. In the long run, the game mechanism has a high development prospect. The game itself is adapted to breathing exercises, and functions such as multi-scene and multi-player can be added to the game to improve gameplay.

Figure 2.4: The Game Amge-Gardoem: The Towers (Blue Jelly) and The Devils (Green Jelly) (Balli, 2018)

2.2.2.4 PEP Hero

PEP Hero (Balli, 2018) (see Figure 2.5) is designed as a side-scrolling space shooter game with GCP features. In the game, the children control a space ship to collecting items and breathe would help to make the spacecraft take off or down. After a cycle of play, the children would need to cough out the mucus and start the cycle again. The items are aligned to adopt a regular breathing pace in the test.

Positive feedback are received and the advantage of the game is that the simple game mechanism fits the rhythm of breathing exercises and provides guidance for children's breathing

exercises. However, long term interest could be difficult to guarantee since the very easy game mechanisms.

Figure 2.5: The Game PEP Hero: The Spaceship and Items (Balli, 2018)

2.2.2.5 Heritage

Heritage (Balli, 2018) (see Figure 2.6) is a side-scrolling game with PEP device. The game is more about integrating cultural and educational meaning into the game. Children of different cultural backgrounds will come into contact with different composites of their culture to find goals, protect goals and story flow. For example, for Greece, "the character has to catch a thief who has seized Poseidon's trident-fork. If he recovers it, the character must then find who stole Aphrodite's potion, Zeus' lightning, Gaia's seed bag, Hermes' shoes, and Athena's rouet to collect more points. In the end, the God of the Seas will then prepare the meal for the party with his trident-fork; Aphrodite will take care of the atmosphere; etc." (Balli, 2018)

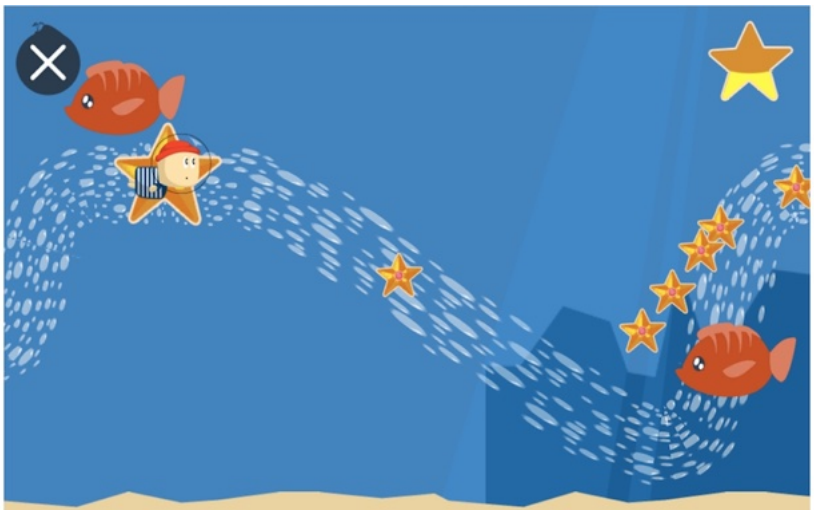


Figure 2.6: The Game Heritage with Ocean Background(Balli, 2018)

2.2.2.6 Bloïd

Bloïd (Balli, 2018) (see Figure 2.7) is a side-scrolling space shooter game with PEP device. The game idea is similar to the PEP Hero. The game aims to allow children to smash and avoid meteorites through breathing exercises to reach the mysterious planet. The number of meteorites increases with the playing time to encourage children to practice more intensive breathing exercise. The mechanics of the game are more complicated than PEP, and the difficulty will increase over time. This kind of game mechanics has great potential.

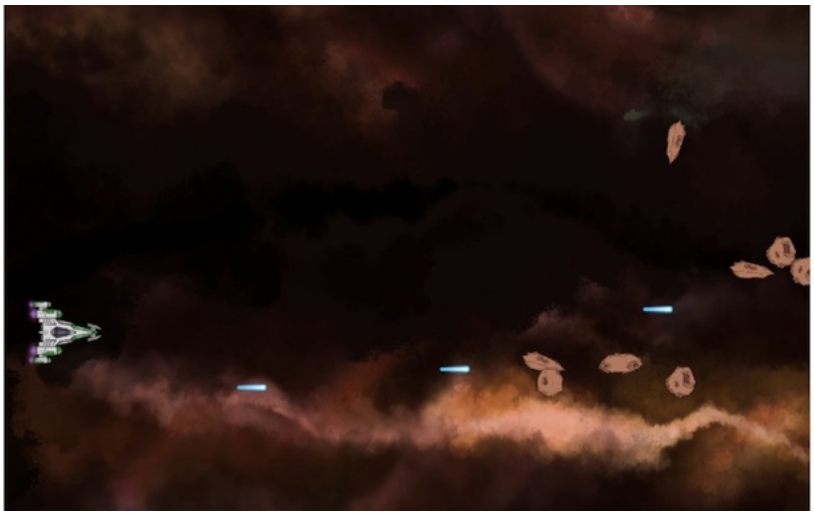


Figure 2.7: The Game Bloïd: The Spaceship and Meteorites (Balli, 2018)

2.2.2.7 Les Aventures du Briand

Les Aventures du Briand (Balli, 2018) (see Figure 2.8) is a adventure game. Player in the game is the captain of the ship. In the game, the crew will face different challenges. Players need to complete the challenges provided by the crew one by one. In other words, the game is composed of many mini-games. This increases the fun and choice of the game, and the development and research of the game continues.

Figure 2.8: The Game Les Aventures du Briand: Each Crew Means a Mini-game (Balli, 2018)

2.2.2.8 Conclusion

Compare and summarize the above several game examples, some game mechanics are worth learning and some design flaws can be circumvented. In the game Globule, the game mechanics and breathing exercises are forced to be divided into two parts without being directly connected, which leads to a longer daily treatment time. In our project, such problems should be avoided and the game mechanics and game methods should be well combined. In the game PEP Hero, items are used to assist the rhythm of children's breathing exercises. And in the game Bloïd, the difficulty of stratification and gradual increase has been affirmed. These designs could be also applied in our project. In almost all games, the problem of lack of choice and content of game elements has been mentioned. This is also one of the fundamental reasons why GCP games cannot provide children with long-term interest. This problem needs to be extra rigorously considered and emphasizes the importance of a complete game play cycle.

2.2.3 GCP's Applicability and Effectiveness

We would need to discuss the applicability and effectiveness for GCP before we apply it in the treatment. At this stage, the application of GCP lacks clinical practice data, which leads to a series of potential risks such as hyperventilation of patients or disrupted breathing rhythm, etc. At the same time, we also need to think about if GCP are really effective in improving the adherence in the treatment of the children.

2.2.3.1 Applicability

A research by Bingham et al. (2010) discussed the safety and applicability problem of GCP. The research focused on two aspects: 1) The GCP video games should not lead the player to hyperventilate or hypo-ventilate. 2) If the player would have enough skills to play such games since the breathing exercise with video games would require eye-breath coordinations. The experiment was designed as following: they designed a "sine wave" game which shows a "sine wave" floating between the upper and lower borders, and the player needs to control and keeps the "sine wave" from touching the upper and lower borders through breathing exercises. The "sine wave" will continue to advance in the game so that the player's score will continue to rise. The "sine wave" is designed to simulate the breathing rate. Players need to maintain the right breathing exercise intensity if they want to get a high score. A digital spirometer is incorporated into the breathing control device to record the player's breathing intensity to identify if they hyperventilated or hypo-ventilated.

Ten patients (age between 7 to 17) were invited to the research, they played 15 minutes of the game for 6 round. After the game, the trial conducted an interview about the game, positive feedback were received and the research had the following conclusions: 1) GCP shows its safety and recreational value, the system also increased patients' breath awareness. 2) experiments have proved that patients can exercise the ability of eye-breath coordinations while playing; 3) games are effective in enhancing treatment compliance. The competitive elements even strengthen patients' social communication, making them more willing to cooperate with treatment. Although the experimental base of this experiment is small, it proves to a certain extent the universal applicability of GCP.

2.2.3.2 Effectiveness

Another research by Bingham, Lahiri and Ashikaga (2012) the effectiveness of GCP. The research discussed the following two questions: 1) If GCP can increases participants' adherence of the treatment? 2) If GCP can actually improve pulmonary fucntion tests? The research was designed as following: two GCP video games are used in the experiment and 13 young participants were involved. The trial used a controlled trial method. The patients were divided into two groups, one group was treated with game, and the other group had only control device. The treatment trial lasted 2-3 weeks.

The research gave the following results: 1) Subjects used games and control devices had similar breathing exercise time. However, the games group had a trend toward more minutes. 2) Two subjects both have positive impacts on the guidance of HEFs (high flow events, records of respiratory conditions). 3) The game subjects had a trend to increased vital capacity. This test proves that GCP is as effective as ordinary Chest Physiotherapy in practical applications and has more potential in adjuvant therapy.

2.3 Gamification and Game Design

The section would highlight the gamification and game design principles. BreatheHero project would based on the gamification loop (Zichermann and Cunningham, 2011) and extends the contents with various game elements.

2.3.1 Gamification

Gamification is a technology for influence user behavior by activating users' individual motivations with game-design elements (Petkov et al., 2011). And Zicbermann and Cunningbam (Zichermann and Cunningham, 2011) also mentioned gimification as "The process of game-thinking and game mechanics to engage users and solve problems.". The definition helped us define the scope of gamification and its role in the study, that is, using play thinking to help children solve the problem of low treatment compliance. This process will also involve converting non-game scenes and non-game actions into games (Zichermann and Cunningham, 2011). To achieve this goal, we can start by understanding the basic principles of this type of transformation.

The first thing we would need to consider is the fun quotient (Zichermann and Cunningham, 2011). In our case is the fun quotient of the breathing exercise. The breathing exercise itself is painful and boring and you may consider how it is possible for it to be fun? Zicbermann and Cunningbam (Zichermann and Cunningham, 2011) have answered the question with an example, a game called Flight Control for players to perform the job air traffic control, which is one of the most stressful jobs in our society. Despite the pressure of this work, the game is still well received. This also leads us to the conclusion: "It is the mechaniscs of a game - not the theme - that make it fun". Breathing practice has his unique mechanism, namely the process of exhalation and inhalation. It has great potential to successfully integrate this process into game elements. At the same time, it is also one of the core ideas of breathing exercises in the game.

After determining the starting point of the game, what we need to consider is the overall mechanism of the game. Zichermann and Cunningham (2011) also mentioned a compulsion

loop mode for gamification. For more specific, is the Challenge-Achievement-Reward Loop (see Figure 2.9). In this cycle, we issue challenges to players. Players can obtain corresponding achievements and rewards after overcoming the challenges. While receiving rewards, players feel pleasure and this pleasure enhances the player's interest in playing and pushes the player into the next cycle. This cycle is also effective in breathing exercises. At the same time, as mentioned by Balli (2018), we need to add punishment and monitoring mechanisms in the game to ensure that players complete breathing exercises of the correct intensity.

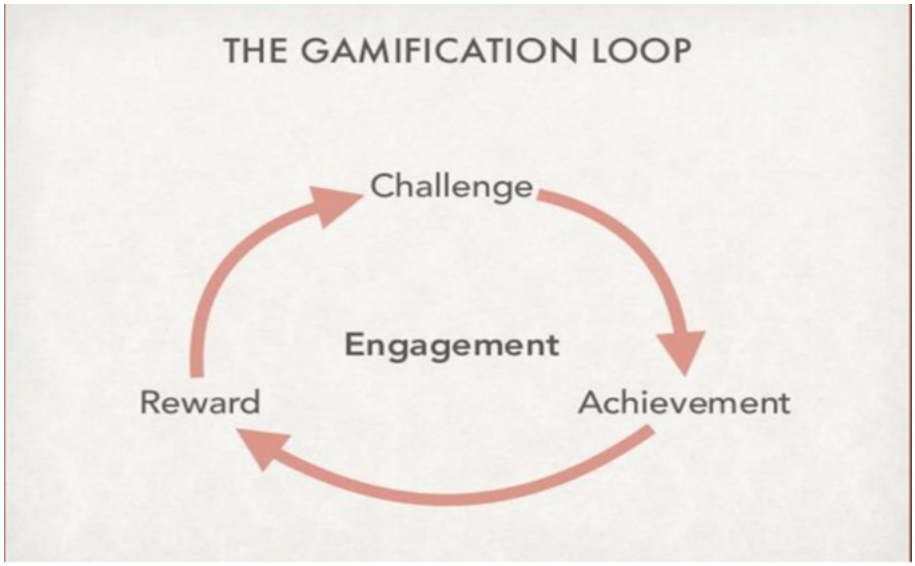


Figure 2.9: The Challenge-Achievement-Reward Loop Mentioned by Zichermann and Cunningham (2011); TO (2020)

After establishing the overall mechanism of the game, we can add game elements to it to complete the game design. The common game elements that meet the Challenge-Achievement-Reward loop include game levels, points system, stores, leaderboards, achievement systems and story lines, etc. (Zichermann and Cunningham, 2011), which will be selectively implemented later through questionnaires and patient participation.

2.3.2 Game Elements

2.3.2.1 Onboarding

Onboarding has been an important process for any new experience same as in the game design area. Zichermann and Cunningham (2011) have defined onboarding as the necessary step for increasing system's complexity, avoiding early fails and letting players get to know the system. A onboarding phase should provide players good senses of mastery, purpose and immersion (ZHANG and YIBIN, 2021). In ZHANG and YIBIN (2021)'s onboarding phase design guidance, by analysing various successful video games' onboarding phases, a onboarding pahse model with critical onboarding factors was generated (see Figure 2.10). As the model demonstrates, three main senses: mastery, purpose and immersion would need to be delivered in this phase and narrative would narrative will play a supporting role. More straightforward, the following factors are required for our design: tutorial, motivation and enjoyment. For achieving that, a short tutorial but which contains most of the game elements should be designed. This is for players to get a basic understanding of the game mechanism and at the same time, experience

the full play circle as much as possible to get more enjoyment in this phase. Elements such as rewards, achievements and tips would need to be considered to help players to get through the stage.

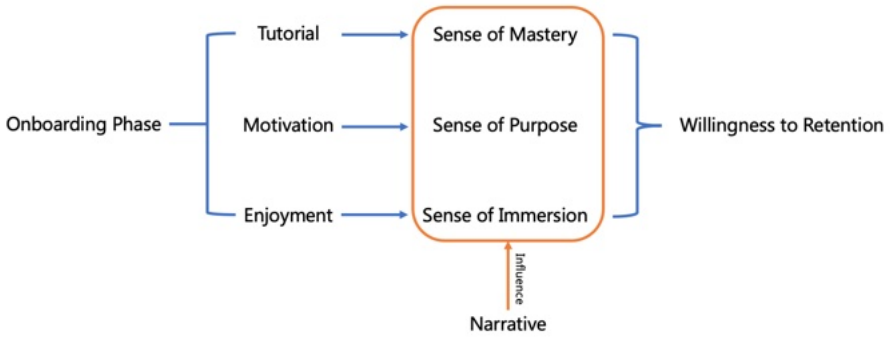


Figure 2.10: The Onboarding Phase Model with Onboarding Factors by ZHANG and YIBIN (2021)

2.3.2.2 Levels

Level is a hierarchical structure that accumulates players' game experience in their play progress. Such hierarchy structure is inspired by "A Novel Representation for Rhythmic Structure" which is a paper on complex rhythmic patterns models (Iyer 1997) (Compton and Mateas, 2006). The hierarchy model then was referenced to the level design area and be used in the game *Rogue* (considered as the first game with level design). It is reasonable to relate musical composition to the design of plat-former levels since the levels are heavily relies on rhythm . A successful level design should make players "in the flow" and "in the rhythm" to push the process forward (Compton and Mateas, 2006).

When design levels, we would need to consider the relationships between the challenge levels and the players' skills. A suitable challenge level means it is hard enough to challenge players to bring them excitement and enjoyment but not too difficult to demotivate them (TO, 2020). Zichermann and Cunningham (2011) have also created a model on players' mental state and challenge levels (see Figure 2.11). The flow zone means a mental state for players to fully engage the game play. Zichermann and Cunningham (2011) assumed in the flow zone, players would get most enjoyable, active and immersed experience. Both level design with or without suitable challenges would possibly demotivate the players.

The model also reveals a basic principle in game design: step by step. In "The Art and Science of Level Design" (Bleszinski and EpicGames, 2000), "Pacing", "Risk Incentive" and "Supply And Demand" have been mentioned as three of important level design factors. It is important to gurantee these factors balance the game level well and progressively. This is also a critical part for players to practise their skills to adapt the level challenge. The other level design elements such as "Controlled Freedom" and "Sound" etc. would also need to be considered.

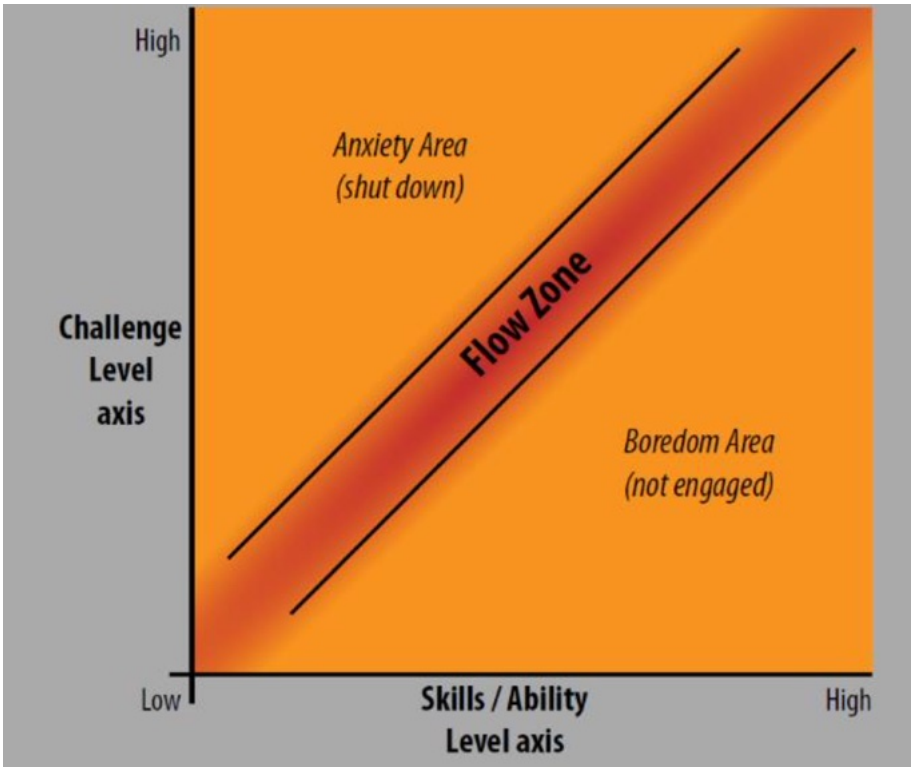


Figure 2.11: The Relationship between Players' Mental State and Challenge Level (Zichermann and Cunningham, 2011; TO, 2020)

2.3.2.3 Rewards

Rewards system or points system is a significant component of the CAR Loop (Zichermann and Cunningham, 2011). The system would directly act on players' motivations since they know they would be rewarded after getting a higher mark or after several play circles. A reward system is complex and multifaceted structure (Jakobsson et al., 2011). On the one hand, the system can be designed as a point accumulation system including skill points and redeemable points etc. (TO, 2020), which helps connect the game's inner component such as shop system and ranking system etc. On the other hand, the system has the social side effect on the game. The reward system translate the players' investments to a more comparable and communicable form, which provides the possibility to build bridges between very different games. Such community systems would pushes gameplay experience beyond single games (Jakobsson et al., 2011). Overall, the rewards system is the link between many internal/external systems of a game and the critical part to guarantee the game system to work as expected.

2.3.2.4 Shop

As mentioned above, a shop system can be considered as the branch of the rewards system, which helps realise the redeemable points system. A shop would also help to balance the "Supply And Demand" (Bleszinski and EpicGames, 2000) element in the game world. It is an indispensable part of the complete game ecology.

2.3.2.5 Boss Battles

Boss battles could be considered as another important branch generates from the reward system or we can consider it as "there will be huge rewards after". At the same time, it is also a representation of "challenge upgrade" in the challenge level designs to keep players being active (Zichermann and Cunningham, 2011; Minotti, 2017). A boss in modern video game nowadays can be given character, refer to the world view and a boss fight can be hours-long, with story reveals and involving multiple stages (Lee and Tyler, 2015). In BreatheHero's case, the boss fight should adopt how the PEP controller and breathing exercise works.

2.3.2.6 Storylines and Morality System

As shown in the onboarding section, story lines can have influences on players' experience, literally such influences could be global to the whole game. A story line would help to push forward levels and make the transaction more natural (Rollings, Andrew and Adams, 2006). Morality system generate from story line and world view has also been mentioned above, which is a complex architecture beyond the video game itself. The story of the main character could become the mapping of the players themselves, which can bring positive impact when it is well designed (Casas-Roma et al., 2019).

A successful story line normally based on a story circle, which is a kind of narrative structure (Reedsyblog, 2021). Director Dan Harmon has argued a complete story circle into 8 steps (Dan Harmon's Story Circle, see Figure 2.12 (Reedsyblog, 2021; Studiobinder, 2020)):

- **You** — A character is in a zone of comfort
- **Need** — But they want something
- **Go** — They enter an unfamiliar situation
- **Search** — Adapt to it
- **Find** — Get what they wanted
- **Take** — Pay a heavy price for it
- **Return** — Then return to their familiar situation
- **Change** — Having changed

The extremely simplified model can create the connections between the game's player, game environment and world view. Compared to a linear story line structure, a story circle structure "provides an intangible momentum to the story, almost like a rollercoaster." (Studiobinder, 2020) This means that the protagonist starts at the top of the story circle would "descend figuratively" as how the circle goes and reach their "figurative low point" in their adventure as reach the bottom of the circle. Then they "rise to success" as visually represented in another half circle (Reedsyblog, 2021; Studiobinder, 2020). Such model is easy but complete and has more twists and turns. More importantly, it metaphors simple moral values "pains and gains" and "difficulties and hopes", which exactly what we want to represent to our patients to connect them to the main character and bring them positive impacts.

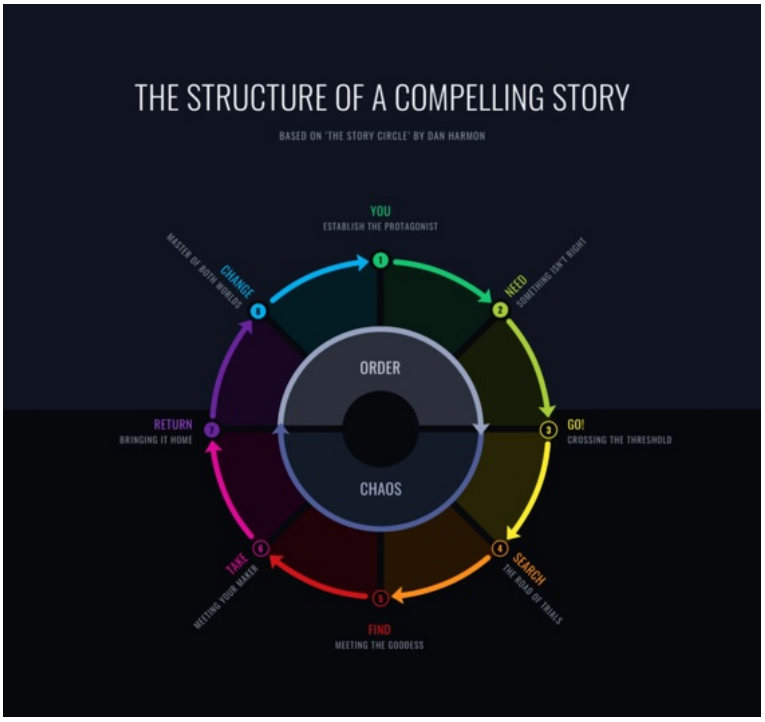


Figure 2.12: The Dan Harmon Story Circle (Studiobinder, 2020)

2.3.2.7 Challenges and Achievements

Challenges and achievements is another circulatory system in game design. The system records player's milestones and aims to bring players suddenly surprise for enhancing their sense of accomplishment and enjoyment (Ge, 2018). By applying the rewards (badges, titles and ranks etc.), the system would help to motivate the players. The achievements can be generated from the following aspects: mechanics, game style, progress, challenges, events (story related elements) and wins (such as different ways to win) (Ge, 2018).

2.3.2.8 Conclusion

All the game design elements above have the ability to motivate players and bring them longer term interests. By applying them, we aimed to create a game ecology as complete as possible (see Figure 2.13). The game would have the CAR loop (Zichermann and Cunningham, 2011) as its core mechanics and be assisted with boss battle and achievement systems. The shop system would be the complementary incentive mechanisms for the reward system. The story line and morality system would effect on all stages of the game.

For each branch system (CAR, Boss Battle and Achievement etc.) of the game, there would be several functions to be implemented. However, applying all the elements at the same time would be insufficient in development and may make players feel overwhelmed and even bring negative effect to the positive treatment effect we expected (TO, 2020). The development direction and functions would need to be chosen carefully and as far as possible in line with the needs of patients. In this case, we would like to take the design and development principles form agile development and generate our development plans: The development would be divided into several stages (called sprint). For each sprint, we would choose several functions from different branch systems as candidates and start a survey for collecting willing on which

functions we should develop. In this way, we aim to guarantee that what we developed meets the requirements of the players as much as possible and get the balance of the integrity of the game ecology.

Figure 2.13: The Game Ecology in Design

2.3.3 Game Mechanics

A game's mechanics are the rules and procedures for players and game to interact with each other. The mechanics defines how the players would play the game and the game responses to them (Boller, 2013). It is not difficult to notice the game BreatheHero requires special game mechanics to connect the player's breath to the game control.

According to the steps of breathing exercises above, the game mechanics would need to 1) guarantee the players breathe long enough in a single breath. 2) guarantee the players need to breathe strong enough in a single breath. 3) guarantee the player need to cough out the mucus after several breaths. In previous researches by Balli (2018), the breathing movement of the players has been mapped as:

- Character's movement (Globule and Heritage)
- Projecting (Ange-Gardien)
- Controlling vehicles such as space ships etc. (PEP Hero, Bloïd and Les Aventures du Briand)

It is not difficult to infer that relates movement to breath would be a good idea for 2D side-scrolling games but also as mentioned, such mechanics lack of performance on control over the rhythm and intensity of breathing (Balli, 2018). The connection between projecting and breath has been considered as the most logical design in previous BreatheHero research (TO, 2020), especially breathing fire. The players would need to breathe harder and longer to

make more damages with the design, which can enhance the control over the breathing norms in breathing exercises.

2.4 Development Life Cycles

Software engineering is a subject that aims to create 'systematic, disciplined, quantifiable' architectures for software development (IEEE, 1990). For supporting such purpose, a series of software development life cycle (SDLC) models were developed. As a software based research project, scientific SDLC is required for managing the development process, sorting the requirements of the players and keeping connections with the users.

Agile is one of the SDLCs which defined as "a development mode that is easy to planning and understanding, flexible to develop and friendly to changes." (Fowler, Highsmith et al., 2001) Such SDLC can perfectly fit the development process of projects with lightweight teams and short development terms such as BreatheHero. An agile SDLC would usually divides a project into several development cycles (in our case we would call it as Sprint as Scrum does, which is one of the development models under agile (Schwaber, 1997)) and for each sprint, it would have three phases:

- **Pregame Phase:** The phase for creating backlog (requirements collection from customers, in our case, the young patients)
- **Development Phase:** The phase for implementing the content in backlog from pregame phase, normally with a reasonable time limitation
- **Postgame Phase:** The phase for releasing the product, for each sprint, the developers should always guarantee the software's runnability (for testing and release purpose)

The Sprint structured with the three phases will iteratively appear in the development until the software is completed. Such model guarantees the communication between the participants (pregame phase), the development process (development phase) and test and release (postgame phase).

2.5 Field Study for GCP

Field study is the only way to test the result, receive feedback and conclude experiences on the game design. The field study would normally contains the following methodologies: organizing participants, designing questionnaire, collecting data and even interview. Evaluating the field study models in previous researches would help us to generate our measure models. In this section, we would discuss Balli (2018)'s field study model, Oikonomou et al. (2014)'s field study model and the model used in previous BreatheHero (TO, 2020). All models above provide constructive suggestions for our model design especially the model used in previous BreatheHero (TO, 2020), which is the most instructive model.

2.5.1 Balli's Research (2018)

We have evaluated Balli (2018)'s research which contains 6 GCP game prototypes (Globule etc.) above. The research was designed as participatory and agile. The field study was well integrated into the development life circle (see Figure 2.14) which was organised as iterative

process (similar to idea of Sprint). The field study was happened at every circle's testing phase. Such model can investigate the prototype's effect on the treatment in each development stage and involves the children's feedback to the development.

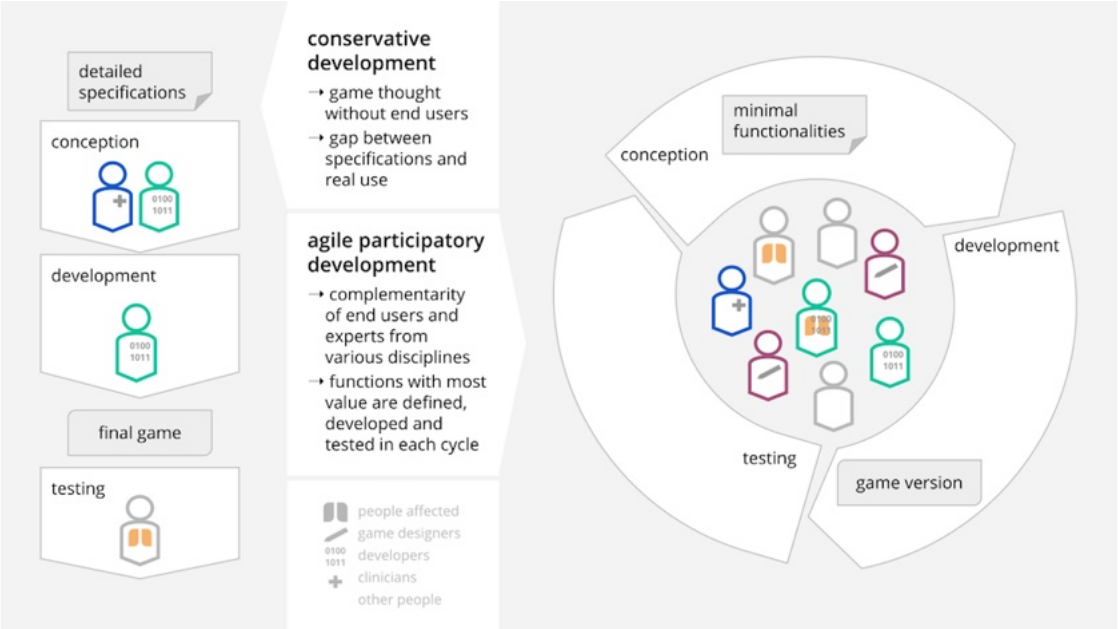


Figure 2.14: Development model for Balli's Research, the Field Study Happens at Testing Phases in Each Circle (Balli, 2018)

The initial tests were designed as following. Parents and the children with CF who were interested in the project were invited. The research obtained 10 participants aged 7-11 years. A preliminary questionnaire on their play preferences (e.g. favorite game, daily gaming time etc.), design preferences (e.g. favorite music, favourite spaceship design etc.) and backgrounds (accessed technologies and devices and willing to the treatment etc.) were answered. The questionnaire is for collecting basic user data and designing long term treatment plan for them.

In each testing phase, children were invited to play 2 or 3 games with each game for 1 or 2 levels and then answered another questionnaire. The second questionnaire was mainly on evaluations of each game (e.g. difficulty of the game, integration of physiotherapy, directions for improvement and the worst part of the game). The feedback then were used for generating conclusions and as for development suggestions for next stage. The result effectively concluded the most attractive game design and the inadequate design for the prototypes (Balli, 2018). However, although the research showed positive results on enhancing children's adherence of breathing exercise, it was too subjective and lack of criteria. The advantage of such model is mainly on its ability for timely feedback.

2.5.2 Oikonomou et al.'s Research (2014)

Oikonomou et al. (2014) started a similar research in 2014. The research created three more completed GCP game prototypes (Pirate Quest, Flower Garden and Dragon Cave) for testing whether the prototypes can help young CF patients enhancing their adherence on their treatment. The study field also provided instructive guidance for building the model in the previous BreatheHero research (TO, 2020). The field study in the research had more statistical instruction, data collection plan and purposeful questionnaire design. Totally 14 participants

aged 2-12 years took part in the research. For each participants, they would experience two stages of trials: the baseline trails and game-based trails (for comparison purpose). For each trial, there would be different questionnaires for user information and feedback.

In the baseline trial (the trail for their normal PEP sessions), the questionnaire was designed into two part and the information on their standard PEP sessions were collected. The data is for comparison the adherence for breathing exercise with the game-based trial.

- **Part I:** The questions on parameters of children's breathing exercise
  - Average duration of per PEP therapy session
  - The times of PEP therapy sessions daily
  - The repetition set times for each session
  - The number of sets for each session
  - Times of day for daily PEP therapy etc.
- **Part II:** Daily based questions would need to be filled per session
  - Ease of persuading the children to do the breathing exercise
  - Ease of keep the children's interests of the exercise
  - The willing of children to take the exercise on their initiative etc.

In the game-based trial (the trail with GCP games), the data was collected by the system (game play time and breathing times in a session etc.) and usability study (user feedback etc.). The research was designed as following.

- **Game System Collection:** For determining if the GCP games designed can literally help children enhance their engagement of the treatment and the features can be improved, direct feedback from players including setting preference, time spent on game etc. would need to be collected.
  - IP address (identifying different players)
  - Player's age (research purpose, for determining age-based gaming preference)
  - Level, score and death number in the game (For game development)
  - Time spent in the game (total), in-game and in-menu, start and end time for the sessions
  - Maximum pressure, average pressure (treatment research)
  - Number of breaths, average breath time, maximum breath time etc. (treatment research)
- **Usability Study:** The study is for collecting the gaming information, impression gaming experience and further suggestions on the testing game from the user.
  - Previous gaming experience (e.g. how often the child plays video games, the game preference, game play devices and if have previous GCP history and the opinions on it etc.). Such data was used for discerning how the previous gaming experience would effect plays' expectation and impression for the testing games.

  - A statements list survey for collecting playing feedback
  - Subjective and detailed answer were required from the players on their opinions about the different features of the game. What they enjoyed the most and the didn't like.

In the result session, by analyzing the collected data and comparison the two trials, the following conclusion were made:

- **For the general information collected**, all participants performed daily PEP sessions. Most sessions last for 10-15 minutes, some other with a time of 20 minutes and even up to 45 minutes. This depended on the children's current health.
- **For data collected from baseline trial**, 1) The average duration of time per treatment session was between 10 to 30 min. 2) The times for PEP therapy was between 1 to 2 times per day. 3) The repetitions of each session was 10. 4) The number of sets performed per session on average was 6. 5) The PEP therapy usually performed early in the morning and late in the afternoon (before and after school). 6) The difficulty on persuading the child to start a PEP therapy was 2.5/5. 7) The difficulty on maintaining the child's interest in the PEP therapy was 3/5. 8) The ways on motivating child to engage with PEP therapy are usually: none (no actions), followed by offering incentive, followed by incentive and threaten discipline, followed by only threaten discipline.
- **For data collected from game-based trial**, 1) A number of erroneous data was collected by the system but filtered out, which possibly due to abnormal session terminations etc. 2) In a two weeks research, the total sessions recorded were 719, the total completed sessions were 264. Total time played by all participants was 42 hours. 4) Total breaths by all participants was 14876. Time spent on menus was 46 minutes. Player death in the game in total was 152 times. 7) Player average age was 7 and age range was from 2 - 12.
- **By comparison the data from two trials**, 1) The total time spent on PEP therapy in baseline trial was 49 hours and game-based trial was 42 hours. It is surprisingly that the baseline trial has more spent time than game-based trial. However, the research team suspected that the total amount of time patients engaging with the games was higher than it was present since there were large number of incomplete sessions (455 sessions). The lack of data for these sessions has resulted in potential conclusion deviation. The reasons of such data losing had been under investigated and would be resolved in next study.

Although the study has problems on insufficient participants and short experience period, which has bought experimental deviation and inaccurate results. It has bought significant overview on the children's PEP therapy situations, how much the motivations they have and the struggle of them. This is important since the information gives guidance for GCP's development directions. In the practise, the data losing situation has given us a lesson on the design of data capture. The information on game preference has inspired us to generate our own game features. The field study design also provided us the pattern for our user study.

2.5.3 TO's Research (2020)

In BreatheHero's previous work (TO, 2020), a field study model based on System Usability Scale (SUS) (Brooke, 1995) and Intrinsic Motivation Inventory (IMI) (Ryan, 1982) has been

designed. The study answered two research questions with the testing BreatheHero version: RQ1) What kinds of game features can be used to motivate the young patients' engagement of PEP therapy? RQ2) Can BreatheHero enhance young patients' engagement of treatment? It was a pity that due to the restrictions of COVID-19, only 5 healthy participants were joined in the research.

The field study was designed into two parts: 1) The participants were invited to play the game for one week everyday (for simulating daily PEP therapy). 2) Questionnaires on user feedback were filled after the gameplay. The questionnaire was designed into five sections and covered a wide range.

- **Section 1: Player's Background Information**
  - Gender, age and medical history on lung diseases
- **Section 2: Game History**
  - The time spent, preference and play devices on video games
  - History and knowledge on gamification technology for therapy
  - The game play style that applies to the player
- **Section 3: System Usability Scale (SUS)**
  - The usability and opinions on the system
  - The understanding and opinions of the game's game objects
- **Section 4: Game Features**
  - The enjoyment level for the game features
  - Open questions on enjoyed game features and suggestions for improvement
- **Section 5: Intrinsic Motivation Inventory (IMI)**
  - The interest level when play the game
  - The tension level when playing

The field study model was instructive since it evaluates the system comprehensively including usability (if the system is easy to use), game's effect (interests, enjoyment and tension level) and game designs (features). Positive feedback were collected from the study, previous BreatheHero was given high SUS score (81.0 ± 3.4) and IMI interest/enjoyment score (4.7 ± 0.7) and became strong evidence for answering RQ1 and RQ2. Although the field study had the problem on lack of participants, the model can be continued to use and improved in the new BreatheHero study.

Chapter 3

Requirements

As a programming based project, we would need to analyze the requirements of the software so that products are produced correctly. Being from the section, the development process of BreatheHero would start and this session is consisted of collecting and analysing requirements from previous research, extracting demands from the collected requirements and generating the requirements fit the current BreatheHero Project.

3.1 Requirements Collection

BreatheHero is purposed to improve the adherence of young patients' chest physiotherapy and keep them long-term interest on the breathing exercise. The project is a combination of gamification technology and chest physiotherapy technique. In previous chapters, we have introduced the principles of gamification (Zichermann and Cunningham, 2011) and the mechanism of breathing exercises (Physiopedia, 2021), which helps us link the game with the breathing therapy and the core mechanism can be concluded as the interaction between breathing behavior and CAR loop (Zichermann and Cunningham, 2011) (see Figure 3.1). Based on the core mechanism, the other requirements would be generated and be the extension of the system.

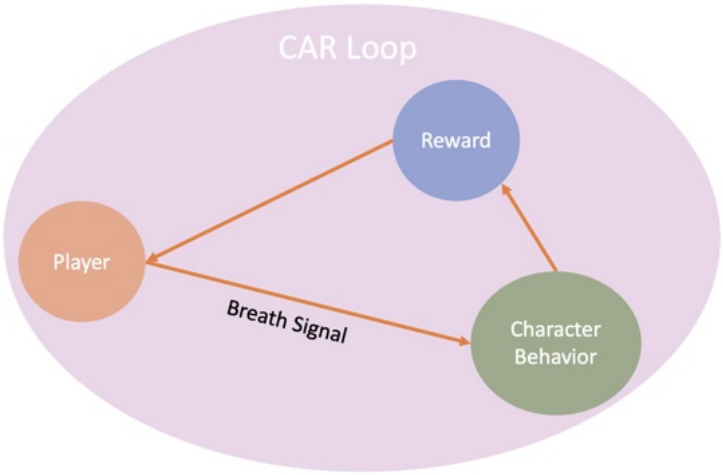


Figure 3.1: The Game's Core Mechanism

The previous researches mentioned in chapter 2 related work have provided instructive guidance for locating the requirements and mapping out the pain points and shortcomings of the system that needs to be solved. These experience could be valuable suggestions for generating and specific our requirements for the current stage.

A serious of challenges of the previous system can be concluded as following (see Table 3.1), the data can be significant supplementary materials for evaluating the priority of the requirements later.

Table 3.1: The challenges concluded from the previous researches

Challenge No.	Challenge Description
C0	A basic mapping of the therapy and the game system
C1	Vivid mapping of the player's breathing behaviour
C2	The replaybility of the game system
C3	The collection of the valid breathing data
C4	The portability and instant-playability of the system
C5	A better combination of system and breathing exercise mechanism
C6	Monitoring and recording of the player's long-term treatment

Requirements from previous researches are also collected and a table for the requirements backlog were created in previous BreatheHero research. Based on the previous requirement backlog (TO, 2020), we created our analysis table (see Table 3.2) by extending the collection, linking the requirements to their resolved challenges (see Table 3.1) and evaluating the priority for each requirement (level 1-3, level 1 means the highest priority, the value is calculated by subjective weighting, recommendation and weight in the previous researches). We then analysed the requirements and extracted the most needed entry into our backlog for current BreatheHero.

As shown in the table 3.2, R1 to R7 are related to C0 as they are the basic elements of the system. These elements also cover the functions for collecting and monitoring breath data and system optimization. Due to they are the fundamental features GCP games, the priority has been set as 1. The left requirements are mostly related to domain C2. These elements will contribute to the basic system to make it more like a game and try to figure out one of the major challenge - the game's replayability and long-term interest for the children. After considering the time constraint and the recommendation level from the previous researches, different priority are assigned.

We have also introduced four novel features which are not fully developed in previous researches: N1) A PEP controller attached somatosensory and sound sensors. N2) Refactored with Godot engine so that supported the web platform. N3) More completed game circle. N4) More scientific development process. And N1 is related to C0, N2 related to C4 and N3 related to C2. The features would also be defined as requirements and wait for implementation in the development. N4 requires applying industrial development technology to the project and would be specified in the design section. It is the guarantee of efficient development and excellent product.

Table 3.2: The requirements backlog from previous researches with description, linked domain (see Table 3.1) and priority

Requirement No.	Game Features	Description	Relevant Domain	Priority
R1	Breathing controller connecting to laptop	The breathing controller would need to be connected to the computer for translating the breathing pressure to digital signal.	C0	1
R2	Simple and clear game mechanism	The game need high playability for fit all age patients.	C0	1
R3	Suitable game sessions	The length of the game session should fit the length of the breathing exercise in case of any overdoes or understate.	C0, C5	1
R4	Character's response to breath behavior	The game character would need to response to the players' breath.	C0, C1, C5	1
R5	Game's response to cough behavior	The game would need to response to the players' cough for finishing the breath exercise.	C0, C5	1
R6	Breath data collection	The system would need to have the ability to collect the player's breathing pressure data etc.	C0, C3	1
R7	Breath exercise record	The system would need to have the ability to record the player's breathing exercise length and times daily.	C0, C6	1
R8	Game tutorial	The system needs to give guidance to the players.	C2	2
R9	Progressive level difficulty	Practising players' game skills and provide longer-term interest. Incresing playability.	C2	2
R10	Storyline and morality system	We hope the story line can provide players more fun and longer-term interest.	C2	2
R11	Cartoon/Colourful visuals and environment	For creating attractive and friendly visuals for young patients.	C2	2
R12	Achievements and challenges	For longer-term interest	C2	2
R13	Web platform game version	The game would need to have the web version for easily accessing and playing.	C4	2
R14	Various play modes	Create more play modes for yours for long-term fun.	C2	3
R15	Missions	Create longer-term goal for the players.	C2	3
R16	Mini-games	Create mini-games for game's playability.	C2	3
R17	Game editor	For players to customize their game.	C2	3
R18	Level generator	A level generator can help create different content of the level, make the game more playable.	C2	3

3.2 Requirements Analysis

As mentioned, BreatheHero has been refactored from beginning so it is important to choose and list the requirements for the new BreatheHero project wisely due to the time constraint.

R1 to R7 are the basic for implementing the system and they would be included in our backlog and need to be developed with a high priority.

- **R1**: Connecting PEP breathing device to the laptops and using as the input device to capture breath behavior, is a core logic to map the chest therapy to video games. The breath pressure is then translated into digital signals and turned into character's behaviors, which triggered the begin of the game loop. Such model has been widely used in previous researches and showed great potential (Oikonomou et al., 2014; Balli, 2018). Such design is going to be used in new BreatheHero as well and in our case, we attached sensors to the PEP device which help players control the character directly with the PEP device and monitor their cough and expectoration so that the game can be paused at the right time automatically.
- **R2**: A simple and clear game mechanism which also follows the process of breathing exercise is necessary. The requirement meets the gamification principle (Zichermann and Cunningham, 2011) and is also significant for making the game fit all ages. A complex and cumbersome system is not good for therapy purpose (Balli, 2018). The game mechanism for new BreatheHero is easy and lightweight. The player would control the movement of the character, breathe for attack and then gain rewards and the rewards encourage the player to breathe again (CAR loop (Zichermann and Cunningham, 2011)).
- **R3**: In GCP's applicability and effectiveness, the potential risks of hyperventilation, hypoventilation and exercise overdo have been mentioned and (Bingham et al., 2010) has tried to prevent such risks by rationalizing the length of the game. The length of the game needs to match the rhythm of the breathing exercises, which is also valuable experience from previous prototypes (Zichermann and Cunningham, 2011; Balli, 2018; Oikonomou et al., 2014). The game round in the new BreatheHero is organised as its levels, and by setting up the time limitations, the rhythm of the game can be adjusted macroscopically.
- **R4, R5**: R4 and R5 are the components of R2. In previous research Bloïd (Balli, 2018), the control of the game was divided into two parts, the keyboards was used for controlling the character's movement and PEP devices for breathing. The control mechanism has been criticized a lot in the experiment. As mentioned earlier, by combining sensors with PEP devices, we aim to integrate the control mechanism and bring better experience.
- **R6, R7**: R6 and R7 refers to the data collection of two parts: breathing exercise records and breathing performance data (e.g. breath pressure, breath times in a session and cough times etc.). By monitoring the physiotherapy performance of the player while gameplay, we aim to evaluate the effectiveness of the prototype, improve the therapy design and gain experience for further development.

R8 to R18 are the desirable and recommended features from the previous studies, most of them are related to the replayability of the game. Some of them would be included in the new BreatheHero and some of them would not because of the time constraint, which would be decided on its usage and priority.

- **R8, R9**: R8 and R9 are about the level design and difficulty of the game. As discussed in the onboarding section. Guidance is important and necessary for new players to avoid early failures and preparing the increasing level's challenge (Zichermann and Cunningham, 2011; ZHANG and YIBIN, 2021). It has been shown that the breathing therapy's effects is potentially positive related to the factors such as breathing times and breathing intensity etc. (Darbee et al., 2004), a progressive level difficulty design can help to not only providing challenge and longer-term interest to the player but also practising their skills and enhance the load of breathing exercises. The requirements would be included in the backlog of new BreatheHero in advance.
- **R10**: The effect of story line and its morality system has been emphasisedly introduced in earlier section. The story line is not only for pushing game forward and attracting players but also creating a more completed world view, making the game world more vivid, passing on positive effects and encouraging the children. The requirement is strongly recommended in the new BreatheHero.
- **R11**: The requirement is on the visual design. As the game's players are defined as young players, colorful and cartoon visuals and environments are recommended in the previous work. Such visual style could also bring players positive senses and vibrant psychological cues. (TO, 2020; Oikonomou et al., 2014). The design style is considered in the project.
- **R12**: The achievements and challenges system has also been considered as a supplementary component of the reward system and CAR loop. The importance of the part has been discussed in the gamification section. The requirement would also be included in the system.
- **R13**: R13 is related to our novelty N2. Such requirement has been seriously underestimated in previous researches. In the related work, most of the field studies heavy reliance on the hospital equipment which brings huge inconvenience for the children and their parents. As a treatment process requires daily, we believe lightweight platform is necessary. By refactoring with Godot engine, the game can be easily presented as a web application, which brings huge portability and instant-playability. Such technology can greatly improve the patients' therapy experience and furthermore potentially reduce disgust and hostility feelings on the breathing exercise.
- **R14 - R18**: R14 to R18 are the requirements for improving the game's playability. R14 (Different play modes) and R16(Mini-games) can integrate multiple play methods together to providing players various game experiencing. R15 (Missions) is a great motivator to set game goals for players at different game stages. R17 (Game editor) can provide players chances to customize their game, however, such feature are not directly related to breathing exercises and might lead to gameplay that not related to treatment (TO, 2020). R18 (level generator) is a useful feature for generating different levels autonomously, which can strongly increase the game's replayability. Due to the time constraint and position of the features (helpful but not necessary), the features are evaluated with priority 3. They would not be included in the new backlog at current stage of new BreatheHero.

By collecting and analyzing the requirements above, a backlog for new BreatheHero with a series of requirements and their user stories are created (see Table 3.3). These requirements will be further split and detailed in the design and implementation section.

Table 3.3: The backlog with user stories and priority for new BreatheHero

Feature No.	Game Features	User Story	Priority
F1	Breathing controller connecting to laptop	As a player, I can connect the PEP device to the laptop and control the character with it.	1
F2	Simple and clear game mechanism	As a player, I hope the game has a easy play method which no need for extra trainings.	1
F3	Suitable game sessions	As a player, I hope the game's rhythm can fit my breathing exercise, I will not be too tired or too easy.	1
F4	Character's response to breath behavior	As a player, I hope when I breathe the character I control can give vivid response.	1
F5	Game's response to cough behavior	As a player, I hope when I need to cough, the game can be paused automatically and the character I control can give response. The game would be continue after I cough.	1
F6	Breath data collection	As a player, I hope the data of my breath can be stored safely for later therapy	1
F7	Breath exercise record	As a player, I hope my breathing exercise can be recorded and the system can help me manage my next exercise.	1
F8	Game tutorial	As a player, I hope when I login the system, guidance can be provided so that I can get used to the system quickly.	2
F9	Progressive level difficulty	As a player, I hop the difficulty of the levels would progressively increasing so that I can enhance my therapy and practise my skill.	2
F10	Storyline and morality system	I hope a interesting story can push the game forward and bring me positive affect.	2
F11	Cartoon/Colourful visuals and environment	I hope the game's visuals and environment are nice and friendly so that I can enjoy the game more.	2
F12	Achievements and challenges	I hope there would be challenges for more fun and I can get reward and achievements after I finish the challenge.	3
F13	Web platform game version	I hope I can do my breathing exercise at anywhere and anytime, I don't have to go to hospital for my treatment.	3

Chapter 4

Design

The section would discuss the design and give more details on the features concluded in requirements section (Table 3.3) and overall development plans. The features would further more be split into elements and how we planned to implement them would be introduced. The main system was contributed by Dr. Christof Lutteroth and me including login, control and movement, data collection, menus, game objects (coins, obstacles and enemies etc.), levels, attack mechanism, boss fights and achievements systems. The development plans (e.g. quality plan, coding style and test plan etc. See Appendix C.) and models were also designed in the stage. We would go through hardware design, software design, game design and development plan for the project.

4.1 Hardware

The hardware was designed and set up in previous BreatheHero and continued to be used by the new BreatheHero. The introduction is mainly referred from TO (2020)'s research. A computer, mouse and keyboard sets and the PEP input device were consisted of the hardware necessary setup. The setting up is on two parts: Minimum specifications for supporting the game and the PEP input device.

Minimum specifications
The original design for BreatheHero was required of a computer with Windows 10. In the new BreatheHero, such limitation was resolved by deploying web platform.

For **Windows platform**, the minimum specifications for a laptop require a CPU supporting SSE2, graphics card with DX10 (shader model 4.0) and supporting USB 3.0 (TO, 2020). The game currently does not support MacOS or other operating systems but instead, other operating systems users can play it with a web browser.

For **other operating systems**, with the help of Godot engine, a web version of the game was exported and it can be played with browsers by visiting `http://breathehero.com/`. A computer that supports modern browsers (e.g. Chrome, Microsoft Edge and Firefox etc.) and USB 3.0 is required. The application was deployed with Google Firebase. Such design is greatly improved the game's portability and instant-playability.

PEP input device
The PEP input device was developed by Dr Lutteroth's team in previous BreatheHero. Its

design was introduced in TO (2020)'s work as following: the PEP input device has three components - a PEP breathing input component, a white adaptor (contains remote, sound and breathing pressure sensors and the USB component) and a mouthpiece (see Figure 4.1). The device is similar to the design of the input device in Balli (2018)'s research. However, the sound and remote sensors it has can help detect play's coughs and accelerometer to achieve features that pause the game intelligently and integrates the control system.

Figure 4.1: The Game's PEP Input Device with USB (TO, 2020)

4.2 Software

The previous BreatheHero was developed with Unity Engine version 2018.2.14f1. Since Unity is very sensitive with its version and the bad usage of the version control and file management, the code was broken. The project was refactored in July, 2021 and different game engine and version control tools were used.

Development software setting up

The new BreatheHero was refactored with Godot engine v3.3.2.stable.official. The engine provides more concise file management system, scene settings, multi-platform export support and stronger robustness. By porting the game to the godot, the following problems have been solved or optimized:

- Messy file management
- Incompatibility caused by version sensitivity
- Operating system limitations

The project was used godot script as programming language and complete development guidance was provided by the official website (godotofficial, 2021).

Version control
Github was used as the version control tool for the project. A version control plan would be specified in development life cycle and management section and attached in the Appendix C. There would be dev branches for each developer and each pull request would be reviewed by Dr Lutteroth. This aims to guarantee the runnability of the main branch and in case any broken of the code.

4.3 Game and Design

The section would furthermore break down the requirements into components and discuss the rationale of the design. The section would include menu design, gameplay mechanism, game objects, level design, Storyline, reward and shop system and achievements system. Due to the time constrains, the game fnished a full play circle for chapter1, which has contained most of the game elements and features. A series of Demos has been made and the links are attached in the appendix, which aims to provide a better understanding of the game features (see Appendix A).

4.3.1 Menu Design

The subsection is going to discuss the design of the menus and scene transition. The menus include the login scene, therapy records,level selecting and other scenes. The pages used a unified grid design provided by godot engine, which is more clean and simple than the previous BreatheHero.

4.3.1.1 Login Scene

The login scene is designed as a input of username (email) and password. Register was also provided (see Figure 4.2). The login data then would be sent to database provided by Google Firebase for verification. Keyboard and mouse are used as the input device in current stages. The PEP device would only be used during the gameplay. The login feature is for identifying the patients and providing a series of treatment recording and tracking purposes.

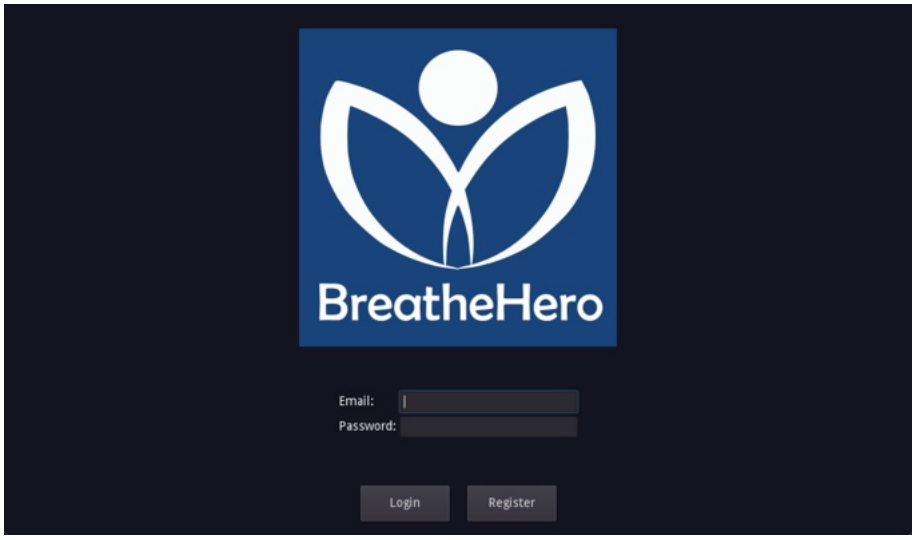


Figure 4.2: Login Scene

4.3.1.2 Clinician Scene

After the verification, the patient's information would be downloaded from the database and shown in the clinician scene (see Figure 4.3) including the basic information, daily CPT minutes, CPT pressure and current therapy records. The data would provide instructive suggestions for further therapy.

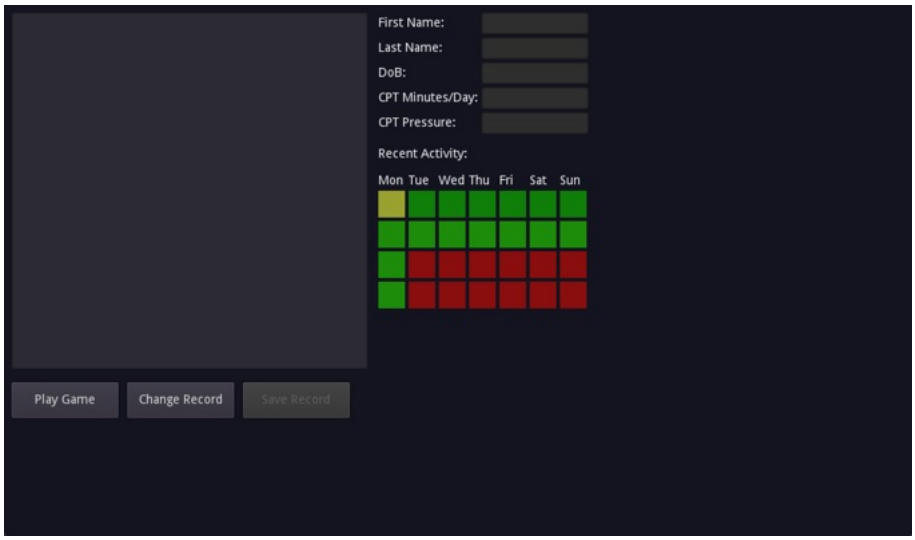


Figure 4.3: Clinician Scene

4.3.1.3 Game Entering and Level Selection Scenes

Under the guidance and evaluation of a physician, the scene then would be translate to the game entering scene (see Figure 4.4). A device connection pop-up would appear at the scene for connecting the PEP controller device. By clicking the play button, the player would be sent to the level selection scene (see Figure 4.5). Additionally, there is feature for time limitation, which aims for preventing patients from indulging and overdoing. The feature would be presented as a pop-up. Players can freely select the levels and stories want to play and review.

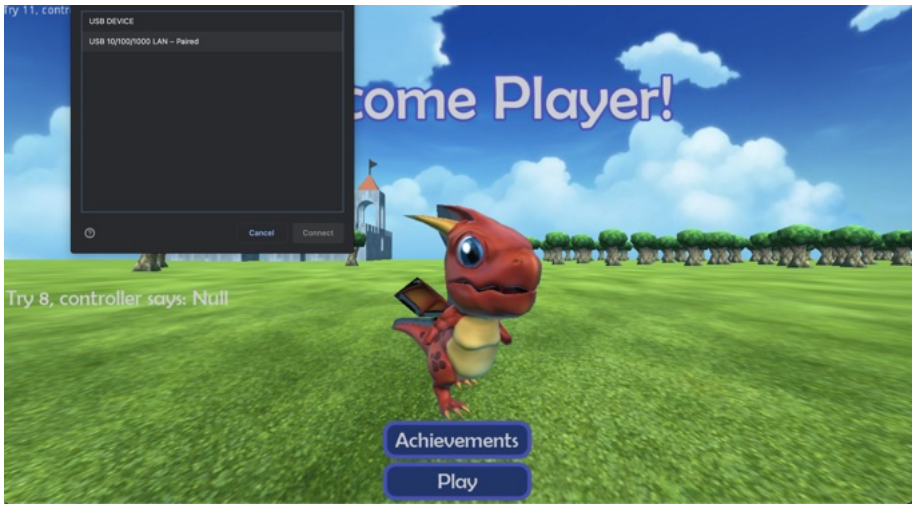


Figure 4.4: Game Entering Scene

Figure 4.5: Level Selection Scene

4.3.1.4 Other Scenes

The other scenes including achievements and shop etc. The shop due to the time constrains was not able to be deployed in time. The scenes has constructed a navigation cycle as following (see Figure 4.6). The guideline created by the circle is linear and simple, which aims to make the system easy to use.

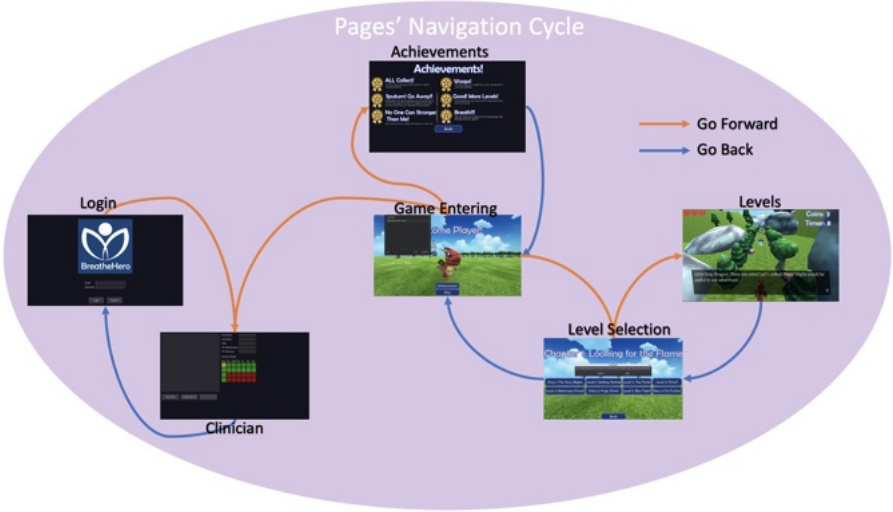


Figure 4.6: Navigation Cycle for Scenes

4.3.2 Game Objects

The game objects were designed as reusable, which means an object can be used in many scenes to fit different situations that require similar features. The code was designed with same idea. Same objects share the same script but it would meet different requirements. The game objects mainly include pickups and obstacles.

4.3.2.1 Pickups

Due to the refactor, we were not able to create many kinds of pickups. There are three kinds of pickup: Copper Coin, Silver Coin and Gold Coin. Each coin has different score value and two states - frozen or not frozen. All the coins were generated from the same object and connected with the same script which can control its values and states (see Figure 4.7). The states (frozen or not frozen) are related to the main play mechanism of the game. The player would need to breath with the PEP device to melt the ice then collect the coins.

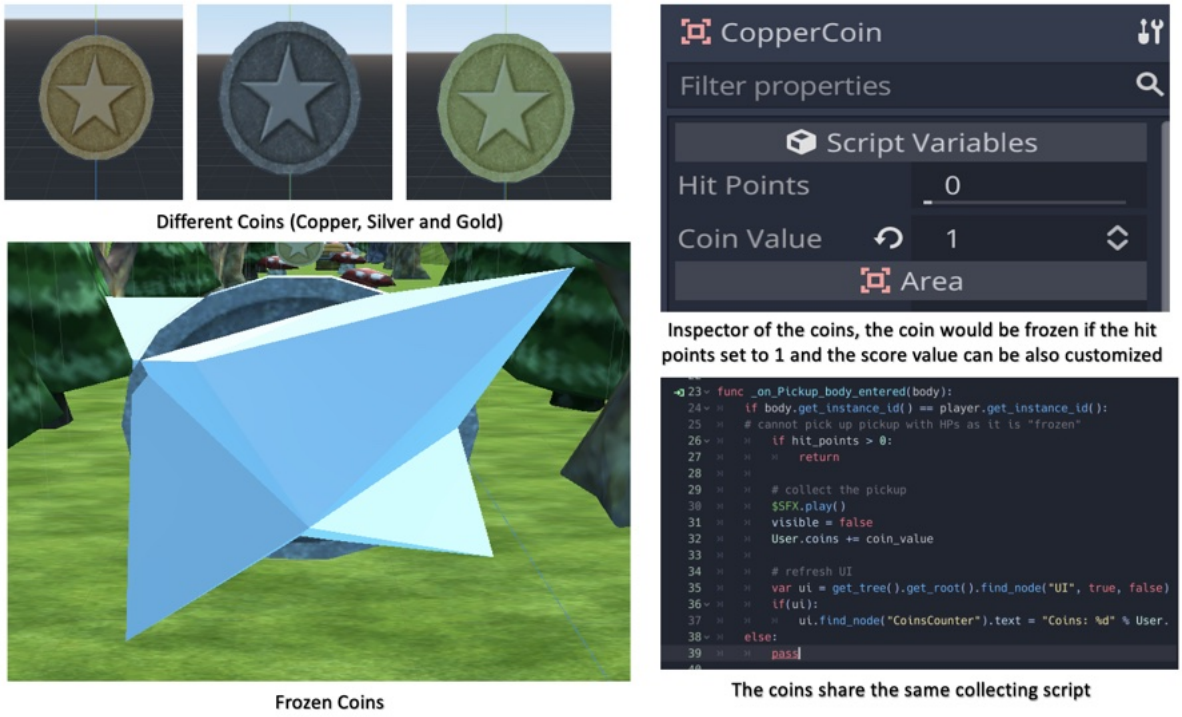


Figure 4.7: Setting-ups for Coins

4.3.2.2 Obstacles

There are two kinds of obstacles developed, treasure chest and slime. The reusable idea was also used in the development.

Treasure Chest

The treasure chests were designed as the huff-cough hurdles and deployed on the player's only way to the finish line. When the players enter the treasure chest area, they would need to cough three times for breaking it to continue the game. The treasure box is used for forcing players to cough. By applying the combination of coins and treasure chest, we aim to adjust the player's breathing and cough rhythm at a macro level (see Figure 4.8).

Figure 4.8: Setting-ups for Treasure Chests

Slime

Slime is designed as the enemies in the game. They are distributed at the side of the paths the players need to go and mainly plays the role for obstructing players collecting coins. There are two kinds of slime, little slime and boss slime. A little slime is coded as would keep moving left and right in the path. When a player gets hurt by it, the player would lose HP. The little slime is significant to increase the difficulty and fun of the game (see Figure 4.9 little slime).

A huge slime is a large version of slime that generated from little slime, which is designed as a boss in the game. Instead of only moving left and right, it would generate little slime rushing towards the players. Players would need to breathe to the little slime to push it back to hit the huge slime (see Figure 4.9 huge slime). The mechanism of the boss fight would be introduced in detail in later section.

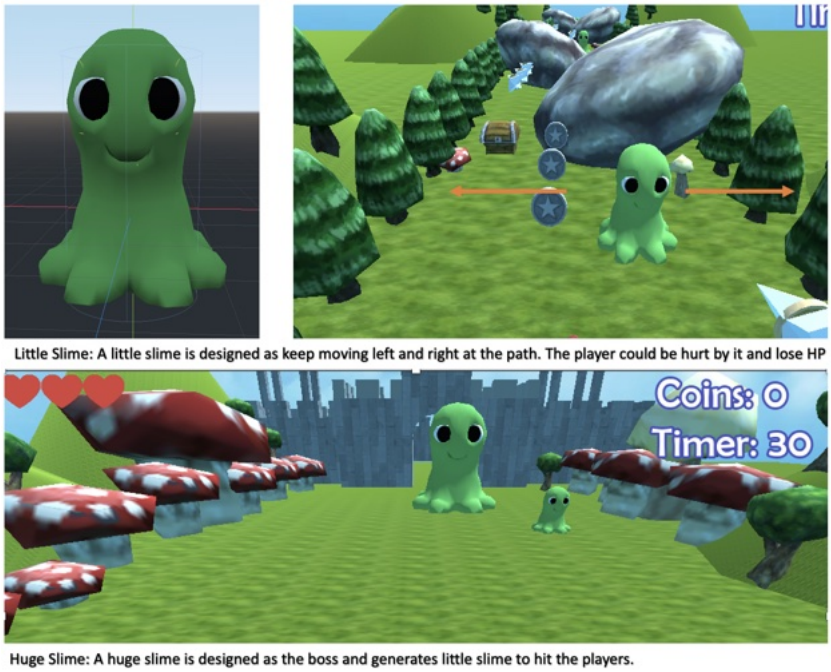


Figure 4.9: Setting-ups for Slime

4.3.3 Gameplay Mechanism

The new BreatheHero reused some of the gameplay mechanism designed in previous research (TO, 2020). And there are two main game mechanisms - regular levels and boss fight. Both mechanisms integrate the breathing exercise with the game and push forward by the CAR loop.

4.3.3.1 Regular Levels

For regular levels, the game was designed as the player controlling a little red dragon traversing a linear path to reach the finish line (see Figure 4.10 level overview). For each level, there are two basic elements: coins and huff-cough hurdles. In later levels (starts from level 3), enemies such as slime would be added in. The goals for the normal levels are reaching the finish line and collecting a certain number of coins. The dragon would go forward constantly (which no need for control) and the players would need to control the dragon left and right to collect coins and avoid the enemies with the PEP device by the accelerometer sensor attached. Breathing exercises are integrated into the game in the following way: The frozen coins would distributed in the countryside and normally three consecutive as a group (this is for guaranteeing the players to breathe long enough to reach the therapeutic effect). The players would need to breathe with the PEP input device to control the little red dragon to breathe out flame to melt the ice so that they can collect the coins. After three frozen coins, a huff-cough hurdles would be set on the path for forcing players to cough out the sputum, the game would be paused when the players reach the huff-cough hurdles area (see Figure 4.10 frozen coins and huff-cough hurdles sets). The treasure chests would be broken by coughing three times so that the players can continue the game. The cough sound would be detected by the audio sensor attached on the device.

Level Overview: The basic elements including coins, treasure chests and slime. The players need to reach the finish line to pass the level.

Frozen Coins and Huff-cough Hurdles Sets

Players would need to breathe to melt the ice on the coin to collect them.

Figure 4.10: Regular Level Mechanism

4.3.3.2 Boss Fight

Boss fight is new for the project and plays an important role for supplementing the game play circle. A huge slime is set at the last level of chapter 1 and generous rewards would be given after the battle. Different mechanism was designed for the scene. For the boss fight, the players would be set at the start line and can only move left and right (which is also achieved by the accelerometer sensor attached). The boss slime is moving around in a distance with the player and would keep generating little slime every 10 second to rush towards the player. For hitting the boss and winning the fight, the player would need to keep breathing fire (with the PEP device) to the little slime that rushes to the little dragon and cough three times to push the little slime back when it is spin (which means the player has breathed long enough to hit the little slime). When the spinning little slime touched the boss, the boss would be damaged (see Figure 4.11). Similar to the regular levels, the rhythm of breathing exercises are controlled by the speed of the little slime appears and breathing length is decided by the hit-points of the little slime. The mechanism is different from the regular levels which can bring more fun for the game and moreover manages to integrate different play modes with the breathing exercise.

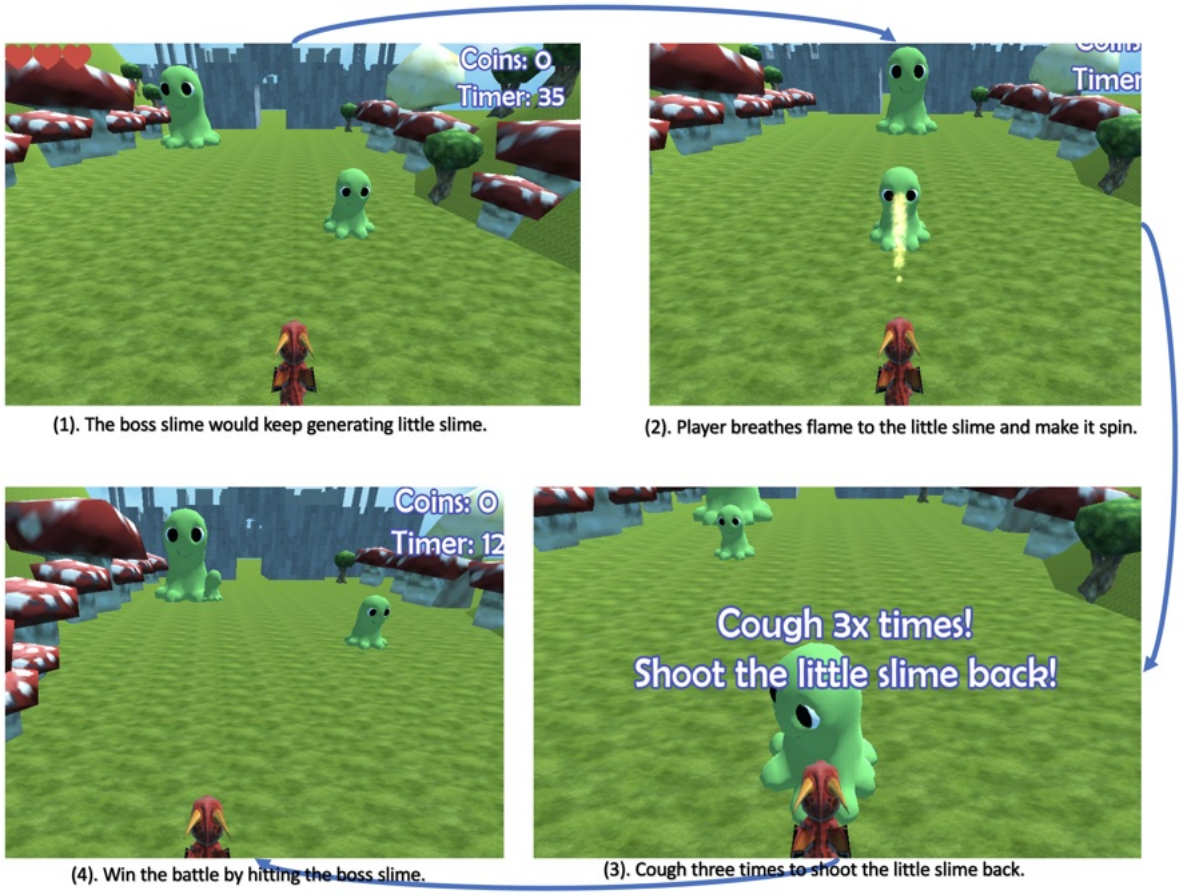


Figure 4.11: Boss Fight Mechanism

4.3.4 Level Design

As mentioned in Zichermann and Cunningham (2011)'s research, level design with increasing difficulty could, on the one hand, guide players to practice their skills, on the other hand, gradually increase the fun of the game. The level design of BreatheHero has fully followed the principle. The first level has been designed with no enemies and tips for breath and cough was given. Players would only need to breathe at least twice to pass the first level safely. In a later level, enemies were set and the player would need to at least breathe four to seven times to pass the level (see Figure 4.12). Such design was evaluated by the daily entertainment score in the field study and showed positive result for maintaining the player's interest curve, which would be discussed in the later section.

Figure 4.12: Level Design with Increasing Difficulty

4.3.5 Storyline Design

Inspired by Dan Harmon's story circle, a script based on the following story circle (see Figure 4.13) has been designed. In the story, the little dragon was designed cannot breathe flame at the beginning. However, the little one never gives up looking for his/her flame and keeps forward on the adventure. With the support of friends and practise thousands of time, the little one finally finds the flame and starts another adventure. We designed the story that aims to mapping the young patients and guide them to move forward bravely and don't give up. We hope to convey a positive worldview and moral system. In the field study, a question on "If the participant has been inspired by the plot?" has been set and we received positive replies from all participants, which is exactly what we want to convey.

The story appears in the form of one scene after another and interspersed in the game levels, which is designed to push forward the levels so that all levels can be linearly connected and finally improve the game fluency and experience. Due to the time constraint, only chapter1 for the storyline is completed. Nevertheless, we still maintain the integrity of the content of this chapter and leave a suspense to facilitate future design and development. The complete script can be seen in the Appendix B.

Figure 4.13: The Story Circle Designed for BreatheHero

4.3.6 Reward and Shop System

The reward system is one of the most important components of the CAR loop (Zichermann and Cunningham, 2011). And according to Zichermann and Cunningham (2011)'s gamificaiton principle, the reward achieved in the CAR loop should be further used to better provide players with motivation to continue playing. In this case, reward and shop system was designed for completing such mechanism. The reward (in our case, coins) can be collected in the levels and would be stored into the user's database so that it will be preserved as a long-term asset of the user (see Figure 4.14 reward). This is the basic implementation of the CAR loop, which is "users play the game, gain the asset and the accumulated wealth allows players to play more games".

The shop (see Figure 4.14 shop) is designed for further using of such asset and based on simple trading rules. In the project, the shop system was designed and implemented by Gaozhan Sun, one of my colleagues. Due to the time constraint, the shop was not fully implemented and ideally, after her implementation, we would be able to connect the two components (reward and shop systems) together.

Figure 4.14: Reward and Shop System

4.3.7 Achievements System

The achievement system is positioned as an post-credits scene system and in order to make the software icing on the cake. The achievements were hidden at the beginning and appear when the players trigger them with some movements. We have also carefully considered how achievements are expressed, and they have been designed to be interesting and positive (see Figure 4.15). Such a subsystem greatly increases the interest of the main system and to a certain extent gives players the motivation to continue to explore. The field study also asked the participants for their opinions on the achievements and most of them have given positive feedback.

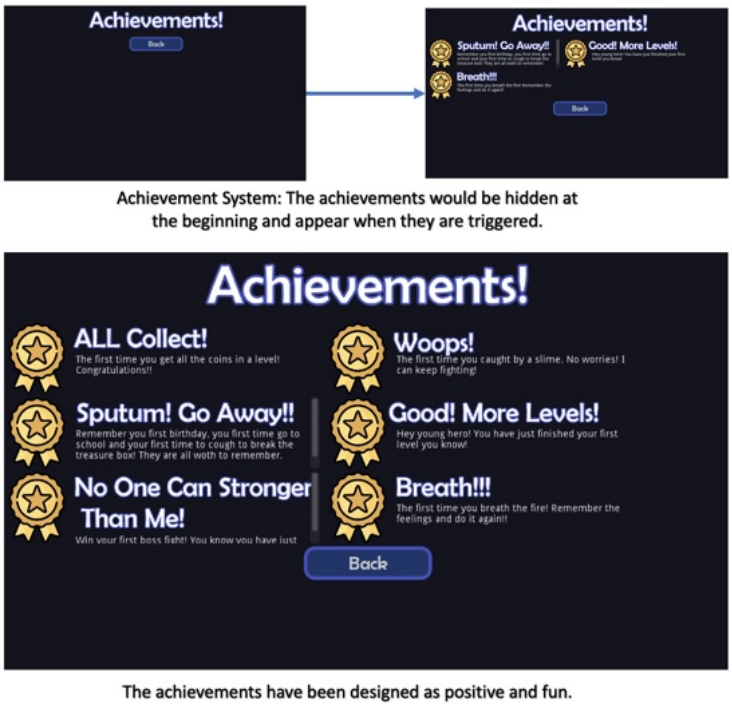


Figure 4.15: Achievements System

4.4 Development Life Cycle and Management

After the code broken was detected, we realized that standardized development process and management for the project is required. This is for making further development possible and at the same time, it is the basic of success of the project. In this section, global development rules would be defined including development life cycle we token, development plans and other development tools we have used.

4.4.1 Development Life Cycle and Plans

As mentioned in the early section, agile development life cycle has been used for the project. There are different development patterns in agile such as Scrum, Extreme Programming and Feature Driven Development etc. (Abrahamsson et al., 2017) Since agile is defined as "can be arbitrarily combined and customization" and is a highly liberalized model collection rather than a fix development pattern, the following agile methods/ideas have been taken and considered as the methods/ideas that most fit the project's development scene.

- **Pre-planning:** In some of the agile models, pre-planning is included. The plans help to build the high level architecture and standardize the development process, test process and file management. The following plans were written for controlling the development macroscopically and is useful for further development including quality plan, coding style, test plan, review plan and version control (See Appendix C Development Plans).
- **Sprint:** The idea of Sprint is collected from the development model Scrum, which is a lightweight development model suitable for small teams. As introduced in related research section, after the high level design, the project would be pushed to the development phase and be into the loop of sprints. During the loop, the features in the backlog list would be implemented sprint by sprint. The advantage of the model is that such loop can jump out at any time which means the backlog can be complemented anytime and even the process needs to go back to the high level design stage.
- **Brainstorming:** Brainstorming is the process for gathering ideas and inspirations. As a project includes design and pursues novelty, the method is also included. The brainstorming can happen at anywhere anytime and aims for recording and organizing ideas. A whiteboard was used for the purpose and a lot of interesting designs were literally starting from a post-it on the whiteboard. The whiteboard and designs during the development is attached at the Appendix D Design Diagrams D.1 Whiteboard and Brainstorming section.
- **Let the players join in:** As mentioned, we hope give the right for development back to the young patients. The similar idea can be found in agile as the development group would have regular meetings with the customers for collecting requirements (Abrahamsson et al., 2017), which aims to guarantee the final product can meet the requirements better. In our case, for each sprint, we designed a survey for the players which contains several optional features to vote. The feature with the highest score would be developed in advance. With such form, we want to invite the young patients join in our development, which we believe that can bring them pride and increase interest in the game when the game is finished.

By applying the above principles, our customised agile development model was created as following: 1) High level designs/architectures, plans and requirements would be decided before

the development, which is called pre-development phase. 2) Sprint backlogs would be created according to the requirements collected in the pre-development phase, the project would then be pushed into loops of sprints. For each sprint, a survey would be designed for players to vote their preferred features. 3) After all the requirements are implemented, the development would go to the release phase and start an integration test (see Figure 4.16).

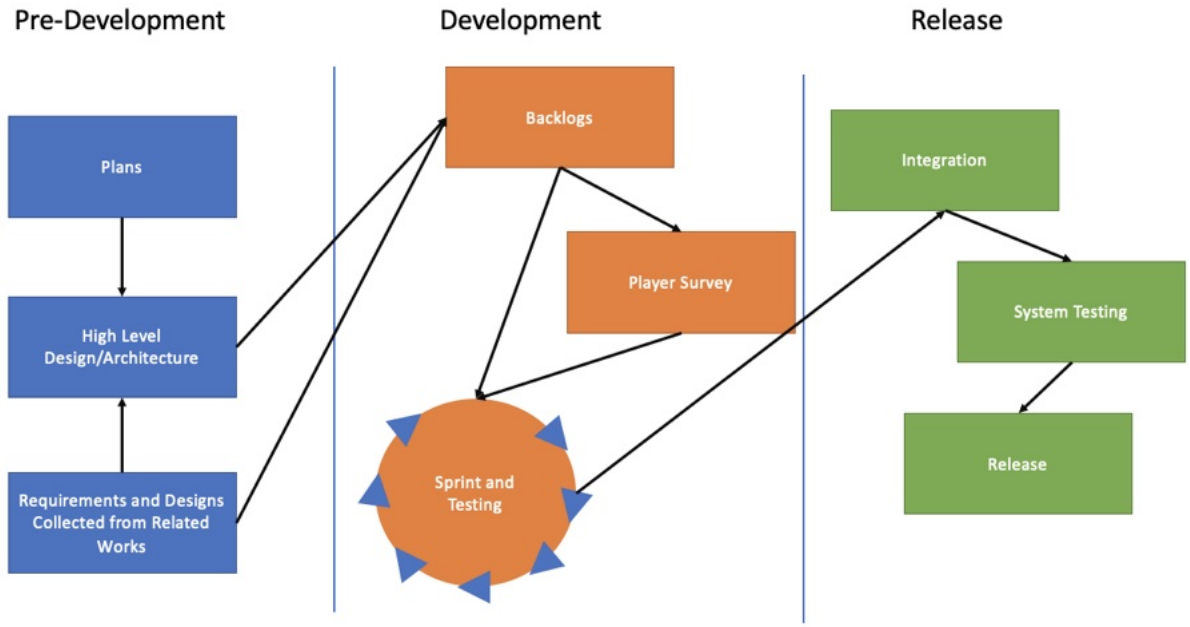


Figure 4.16: The Customised Agile Development Life Cycle for BreatheHero

4.4.2 Tools for Development

For guaranteeing the agile process and standardise the development actions, the following tools were used to assist the progress. These tools are very helpful and can be also used for reference in later development.

- **Agile Development Management - Jira:** Jira is a development process management tool for developers and especially used for agile development. It can help to create agile patterns and road-map for the project. In the development of BreatheHero, a road-map recorded all the sprint is created (see Appendix D D.2 Development Road-map). The tool greatly increased the efficiency of development, standardized the development process and enhanced trace-ability.
- **Version Control - GitHub:** GitHub was used as the version control tool for the project. All steps were in accordance with the version control plan (see Appendix C C.5 Version Control Plan).
- **Brainstorming - Miro:** Miro is a free online whiteboard which contains plenty of mind maps models. It can be easily accessed anytime which gave possibility for recording and updating design ideas and building architectures instantly.

Chapter 5

Implementation and Testing

As mentioned before, the development process would be organised with Sprint and a phased test would be included in each Sprint. In the release phase, an integration test would be set for guarantee the quality of the deliverable. All the development actions would strictly follow the plans above. At the time the development begin, the features including basic system (login, verification, data recording and user profile), character's actions (movement and breathe flame) and level pattern (huff-cough hurdles, finish-line and frozen coins) have been developed by Dr. Christof and packaged as reusable game objects. The development would start based on these basic features and the high-level design above. The section recorded the development process from Sprint 1 to Sprint5 and in each sprint several features were implemented. The development lasted for 3 months (1st/June - 1st/September) and each Sprint lasted about 2-3 weeks. A road-map for the whole development process of BreatheHero is attached in Appendix D D.2 evelopment Road-map for BreatheHero.

5.1 Start up

In the start up section, the logic for the basic features developed by Dr. Christof are going to be introduced for providing readers a better understanding of the project. The code and logic design were all developed by Dr. Christof and the source of the code will be noted.

5.1.1 Basic System

The main system was connected to Google Firebase, which provided us the database and server service. These part of code was implemented by Dr. Christof.

Login & Verification

The login system was designed with the following logic: 1) Get the email and password from the input box when they are given. When click the login button, the data would be sent to firebase for verification. 2) If register button is clicked, the data would be registered as new user if it doesn't appear in the database. 3) The system then would judge the role of the user (patients or Clinician) for login into different scene. 4) Any verification error would be noticed. When the verification is successful, a credential (Json file) would be created locally. The features were currently still under testing and the completed login.gd has been attached in Appendix I I.1 File: Login.gd.

```
# @author Dr. Christof
# When the login or register button pressed
func _on_LoginButton_pressed():
  Firebase.Auth.login_with_email_and_password(
    $VBoxContainer/GridContainer/Email.text,
    $VBoxContainer/GridContainer/Password.text)

# When the register button pressed
func _on_RegisterButton_pressed():
  Firebase.Auth.signup_with_email_and_password(
    $VBoxContainer/GridContainer/Email.text,
    $VBoxContainer/GridContainer/Password.text)

# All the errors would be noticed
func _on_FirebaseAuth_login_failed(code, message : String):
  print("login failed: %d %s" % [code, message])
  match message:
    "INVALID_EMAIL":
      $VBoxContainer/ErrorLabel.text = "Invalid email. Please
   correct it and try again."
          ...
    "WEAK_PASSWORD : Password should be at least 6 characters":
      $VBoxContainer/ErrorLabel.text = "Weak password. Please make
   it at least 6 characters long and try again."
    _:
      $VBoxContainer/ErrorLabel.text = message
```

Data Recording

For the clinician, they would be able to record or amend the patients' data at the clinician scene (see Figure 4.3). After the user successful login and matches the role of clinician, the scene would load the data of the patient selected from database. The clinicians would be able to amend and analyse the data recorded during daily exercise and then push the patients to the game. The feature was implemented by Dr. Christof and still under testing. The Clinician.gd has been attached in Appendix I I.2 File: Clinician.gd.

User Profile

In the game, except the user's information, some variables such as coins collected, achievements and hit points would need to be stored with the player. A User.gd file was created for this purpose. A dictionary structure was designed for storing and unified managing all the game information. This part was contributed by Dr. Christof and me. The completed User.gd is attached in Appendix I I.3 File: User.gd.

```
# @author Dr. Christof, Xiaoyan Zhou
# The dictionary structure for storing game state, another
   dictionary can be stored in as well.
var game_state : Dictionary = {
  "Achievements" : {
    "all_collect" : false,
    "woops": false,
    "go_away": false,
    "more_level": false,
    "boss_fight": false,
```

```
    "breath": false
},
    "coins" : 0,
    "time_left" : 600,
    "stats" : { "sideways_speed": 2.5, "forward_speed": 2.0,
   "fire_damage": 1.0 },
    "items" : { "FireUpgrade1": true, "PinkDragonSkin": true }
}
```

5.1.2 Character's Actions

The following character actions were implemented at the beginning by Dr. Chrsitof including the little dragon's movement and fire breath. This part is controlled by Player.gd and Controller.gd files.

Movement

There two main movements for the little dragon: moving forward constantly and moving left and right. The logic is giving the initial speed for the player and call the built-in method from godot for controlling the movement. Except using the controller, the system was designed to support keyboard as well (Arrow keys to move and space bar to control breath), which is for testing use.

```
# @author Dr. Christof
# Script in Player.gd
# Give initial speed for the player (forward speed and sideways
    speed)
export(float) var sideways_speed : float = 2.5
export(float) var forward_speed : float = 1.5
...
# move forward
move_and_slide(Vector3(Controller.movement * sideways_speed, 0,
   -forward_speed), Vector3(0, 1, 0))
```

```
# @author Dr. Christof
# Script in Controller.gd
# Enable the controller or the keyboard
func _process(delta):
  # input movement
  if Input.is_action_pressed("move_left"):
    movement = -1.0
  elif Input.is_action_pressed("move_right"):
    movement = +1.0
  else:
    movement = 0
...
```

Breathing Flame

Breathing flame is a basic and significant feature of the system. A fire effect would be called when the player breathes or pressures the space bar (testing use). The breathe length and pressure were monitored so that we can standardize the breathing exercises. A damage points was defined for the fire and all the frozen coins have a hit point. When the player breathes

long enough, the hit points of the ice will decrease by the damage points so that the ice would be melted. The Player.gd and Controller.gd has been attached in Appendix I I.4 Player.gd and Controller.gd.

```
# @author Dr. Christof
# Script in Player.gd
# give fire damage
export(float) var fire_damage : float = 1.0  # damage dealt per
    second
var fire_is_on : bool = false
...
# breathe fire
  if Controller.breath > 5 and not fire_is_on:
    User.game_state["Achievements"]["breath"] = true
    fire_is_on = true
    $Mouth/Fire.emitting = true
    $FireCollisionTimer.start()
    $FireSFX.play()

  if not Controller.breath > 5 and fire_is_on:
    fire_is_on = false
    $Mouth/Fire.emitting = false
    $FireCollisionTimer.stop()
    $FireSFX.stop()
  pass
```

```
# @author Dr. Christof
# Script in Controller.gd
# Enable the controller or the keyboard
func _process(delta):
...
  # input breath
  if Input.is_action_pressed("breathe_fire"):
    breath = 10
  else:
    breath = 0
```

5.1.3 Level Pattern

A level pattern with basic features including collecting coins, huff-cough hurdles and finish-line was created by Dr. Christof. And then amended by me for adding more feature during the development.

Coins

Pickup.gd is in charge of the part. It gives the settings of following: 1) the value of the coin. 2) If the coin is frozen (if frozen, call the ice model) and the hit point of the coin. 3) When the player enters the area of the coin, adds the value to the player's profile. The code is attached in Appendix I I.5 Pickup.gd.

Huff-cough Hurdle

The huff-cough Hurdle was used for forcing the player to cough and capturing the sound. The system works as following: 1) When the player enters the area of the hurdle, stops the

game and forces the player to cough three times to break the hurdle. 2) Capture the cough sound and continue when the player finishes coughing. The code is attached in Appendix I I.6 HuffCoughHurdle.gd.

Level

A Level.gd file contains the start and end functions of the game at the beginning. It initialised the game goal and the finish-line (as player enter the area, finishes the game). In the later development, features such as achievement controller, hit-point of the player and timer of the game etc. were added to this file by me as well. It can be seen as the main controller of the game process. The Level.gd is attached in Appendix I I.7 Level.gd and the features added in by me would be introduced one by one in the Sprints.

```
# @author Dr. Christof, Xioayan Zhou
func ready():
...
    # give the game goal at the beginning.
  if goal_coins > 0:
    $UI/Message.text = "Goal:\nCollect %d coins!" % goal_coins
  if goal_coins == 0:
    $UI/Message.text = "Goal:\nReach the finish line!"
  if goal_coins < 0:
    $UI/Message.text = "Goal:\nFight the boss!"
...
# Game's finish line function
func _on_FinishLine_body_entered(body):
  # check level completion
  if body.get_instance_id() == player.get_instance_id():
    var coins_collected : int = User.coins - start_coins
    ...
      $UI/Message.text = "Level complete!"
    else:
      $UI/Message.text = "Level incomplete:\nYou collected %d out of
   %d coins" % [coins_collected, goal_coins]
        ...
    pass
```

5.2 Sprint 1

5.2.1 Sprint 1 Overview

Starting from the Sprint 1, the sections would record the development process of the project features developed by me. After setting up the software and development tools, the Sprint was mainly designing the level 1 and achieving a runnable level with the basic features above. The Sprint would be organised as Sprint Overview, Backlog, Implemented Features and Tests and Review (Debugging section would be added if necessary.). Figure 5.1 is the road-map for Sprint 1, which can provide a better overview of the development of the stage.

Figure 5.1: Sprint 1 Road-map

5.2.2 Sprint 1 Backlog

The backlog is created by the choosing some of the requirements that collected in the Requirements Analysis section. The requirements were further broken down into functions here. Due to the Sprint was mainly for starting up the development (implementing basic functions), no survey for participants at the stage yet.

Table 5.1: Sprint 1 Backlog

Feature	**Description**	Priority
Environment Setup	A basic environment is required for player.	1
Move Forward	The player can move forward in the level.	1
Move Slide	The player can move left and right in the level.	1
Breathe Flame	The player can breathe fire for attack.	1
Collective Items	The player needs to have coins to collect.	1
Haff-cough Hurdles	The player needs to cough with haff-cough Hurdles.	1
Finish-line	The player can finish the game by finish-line.	1
Support Keyboard	The player can use keyboard for control as well.	1
Guidance and Notification	Guidance and notification need to be provided	2
Adjust game length	The game length need to fit the player.	2
Nicer Game Environment	A better environment is required.	2

5.2.3 Sprint 1 Implemented Features

In the Sprint 1, a basic environment of level 1 was implemented. The little red dragon was set at the start-line on a green ground. And by importing the player.gd and controller.gd, the player would be able to pan forward constantly and move slide as well as breathing flame. Haff-cough hurdles and coins sets were interspersedly arranged on the path. A finish-line was placed in a distance from the start-line and the game length was controlled in 2-3 minutes by adjusting such length. More decorative objects were added for a better gaming environment after implementing the basic functions. Code for guidance and notification was then added. Figure 5.2 is the overview of the completed Level 1.

Guidance and Notification

MessageArea.gd was added for giving guidance and notification when it is necessary. The logic is as following: a invisible object was set at the place required for a message for user in the level. When the player entered the area, the message would be shown for several seconds and disappear. The way for popping up messages has also used in other class when a notification needed. The code was implemented firstly by Dr. Christof and amend by me for a bug.

```
# @author Dr. Christof
extends Area

export(String) var message : String = ""
export(float) var duration : float = 4.0
...

onready var message_label : Label =
    get_tree().get_root().find_node("Message", true, false)

func _on_NotificationArea_body_entered(body):
  # Amended by Xiaoyan Zhou
  if body.get_instance_id() == player.get_instance_id():
    # if not message label available in scene tree, stop here
    if not message_label:
      return

    message_label.text = message
    message_label.visible = true

    # hide message after duration seconds
    yield(get_tree().create_timer(duration), "timeout")
    message_label.visible = false
  else:
    pass
```

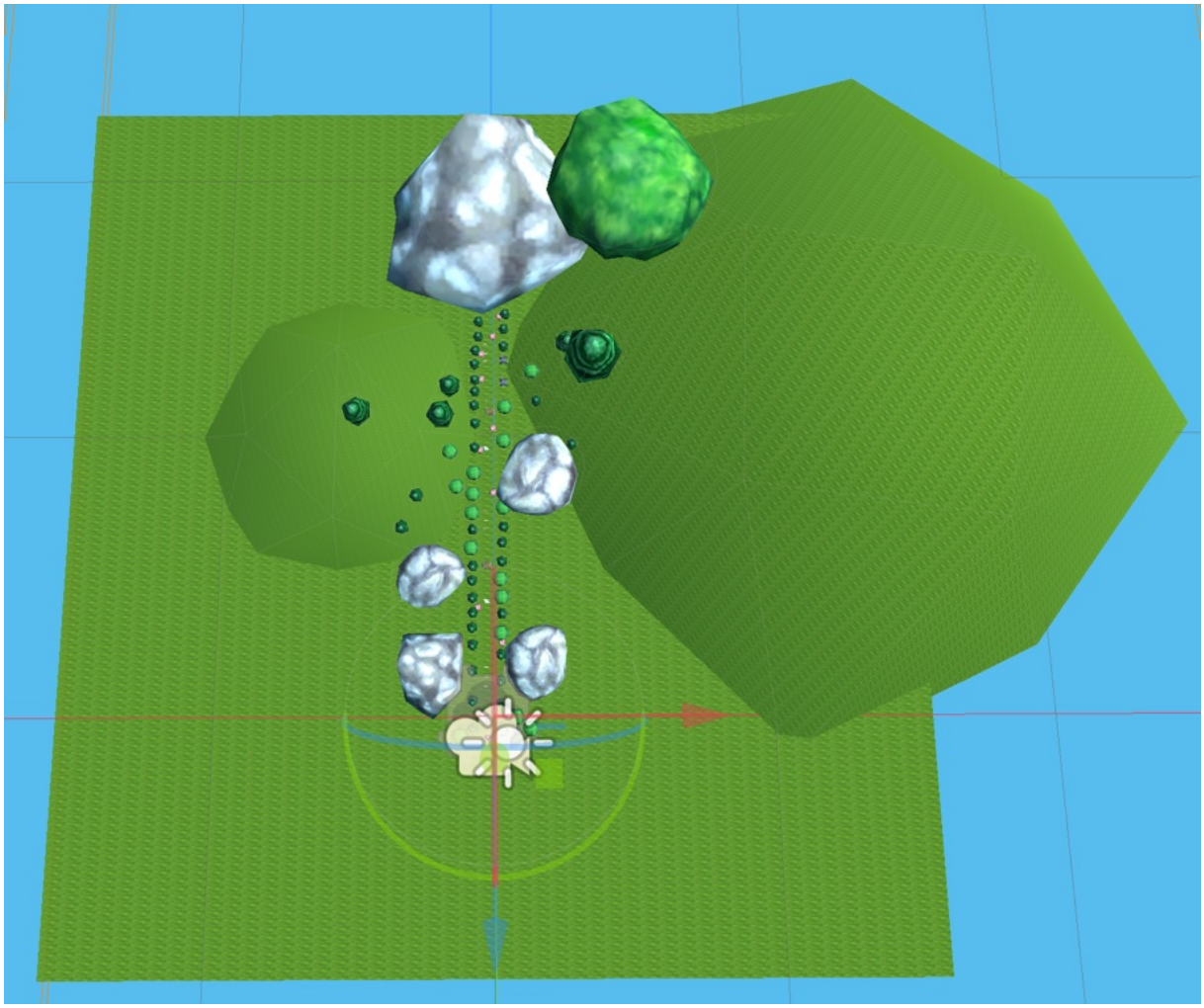


Figure 5.2: Completed Level 1 Overview

5.2.4 Sprint 1 Tests and Review

The section records the testing and review process for Sprint 1. The test cases were created according to the backlog. Bugs were detected during the testing and have been reported. The way for expectation handling can be found in Sprint 1 Debug section.

Tests Result and Review

The following table is the result of the phased test for Sprint 1. All cases were tested and reviewed. Two cases were failed which located as no restrictions on the code for specific body enter.

Table 5.2: Sprint 1 Test and Review Result

ID	1
Test Feature	Move Forward
Test Objective	Player
Description	The player can move forward correctly.
Result(Pass/Fail)	Pass
ID	2
Test Feature	Move Slide
Test Objective	Player
Description	The player can move slide correctly.
Result(Pass/Fail)	Pass
ID	3
Test Feature	Breathe Flame
Test Objective	Player
Description	The player can breathe flame correctly.
Result(Pass/Fail)	Pass
ID	4
Test Feature	Items Collection
Test Objective	Coins
Description	The player can collect the coins.
Result(Pass/Fail)	Pass
ID	5
Test Feature	Half-cough hurdles
Test Objective	Half-cough hurdles
Description	The player need to cough to pass the half-cough hurdles.
Result(Pass/Fail)	Pass
ID	6
Test Feature	Finish-line
Test Objective	Finish-line
Description	When the other objects enter the finish-line, the game over as well.
Result(Pass/Fail)	Fail
ID	7
Test Feature	Keyboard Supporting
Test Objective	Player
Description	The game can be controlled by keyboard
Result(Pass/Fail)	Pass
ID	8
Test Feature	Guidance and Notificaiton
Test Objective	Invisible message object
Description	Except the player, the messages would be called when the other objects entered.
Result(Pass/Fail)	Fail
Reviewed	Yes
Date	11/6/2021
Tester	Xiaoyan Zhou

Bug Report

Table 5.3: Sprint 1 Bug Report

ID	1
Bug Name	Finish-line
Reporter	Xiaoyan Zhou
Date	11/6/2021
Description	Any objects enter the finish-line lead a finish game.
Priority	1
Close Date	13/6/2021
Developer	Xiaoyan Zhou
ID	2
Bug Name	Message Area
Reporter	Xiaoyan Zhou
Date	11/6/2021
Description	Any objects enter the message area, the message would come out.
Priority	1
Close Date	13/6/2021
Developer	Xiaoyan Zhou

5.2.5 Sprint 1 Debug

According to the bugs reported, the following code was located which controlled the body enter of objects. The following piece of code was added which can judge the object's id. If the entered body is the player, then finish the game or show the message.

```
# Function for finish-line
func _on_FinishLine_body_entered(body):
    # the piece of code provided restriction for body enters, it would judge the object id.
  if body.get_instance_id() == player.get_instance_id():
    ...
  pass

# Function for message area
func _on_NotificationArea_body_entered(body):
  # Amended by Xiaoyan Zhou
  if body.get_instance_id() == player.get_instance_id():
  ...
  pass
```

5.3 Sprint 2

5.3.1 Sprint 2 Overview

In Sprint 2, level 2 was designed. With the idea of reusable code and objects, level 2 was easily implemented based on the functions and tests in Sprint 1. Longer game length was provided so that the player would need to breathe and cough more times for pass the level. Scene changing became one of the most urgent features required for connecting all the scenes together as a game loop. Timer were added for calculating the game length. Figure 5.3 is the road-map for Sprint 2 development.

Figure 5.3: Sprint 2 Road-map

5.3.2 Sprint 2 Backlog

Considering the game was not ready for collecting design opinions from the public yet, no survey has started. Basic features were still needed to be developed.

Table 5.4: Sprint 2 Backlog

Feature	**Description**	Priority
Level 2 Design and Implementation	The level needs to be created according to the level template.	1
Scene Changing	All the scenes need to be connected together.	1
Timer	A timer is needed for calculating the game length.	2

5.3.3 Sprint 2 Implemented Features

By applying the level template and with the experience from designing level 1, the development of level 2 became smooth and efficient. The level environment and basic features were implemented by importing the patterns. Timer and scene changing were then developed after finishing the level design. Figure 5.4 is the overview of level 2.

Timer

Unfortunately, there is no built-in timer function in godot and the feature was implemented by adding increasing function the the _ready method, which would update itself every second. The counter would stop when the game is over and restart from 0 when a new level starts. The code was then added into the level.gd file.

```
func _ready():
...
    # The function for timer, update each second.
  while true:
    yield(get_tree().create_timer(1.0), "timeout")
    $UI/TimerCounter.text = "Timer: %d" % timer
    timer += 1
  pass
```

Scene Changing

Godot has the build-in scene changing functions, the thing we need to consider is how to add them to the right place. The specific scene changing design has been introduced in the Design section and the following is the code for implementing it. The method has been added to all the place it is needed.

```
func _():
...
    # The code for scene changing.
    get_tree().change_scene(chapter_filename)
...
```

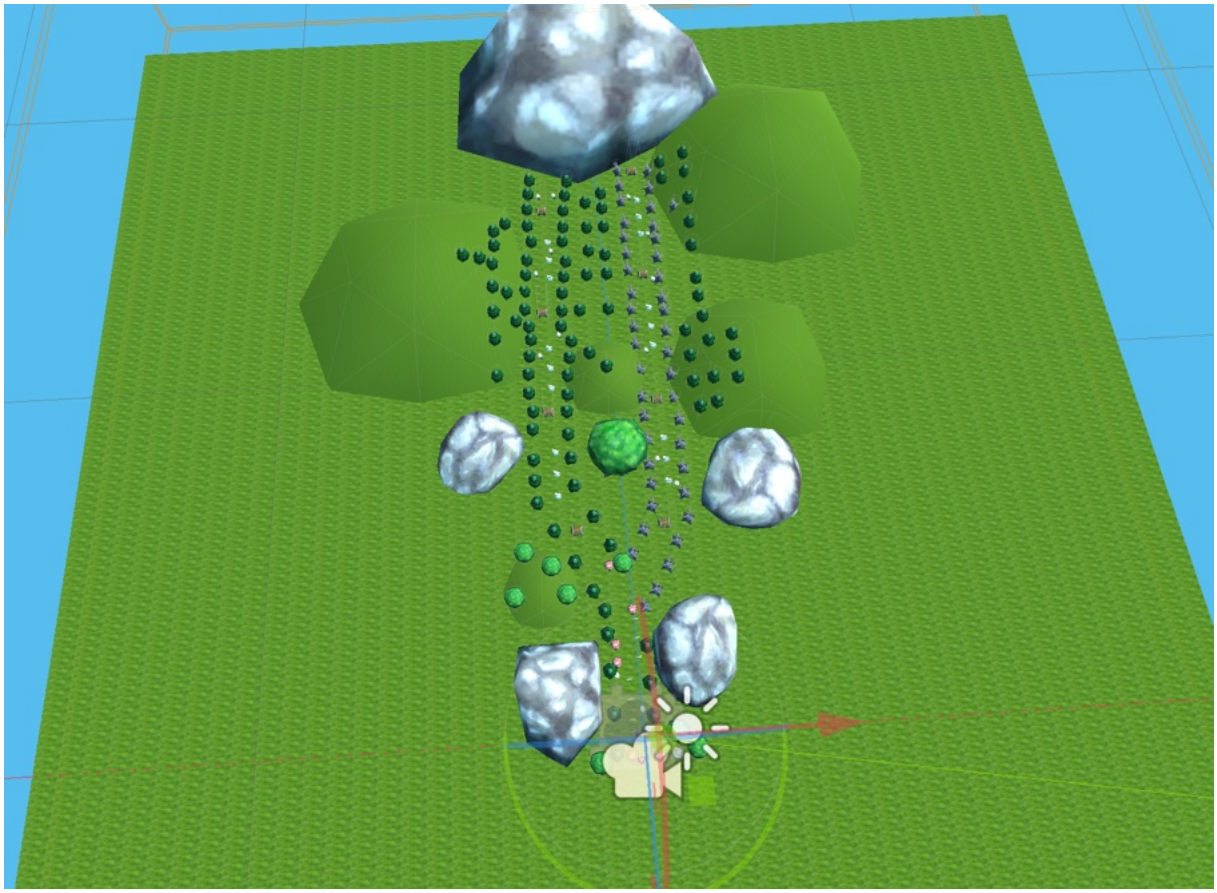


Figure 5.4: Completed Level 2 Overview

5.3.4 Sprint 2 Tests and Review

All test cases have been tested and reviewed. No bugs were detected.

Tests Result and Review

Table 5.5: Sprint 2 Test and Review Result

ID	1
Test Feature	Scene Changing
Test Objective	Scenes
Description	The scenes can be linked and change correctly.
Result(Pass/Fail)	Pass
ID	2
Test Feature	Timer
Test Objective	Timer
Description	The timer can work correctly counting the game length.
Result(Pass/Fail)	Pass
Reviewed	Yes
Date	26/6/2021
Tester	Xiaoyan Zhou

5.4 Sprint 3

5.4.1 Sprint 3 Overview

In Sprint 3, level 3 was implemented and we started to gather design ideas from the public. The survey design and result would be discussed in the section and the following features was developed during the Sprint: Storyline and Slime. Figure 5.5 is the road-map of Sprint 3.

Sprint 3 Participant Survey

Starting from Sprint 3, participant surveys were designed for letting the players to decide what they want and at the same time, we started to keep in touch with our participants, report on the development process and even invite them to join in the development. In this case, we hope we can bring the participants a sense of identity and pride, and let more people pay attention to the children with pulmonary fibrosis. The survey was designed as easy, readable and fun. Several features were chosen from the requirements and asked the participants to vote what they want. In Sprint 3 participant survey 33 people joined in, story-line(94%), battle-system(65%), various props(23%) and shop(13%) were chosen to vote and story-line and battle-system were decided to be developed. A receipt was then sent to all participants after we finished developing for an update and better understanding for our project. The questionnaires, result and receipt for participants have been attached in Appendix D D.3 Participant Survey, Result and Receipt.

Figure 5.5: Sprint 3 Road-map

5.4.2 Sprint 3 Backlog

The backlog was created by collecting basic features from requirements and according to the vote result of the participant survey.

Table 5.6: Sprint 3 Backlog

Feature	**Description**	Priority
Level 3 Design and Implementation	The level needs to be created according to the level template.	1
Stories	A good story is needed.	1
Story Scenes	The scenes for each story.	1
Story Telling System	A controller and a way for telling the story is needed.	1
Slime	Enemies are required for battle system.	2
Slime's Movement	Slime can move around in the level.	2

5.4.3 Sprint 3 Implemented Features

After creating a bsic environment of level 3 (see Figure 5.7), the new features started to be developed. Slime were added to the level as enemies and the player need to avoid touch it or would be caught and lead to a game over. A set of storytelling system including dialogic system, scene changing and controllers etc. was added and the story was presented as dialog.

Story Telling System

By importing Dialogic package (a dialogic tool developed for godot), a basic pop-up dialogic scene (see Figure 5.6) was created and the story script was divided into several scenes. The chapters were then inserted between levels with scene changing method. Controllers were written for controlling the background music and the animation.

```
# Class for controller the background music.
# @author Dr. Christof
extends Spatial
func _ready():
  BackgroundMusic.stream = music
  BackgroundMusic.play()

# Class for controlling animation.
# @author: Xiaoyan Zhou
extends Camera
var appear : String = ""
...
# if the dialogic gives the signal, set the slime visiable
func _on_DialogNode_dialogic_signal(value):
  appear = value
  pass
```

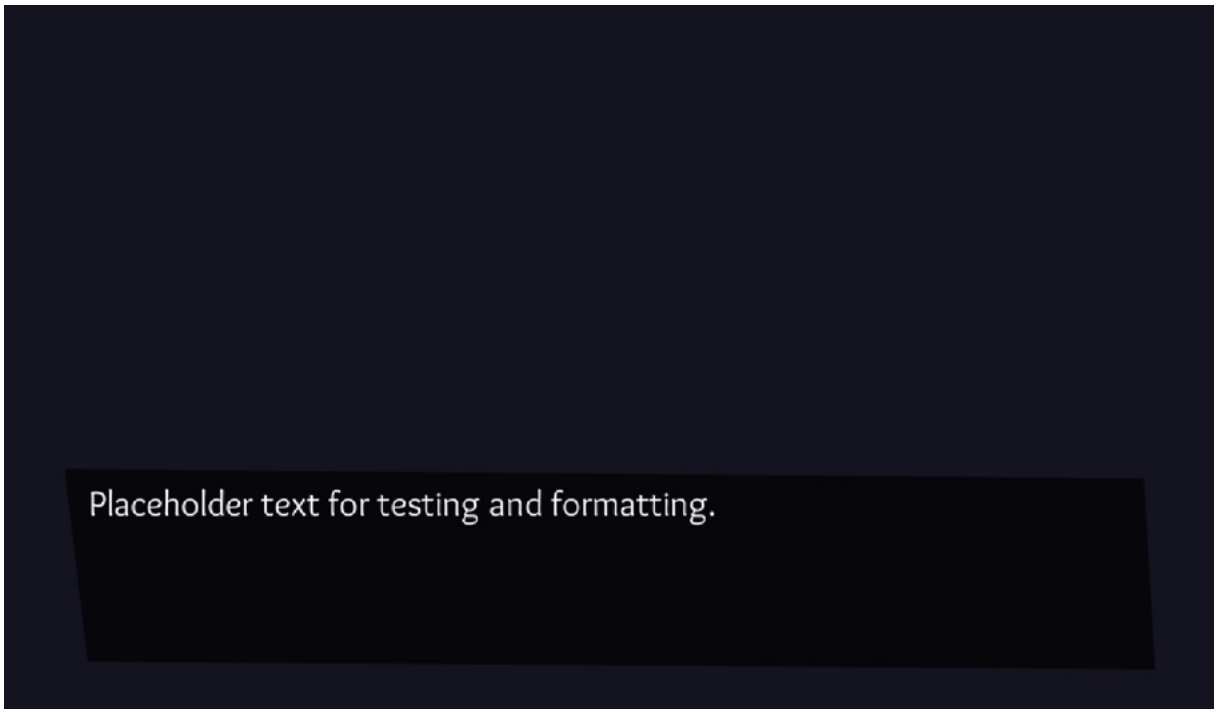


Figure 5.6: Dialogic Templete Provided by Dialogic Package

Slime

Enemies were going to be added since the requirement of battle system. The little slime has been designed to move left and right in the level and give feedback (At beginning it would lead to game over and after the hp system was added, reduce one hit point for the player.). The logic is as following: 1) If slime touches objects when move left, slime change the movement to move right. Same logic when it is moving right. 2) If the object is the player, reduce the hit point. If the hit point becomes 0, game over. The completed file LittleSlime.gd has been attached to Appendix I I.8 LittleSlime.gd.

```
# @author Xiaoyan Zhou
# Controlling the slime to move left and right
func _move_left_right():
  if left:
    move_and_slide(Vector3(-4,0,0), Vector3(0,0,0), false, 1)
    if is_on_wall():
      _game_over()

      move_and_slide(Vector3(4,0,0), Vector3(0,0,0), false, 1)
      left = false
      right = true

  if right:
    move_and_slide(Vector3(4,0,0), Vector3(0,0,0), false, 1)
    if is_on_wall():
      _game_over()

      move_and_slide(Vector3(-4,0,0), Vector3(0,0,0), false, 1)
      left = true
      right = false

# Reduce hit point if the player touches the slime and if the hp is
    0, game over.
func _game_over():
  if get_slide_collision(0).get_collider_id() ==
    player.get_instance_id():
    User.hit_points -= 1
    if User.hit_points == 0:
      ...
    #yield(get_tree().create_timer(1.0), "timeout")
    #get_tree().change_scene("res://Game/GameOver.tscn")
  else:
    pass
```

Figure 5.7: Completed Level 3 Overview

5.4.4 Sprint 3 Tests and Review

Table 5.7: Sprint 3 Test and Review Result

ID	1
Test Feature	Story Telling System
Test Objective	Dialog
Description	The dialog can show correctly.
Result(Pass/Fail)	Pass
ID	2
Test Feature	Story Scene Changing
Test Objective	Scenes
Description	The story scenes are connected correctly.
Result(Pass/Fail)	Pass
ID	3
Test Feature	Slime's Movement
Test Objective	Slime
Description	The slime moves left and right correctly.
Result(Pass/Fail)	Pass
ID	4
Test Feature	Damage and Game Over
Test Objective	Slime
Description	The slime can hit player and lead to a game over.
Result(Pass/Fail)	Pass
Reviewed	Yes
Date	11/7/2021
Tester	Xiaoyan Zhou

5.5 Sprint 4

5.5.1 Sprint 4 Overview

Level 4 was finished in Sprint 4 and another participant survey was organised. HP system was then added to the game. According to the result of the participant survey, achievement system was implemented and we planed to adding level rewards in level 5 boss fight. Figure 5.8 is the road-map for Sprint 4.

Sprint 4 Participant Survey

The questionnaires, result and receipt for participants have been attached in Appendix D D.3 Participant Survey, Result and Receipt. During the survey, the following features has been chosen for vote: Level Rewards(65%), Achievement System(65%), Various Props(45%) and Shop(45%). The achievement system and level rewards were chosen.

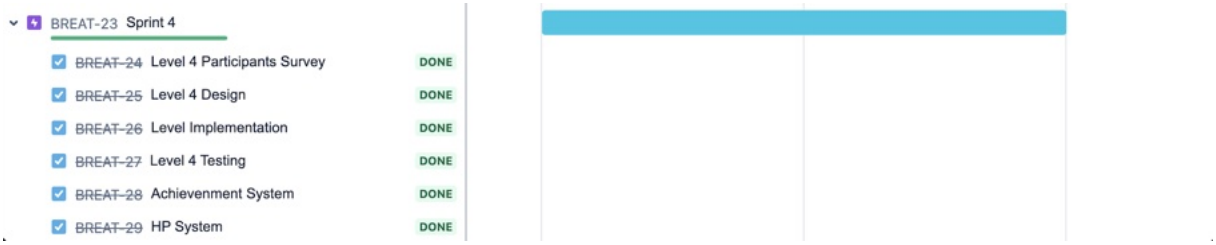


Figure 5.8: Sprint 4 Road-map

5.5.2 Sprint 4 Backlog

Table 5.8: Sprint 4 Backlog

Feature	Description	Priority
Level 4 Design and Implementation	A basic environment is required for player.	1
Achievement System	The achievement would be triggered when player finishes some movement.	1
Hit Point	The player needs to have hit point instead of game over directly.	1

5.5.3 Sprint 4 Implemented Features

The basic environment was completed (see Figure 5.9) and the following features were implemented: Achievement System and Hit Point.

Achievement System

A scene for achievement has been designed and several movements such as first time to collect all coins in a level, first time to breathe fire and first time to be caught by slime etc. have been designed as the triggers for achievements. The achievements information were stored in the user's profile (User.gd) and the triggers (code) were set at the block of these movements (for example, the achievement "first time to breathe"'s code is at the code block control the breath). A Achievement.gd is used as the controller of the system. The achievements were

set invisiable at the beginning and once the player triggered them, they would be unlocked. We hope the system can bring them fun and surprise. The contents of the achievements were designed as positive and inspired. The Achievement.gd has been attached at Appendix I.9 Achievement.gd.

```
func _achievements_check():
  # func for controlling the achievements' visibility
  if User.game_state["Achievements"]["all_collect"]:
    get_node("VBoxContainer/GridContainer/AllCollect").visible = true
...
  if User.game_state["Achievements"]["breath"]:
    get_node("VBoxContainer/GridContainer/Breath").visible = true


...
# The code for the all collect achievements, such code was
    everywhere when they are needed.
if coins_collected >= goal_coins:
      User.game_state["Achievements"]["more_level"] =  true
...
```

Hit Point

Hit Point has been set to the User.gd and initially as 3. The heart on left corner of the game would present as the hp left for the player. The code and logic is as following.

```
# hit point, initially as 3
var hit_points : int = 3
...
# get hp from profile, if hp is 0, game over. Reset when a new level
    starts.
func _get_hp():
  if User.hit_points == 3:
    pass
  elif User.hit_points == 2:
    $UI/GridContainer/hp3.visible = false
  elif User.hit_points == 1:
    $UI/GridContainer/hp2.visible = false
  elif User.hit_points == 0:
    $UI/GridContainer/hp.visible = false
    player.stopped = true
    $UI/Message.text = "Oh No!\nYou have caught by the Slime!"
    $UI/Message.visible = true
    player.stopped = true
    yield(get_tree().create_timer(1.0), "timeout")
    get_tree().change_scene("res://Game/GameOver.tscn")
  pass
```

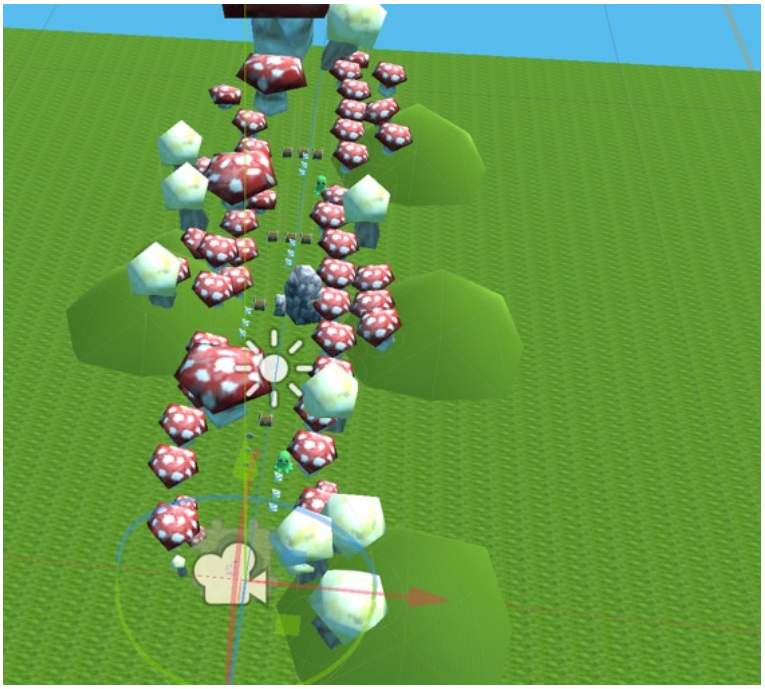


Figure 5.9: Completed Level 4 Overview

5.5.4 Sprint 4 Tests and Review

Table 5.9: Sprint 4 Test and Review Result

ID	1
Test Feature	Achievements System
Test Objective	System
Description	The achievements triggered correctly.
Result(Pass/Fail)	Pass
ID	2
Test Feature	HP System
Test Objective	Player
Description	The hp system works correctly.
Result(Pass/Fail)	Pass
Reviewed	Yes
Date	27/7/2021
Tester	Xiaoyan Zhou

5.6 Sprint 5

5.6.1 Sprint 5 Overview

A final level was designed and set as the boss fight. No participant survey was designed but we started to invite participants for field study. For the boss slime, it can move left and right and at the same time generates little slime to hit the player. The player was set as mentioned in Design section and needs to fight back the boss slime by breathing fire. Huge rewards (coins rain) were set when the player wins the boss fight. Figure 5.10 is the road-map for Sprint 5.

Figure 5.10: Sprint 5 Road-map

5.6.2 Sprint 5 Backlog

Table 5.10: Sprint 5 Backlog

Feature	Description	Priority
Level 5 Design and Implementation	A basic environment is required for player.	1
Boss Slime	The boss slime can move left and right.	1
Generating Little Slime	The huge slime can generate little slime	1
Reward	Rewards required after the boss fight.	1

5.6.3 Sprint 5 Implemented Features

The Sprint was mainly implementing the boss fight and reward after the boss fight. A basic environment was designed for the level (see Figure 5.11) as the battle field.

Boss Fight

The boss fight can be divided into three main parts: 1) Boss slime's movement. 2) Boss slime generating little slime to hit the player. 3) Player breathes to fight boss slime back. These parts were controlled by files BossSlime.gd and LittleSlime_bossfight.gd (see Appendix I I.10).

For boss slime's movement, similar code as little slime's movement was applied. And for generating the little slime, the little slime model was loaded at the beginning of the scene and for each 10 seconds, a new little slime node would be generated. The little slime would disappear when it passes the player or missing the boss slime (saving usage).

```
# move left and right, same logic as the other slime
func _move_left_right():
  if left:
...

  if right:
```

```
...
# generate a little slime in the scene
func _generate_little_slime():
  var slime = LittleSlime_bossfight.instance()
  get_parent().add_child(slime)
  var parent_position = get("translation")
  slime.set("translation", Vector3(parent_position.x,
   parent_position.y-1.5, parent_position.z+5))
  pass

func _on_Generate_littleSlime_timeout():
  _generate_little_slime()
  pass # Replace with function body.
```

The hit points were set for the little slime generated, when the player breathe, the hp would reduce. When the hit point is 0, the little slime would spin and player can cough to shoot it back to hit boss slime (see Appendix I I.10 LittleSlime_bossfight.gd).

Reward

Reward would be generated when the player wins the boss battle. The coins were set as invisible at first and appear when the boss fight is over. The logic is as following.

```
# generate coins
func _generate_coins():
  pick_up.visible = true
  pass
```

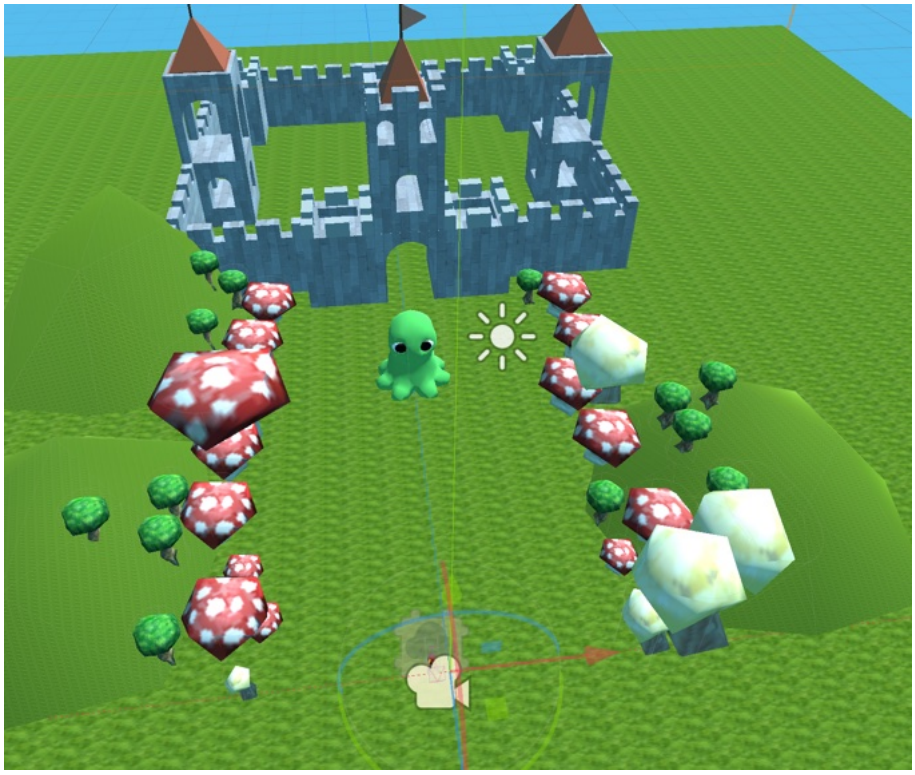


Figure 5.11: Completed Level 5 Overview

5.6.4 Sprint 5 Tests and Review

Table 5.11: Sprint 5 Test and Review Result

ID	1
Test Feature	Boss's Movement
Test Objective	Boss
Description	The boss move correctly.
Result(Pass/Fail)	Pass
ID	2
Test Feature	Little Slime Generating
Test Objective	Boss
Description	Little slime generated correctly.
Result(Pass/Fail)	Pass
ID	3
Test Feature	Reward
Test Objective	System
Description	Reward was generated.
Result(Pass/Fail)	Pass
ID	4
Test Feature	Boss Fight
Test Objective	System
Description	The battle system works.
Result(Pass/Fail)	Pass
Reviewed	Yes
Date	1/9/2021
Tester	Xiaoyan Zhou

5.7 Release

In the Release section, an integrated test has been done for testing the system's runnablity, robustness and the connections between different components. By passing the integration test, the completed game circle mentioned in the Design section was completed and the deliverable prototype was ready for field study. The game was then deployed to google firebase with godot exporting a web version by Dr. Christof.

Table 5.12: Integration Test and Review

ID	1
Test Feature	Level 1 Design and Implementation
Test Objective	System
Description	The level functions work fine.
Result(Pass/Fail)	Pass
ID	2
Test Feature	Guidance and Notificaiton
Test Objective	Invisible message object
Description	The message object comes out correctly.
Result(Pass/Fail)	Pass
ID	3
Test Feature	Level 2 Design and Implementation
Test Objective	System
Description	The level functions work fine.
Result(Pass/Fail)	Pass
ID	4
Test Feature	Scene Changing
Test Objective	Scenes
Description	The scenes can be linked and change correctly.
Result(Pass/Fail)	Pass
ID	5
Test Feature	Timer
Test Objective	Timer
Description	The timer can work correctly counting the game length.
Result(Pass/Fail)	Pass

ID	6
Test Feature	Level 3 Design and Implementation
Test Objective	System
Description	The level functions work fine.
Result(Pass/Fail)	Pass
ID	7
Test Feature	Story Telling System
Test Objective	Dialog
Description	The dialog can show correctly.
Result(Pass/Fail)	Pass
ID	8
Test Feature	Story Scene Changing
Test Objective	Scenes
Description	The story scenes are connected correctly.
Result(Pass/Fail)	Pass
ID	9
Test Feature	Slime's Movement
Test Objective	Slime
Description	The slime moves left and right correctly.
Result(Pass/Fail)	Pass
ID	10
Test Feature	Damage and Game Over
Test Objective	Slime
Description	The slime can hit player and lead to a game over.
Result(Pass/Fail)	Pass
ID	11
Test Feature	Level 4 Design and Implementation
Test Objective	System
Description	The level functions work fine.
Result(Pass/Fail)	Pass
ID	12
Test Feature	Achievements System
Test Objective	System
Description	The achievements triggered correctly.
Result(Pass/Fail)	Pass
ID	13
Test Feature	HP System
Test Objective	Player
Description	The hp system works correctly.
Result(Pass/Fail)	Pass

ID	14
Test Feature	Level 5 Design and Implementation
Test Objective	System
Description	The level works correctly.
Result(Pass/Fail)	Pass
ID	15
Test Feature	Boss's Movement
Test Objective	Boss
Description	The boss move correctly.
Result(Pass/Fail)	Pass
ID	16
Test Feature	Little Slime Generating
Test Objective	Boss
Description	Little slime generated correctly.
Result(Pass/Fail)	Pass
ID	17
Test Feature	Reward
Test Objective	System
Description	Reward was generated.
Result(Pass/Fail)	Pass
ID	18
Test Feature	Boss Fight
Test Objective	System
Description	The battle system works.
Result(Pass/Fail)	Pass
ID	19
Test Feature	Web Deploy
Test Objective	System
Description	The web version works fine.
Result(Pass/Fail)	Pass
Reviewed	Yes
Date	1/9/2021
Tester	Xiaoyan Zhou

Chapter 6

Field Study

The section would give a discussion of the field study. It would conclude an introduction of the study design, participants, data collection and limitations of the field study. The original plan was inviting young patients to participate in the trial but due to the effect of covid-19, only healthy participants were invoked. The experiment was still going well by simulating the breathing exercises, but it also brings us more errors and limitations. All the forms and questionnaires for the study have been attached in Appendix E, F and G (Participants Information Sheets, Consent Form and Questionnaires).

6.1 Study Design

The field study was designed into two parts: 1) A daily trial last for 7 days and 10 minutes game daily by simulating the breathing exercise. A daily questionnaire (see Appendix G Daily Questionnaire) is asked to be fill after the 10 minutes treatment session. This section is for collecting treatment effects information and the game's daily entertainment score for drawing the score change curve (We hope the game can provide the players longer-term interests). 2) After the participants finishing all the daily trial, a system evaluation is required. The following information would be collected including participants' personal information and game history etc. and they would need to score the system's features, usability and motivation inventory (see Appendix G Section I - IV).

Since a web version is available, the questionnaires were separated to the participants and they were asked to play the game according to their schedules. All the sessions were done asynchronously. A verbally briefed session for details of the research was given for all the participants so that they can have a basic understanding of the system but not overdo since the system's usability were need to be tested. Technical assistance was provided when necessary for setting up the game but not during the system under testing. Participants were asked to explore the system freely.

During the field study, the daily questionnaire was asked to be filled once they finish the 10 minutes game session, the information collected daily including the completeness of the session, levels completed, approximately average breath times and lengths, approximately cough times and the entertainment score for today's session. After seven days when the last daily session is completed, a system evaluation questionnaires is required to be filled according to the whole gaming experience. The following data would be collected: participants' background

information and game history (for further development use), game feature score and feedback, system's usability score and a motivation inventory.

The field study used similar evaluation models as previous research by TO (2020) and added long-term experiment, treatment effect evaluation, long-term evaluation module and content evaluation module, which provides a chance for comparing the results, complementing each other and refining experience for later development.

6.2 Participants

As mentioned, healthy participants instead of young patients were invoked for the research. Five healthy participants with age range from 23 -25 (with a mean age 24, the ratio of male to female is approximately 1:1) were involved in the research. One of them had the cough variant asthma in the young age and left were all healthy without any breathing/lung disease history. Four of the participants have over 5 years gaming history and only one was answered as almost none. The participants preferred a variety of game devices, covering the range from the consoles to the mobiles. None of them has the experience for gamified physiotherapy games. The type of game preferred and how to play are quite different due to the different appreciating habits of participants, but role-playing games and the collection of achievements are more attractive to players from a narrow perspective.

6.3 Data Collection

The questionnaires were separated to the participants by the researcher. Since we also hope to test the portability and flexibility of the system brought by the online version, the questionnaires will be completed by the participants themselves every day after distribution, and will be collected and sorted by the researchers after seven days.

6.4 Limitation

Since the field study was taken place between the healthy participants. The breathing exercise can only be simulated, which has brought huge limitations and a great error. The respiration and effect measurement may be extremely inaccurate since the participants were not familiar with the correct steps for breathing exercises. At the same time, the breathing strength and pressure of healthy people are also very different from the young patients. In this case, only the number and length of breaths have a certain reference value.

Another point is that, the age of the test group is different from the target group (the age of the test group is higher than the target group). Over-simplified game mechanics are not suitable for participants with many years of playing experience and complex game concepts, which could lead to a scoring bias for the system.

And finally, the number of participants in the experiment is too small and the results are not universal. We hope to conduct more in-depth and universal tests on the system in the future.

Chapter 7

Result

Excepting the participant's information analysis (discussed in Participant section above), the results are going to be evaluated by two parts: 1) Daily Evaluation (including daily treatment effect and entertainment effect) 2) System Evaluation (including game features score, system usability scale (SUS) and Intrinsic Motivation Inventory (IMI)). The analysed data has been attached in Appendix H and full data record has been uploaded with the dissertation as a corpus.

7.1 Daily Evaluation

The section would evaluate the result with the data collected by the daily questionnaire part, which is about the treatment effect and the daily entertainment level provided by the 10 minutes session daily. These results can be part of the strong evidence for answering RQ3 (Does the project actually help the young patients adhere their treatment?).

7.1.1 Daily Treatment Effect

This part of data was collected by the participants to simulating the breathing exercise. As designed, there were seven set of the data and each set contains the following information: 1) The daily completeness of the session. 2) Game levels completed in the session. 3) Total breath number (simulated). 4) Average breath length (simulated). 5) Total cough number (simulated). 5) Entertainment score. According to the results, all the participants were able to complete their 7 daily sessions. From overall view, the average number of the daily levels completed is 2.89 with a standard deviation 0.72 (range from 2.14 to 3.43), which means a participants did 2.14 levels at least and 3.43 levels at most on average, this difference may be caused by game habits and rhythm. On average, (14.97 ± 4.76) times of breath have been simulated daily and for each breath has an average breath length of (4.89 ± 1.02) seconds. An average cough times as (43.29 ± 13.72) has been recorded, this drop value is caused by the different levels of players playing daily. Although this set of data is obtained by simulation, this still allows us to have a further understanding of our system and has certain reference value for clinical treatment plans. Despite the lack of key data (such as breathing pressure and strength) and not as effective as Oikonomou et al. (2014)'s system, we still achieved positive results at the level of breathing rhythm control, breathing exercise norms and motivating breathing exercises.

7.1.2 Daily Entertainment Effect

The participants were asked to score their gaming experience from 0-5, an average daily entertainment score of (3.57 $\pm$ 0.74) has been recorded. By observing the scoring curve (see Figure 7.1), the daily entertainment score shows a downward or flat trend overall. This is caused by the loss of interest after players lose their freshness and repeat play. After an interview, the rare rebound trend was caused by the boss battle in level 5. On the whole, the system provides players with a longer-lasting entertainment level and freshness, but the loss of interest is still inevitable. Interspersed with novel gameplay helps to bring a brighter feeling to players.

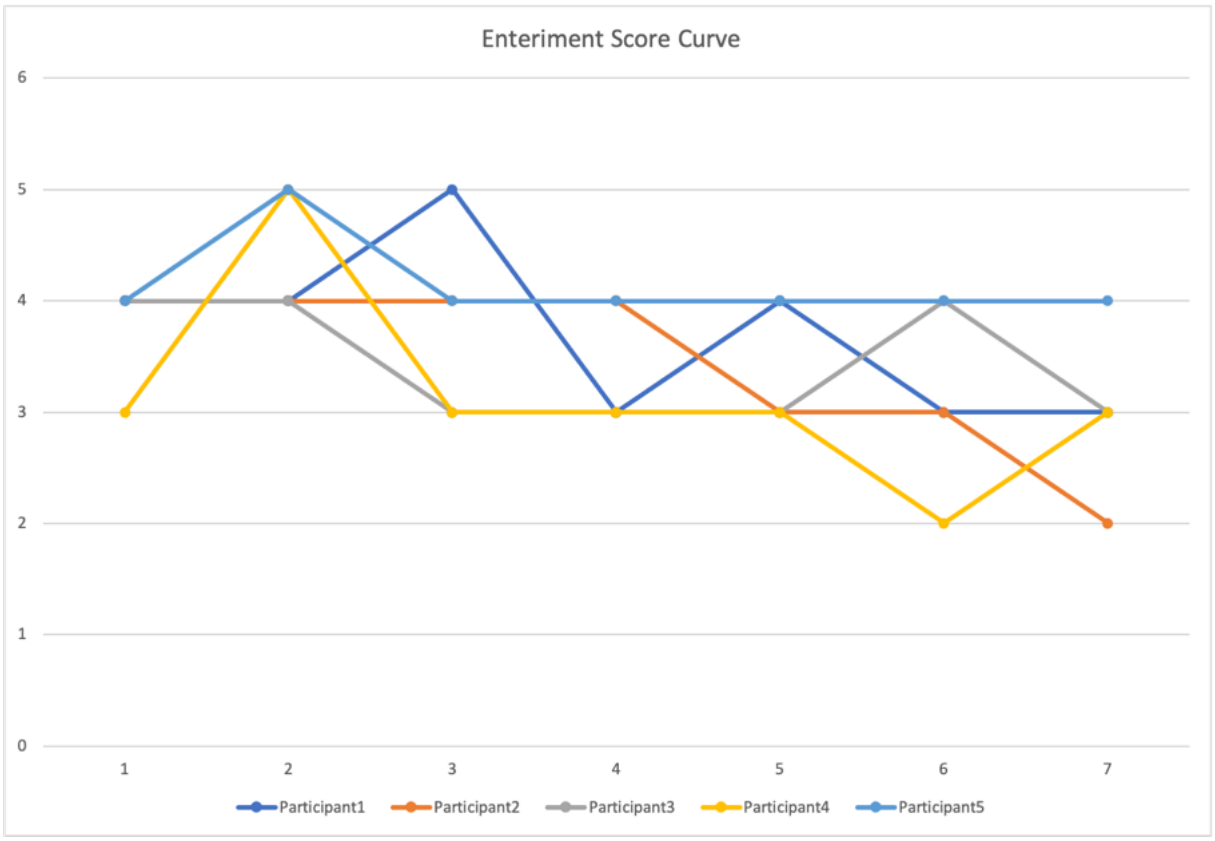


Figure 7.1: The Entertainment Score Curve

7.2 System Evaluation

The section is going to evaluate the results on a system level. Firstly, the game features would be scored and feedback are going to be collected. Then the system's usability would be evaluated by SUS and finally intrinsic motivation inventory would be evaluated.

7.2.1 Game Feature and Feedback

Participants were asked to score the enjoyment level (from 1-5) of the system by its components and score the whole system after the whole game play. For specific scoring items, see Appendix G Section II. An average feature score (4.11/5 $\pm$ 0.81) has been given and system average score is (81.84/100 $\pm$ 9.56). This is quite positive since entertainment level of both indexes is

very high. And according to the results, the boss battle and story-line have become two of the most enjoyable components of the system and the game's control module still needs to be polished.

Valuable suggestions and feedback have been collected from the open questions. 1) For all the participants have joined in the research from the development survey in Sprint 3 and 4 (2 out of 5) have given this kind of development and interaction mode a very high evaluation. 2) All participants have experienced the whole game circle and the boss battle component (5 out of 5) have been voted as the most impressive one. 3) The story line is evaluated as clear and simple, very suitable for children and positive. Everyone seems looks forward to seeing the further development (5 out of 5). 4) The achievement system is great and provides players great senses of fun, pride and motivation. 5) Comparing to a single features, the complete game circle has brought players even better gaming experiences. 6) The web version has brought great convenience but it took too long to load. The issue has been reported and would be optimized in further work. 7) Feedback has collected and including adding an overview of the map, adding more instructions, rewards after opening the obstacles and adding more completed control features (such as restart, pause and quit etc.) etc. The feedback has been categorized and organized, and could play a guiding role in future development.

7.2.2 System Usability Scale (SUS)

SUS is a quick and reliable tool for measuring a system's usability (Brooke et al., 1996). In the SUS, the participants were asked to score the usability of the system components (from 1 -5) and show their understanding of the game objects (see Appendix G Section III). In SUS, generally, a score above 60 will be regarded as a product with good usability. By calculating the SUS score of the system, the result was given as 63.2/100 and with a standard deviation of 7.29. This is related to participants' understanding of the system but overall the system is easy to use. Everyone showed a good understanding of the objects of the game.

7.2.3 Intrinsic Motivation Inventory (IMI)

IMI is a measures model for scaling participants' subjective experience (Needs, 2021). In our case, we hope to measure the entertainment level and pressure level for the game play to ensure that the system can provide participants enough sense of enjoyment and pressure for continuing play. The section has two parts, a table for scoring (1-5) the enjoyment level of the objects and a table for scoring (1-5) the pressure level of the objects (see Appendix G Section IV). The feedback gave a result as the enjoyment average score (69.6/100 $\pm$ 9.21) and the pressure average score (52.61 $\pm$ 19.04). The pressure level provided during the game play is bit low, which could make the players feel burnout during the game. The situation is considered to be improved in further development. The enjoyment level provided by the system is stable which can be a strong evidence for answering RQ3.

Chapter 8

Discussion

We have proposed three research questions for the project and by studying the related works, designing our own prototypes, implementing the features and field study, we are able to answering the questions.

- **RQ1:** How to build game features that can motivate young patients to adhere their breathing exercises?
- **RQ2:** How to make the game more sustainable to provide such motivation?
- **RQ3:** Does the project actually help the young patients adhere their treatment?

For answering RQ1 and RQ2, quantities of related researches have been analysed and discussed in Chapter 2. We have understood the pathology of pulmonary fibrosis, mechanism of breathing exercise, the gamification principles and an overview of prototypes for GCP in previous works. By understanding the mechanism behind and guiding by the instructive principles, we were able to refine the features we need with the help of previous GCP researches in Chapter 3. We assumed the requirements we gathered were the answers for RQ1 and 2.

In Chapter 4, a prototype with a complete game circle for BreatheHero has been designed. The system is driven by CAR loop and assisted with various sub-systems for achieving a better operating result and so that the prototype can be developed and a filed study can be processed for proving our assumptions.

In Chapter 5, an industrial development life cycle has been applied to the project for guaranteeing the quality and deliverable of the product. An comprehensive code overview has been given, which is of great significance to the follow-up development. The development has last for 5 Sprints and a series of features have been implemented.

And finally, a filed study has taken place and results were analysed, which aims to prove our assumptions for RQ1 and 2 and directly answer RQ3. The whole system was evaluated from two parts, the therapy effect and the system effect. According to the data from daily questionnaires, the system was able to maintain breathing exercises on a baseline and at the same time, bring enough entertainment experience to participants. This confirmed our conjecture for RQ1&2 and became one of the strong evidences of answering RQ3. The system then was evaluated by features score, SUS and IMI. The features have gained an average score of ($4.11/5 \pm 0.81$) and the system has been scored as ($81.84/5 \pm 9.56$), both scores are positive and stable, which gives us great confidence in the development direction of the

system especially the direction of exploring novel gameplay models. Comparing to the previous BreatheHero (TO, 2020), new BreatheHero has expanded the game content and built a more complete game loop. This leads to a SUS score of (63.2/100 $\pm$ 7.29), the system still has a good usability but compared to the score of 81/100 for previous BreatheHero, the socre has been seriously decreased. This may be due to the huge increase in the complexity of the system. In IMI evaluation, a high level enjoyment score has been given which should confirm the results obtained in the daily research. However, low pressure score (52.61 $\pm$ 19.04) has been detected, this could because of the effect of tester's age and gaming experience. On the other hand, this still clarifies the direction of development and optimisation. Feedback and suggestions were also collected and discussed in the previous section and gave us a clue for further development. Limitations were raised because of the effect of covid-19 and lack of participants. The biggest impact on research is that no large-scale field study with young participants were organised.

Overall, at the current stage, all the research questions can be answered by the positive results of the project. Valuable test data and feedback have been given for the prototype. Large-scale field study is required for going further of the study.

Chapter 9

Conclusions

BreatheHero is a software development based project for resolving the problem that low adherence for breathing exercise was detected for young patients with cystic fibrosis. By learning the related work for GCP, we realized that the low adherence is because of the breathing exercise is considered as tired and boring by the young patients. We applied gamification technologies and managed to combine the breathing exercise with the game design.

By applying agile process, a complete game prototype was developed in a very short term and the novel GCP game BreatheHero is now coming into being. A field study was then processed with 5 participants and although there are many limitations, positive results were received. Moreover, worthy testing data and feedback have been collected which could play a significant role for further development.

The significance of BreatheHero lies in its attempt to gamification, data collection and development process guidance. The current BreatheHero is still a basic prototype and more content is required. We suggest to keep the development mechanisms and after some more Sprints development, a large-scale field study with young cystic fibrosis patients should be organised for further research on the therapeutic effects and entertainment effects of games. Subsequently, the research questions can be fully answered with universality.

Bibliography

Abrahamsson, P., Salo, O., Ronkainen, J. and Warsta, J., 2017. Agile software development methods: Review and analysis. *arxiv preprint arxiv:1709.08439*.

Balaji, S. and Murugaiyan, M.S., 2012. Waterfall vs. v-model vs. agile: A comparative study on sdlc. *International journal of information technology and business management*, 2(1), pp.26–30.

Balli, F., 2018. Developing digital games to address airway clearance therapy in children with cystic fibrosis: Participatory design process. *Jmir serious games* [Online], 6(4), p.e18. Available from: https://doi.org/10.2196/games.8964.

Bingham, P.M., Bates, J.H., Thompson-Figueroa, J. and Lahiri, T., 2010. A breath biofeedback computer game for children with cystic fibrosis. *Clinical pediatrics*, 49(4), pp.337–342.

Bingham, P.M., Lahiri, T. and Ashikaga, T., 2012. Pilot trial of spirometer games for airway clearance practice in cystic fibrosis. *Respiratory care*, 57(8), pp.1278–1284.

Bleszinski, C. and EpicGames, 2000. The art and science of level design. *Game developers conference*.

Blohm, I. and Leimeister, J.M., 2013. Gamification. *Business & information systems engineering*, 5(4), pp.275–278.

Boller, S., 2013. Learning game design: Game mechanics. *Knowledge guru*.

BritishLungFoundation, 2021. Children's lungs [Online]. Accessed on 2021-4-15. Available from: https://www.blf.org.uk/support-for-you/children.

Brooke, J., 1995. Sus: A quick and dirty usability scale. *Usability eval. ind.*, 189.

Brooke, J. et al., 1996. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194), pp.4–7.

Casas-Roma, J., Nelson, M.J., Arnedo-Moreno, J., Gaudl, S.E. and Saunders, R., 2019. Towards simulated morality systems: Role-playing games as artificial societies. *Icaart 2019 - proceedings of the 11th international conference on agents and artificial intelligence*, 1, pp.244–251.

Cohen, D., Lindvall, M. and Costa, P., 2004. An introduction to agile methods. *Adv. comput.*, 62(03), pp.1–66.

Compton, K. and Mateas, M., 2006. Procedural level design for platform games. *Aiide*. pp.109–111.

CysticFibrosisFoundation, 2021. Chest physical therapy [Online]. Accessed on 2021-4-5. Available from: https://www.cff.org/Life-With-CF/Treatments-and-Therapies/Airway-Clearance/Chest-Physical-Therapy.

CysticFibrosisTrust, 2021. Cystic fibrosisi faqs [Online]. Accessed on 2021-5-13. Available from: https://www.cysticfibrosis.org.uk/what-is-cystic-fibrosis/faqs.

Darbee, J.C., Ohtake, P.J., Grant, B.J. and Cerny, F.J., 2004. Physiologic evidence for the efficacy of positive expiratory pressure as an airway clearance technique in patients with cystic fibrosis. *Physical therapy*, 84(6), pp.524–537.

Davis, A., Bersoff, E. and Comer, E., 1988. A strategy for comparing alternative software development life cycle models. *Ieee transactions on software engineering* [Online], 14(10), pp.1453–1461. Available from: https://doi.org/10.1109/32.6190.

Eakin, M.N., Bilderback, A., Boyle, M.P., Mogayzel, P.J. and Riekert, K.A., 2011. Longitudinal association between medication adherence and lung health in people with cystic fibrosis. *Journal of cystic fibrosis*, 10(4), pp.258–264.

Elkins, M., Jones, A. and Schans, C.P. van der, 2006. Positive expiratory pressure physiotherapy for airway clearance in people with cystic fibrosis. *Cochrane database of systematic reviews*, (2).

Flume, P.A., Mogayzel Jr, P.J., Robinson, K.A., Rosenblatt, R.L., Quittell, L. and Marshall, B.C., 2010. Cystic fibrosis pulmonary guidelines: pulmonary complications: hemoptysis and pneumothorax. *American journal of respiratory and critical care medicine*, 182(3), pp.298–306.

Fowler, M., Highsmith, J. et al., 2001. The agile manifesto. *Software development*, 9(8), pp.28–35.

Franken, S., Kolvenbach, S., Prinz, W., Alvertis, I. and Koussouris, S., 2015. Cloudteams: Bridging the gap between developers and customers during software development processes. *Procedia computer science*, 68, pp.188–195.

Ge, J., 2018. Designing and building a robust, comprehensive achievement system [Online]. Accessed on 2021-8-15. Available from: https://www.gamasutra.com/blogs/JoshGe/20180703/321132/Designing_and_Building_a_Robust_Comprehensive_Achievement_System.php.

godotofficial, 2021. Godot game engine official website [Online]. Available from: https://godotengine.org/.

Hamari, J., Koivisto, J. and Sarsa, H., 2014. Does gamification work? – a literature review of empirical studies on gamification [Online]. *2014 47th hawaii international conference on system sciences*. pp.3025–3034. Available from: https://doi.org/10.1109/HICSS.2014.377.

Hofmeyr, J., Webber, B. and Hodson, M., 1986. Evaluation of positive expiratory pressure as an adjunct to chest physiotherapy in the treatment of cystic fibrosis. *Thorax*, 41(12), pp.951–954.

IEEE, 1990. Ieee standard glossary of software engineering terminology (std. 610.12-1990).

Jakobsson, M., Sotamaa, O., Moore, C., Begy, J., Consalvo, M., Gazzard, A., Medler, B., Williams, P., Nesbitt, K.V., Eidels, A. et al., 2011. game reward systems. *Game studies*, 11(1).

Knowles, M.R. and Durie, P.R., 2002. What is cystic fibrosis? Accessed on 2021-4-20.

Lee and Tyler, 2015. An annotated history of video game boss battles [Online]. Accessed on 2021-7-1. Available from: https://www.polygon.com/features/2015/9/28/9333685/annotated-history-boss-battles.

McIlwaine, M., Button, B. and Nevitt, S.J., 2019. Positive expiratory pressure physiotherapy for airway clearance in people with cystic fibrosis. *Cochrane database of systematic reviews*, (11).

Minotti, M., 2017. Cuphead review — a uniquely beautiful and worthwhile challenge [Online]. Accessed on 2021-7-1. Available from: https://venturebeat.com/2017/10/07/cuphead-review-a-uniquely-beautiful-and-worthwhile-challenge/.

Myers, L.B., 2009. An exploratory study investigating factors associated with adherence to chest physiotherapy and exercise in adults with cystic fibrosis. *Journal of cystic fibrosis*, 8(6), pp.425–427.

Myers, L.B. and Horn, S.A., 2006. Adherence to chest physiotherapy in adults with cystic fibrosis. *Journal of health psychology*, 11(6), pp.915–926.

Needs, B., 2021. Intrinsic motivation inventory (imi) [Online]. Available from: https://selfdeterminationtheory.org/intrinsic-motivation-inventory/.

Oikonomou, A., Hartescu, D., Day, D. and Ma, M., 2014. *Computer games physiotherapy for children with cystic fibrosis* [Online], Berlin, Heidelberg: Springer Berlin Heidelberg, pp.411–443. Available from: https://doi.org/10.1007/978-3-642-54816-1_21.

Petersen, K., Wohlin, C. and Baca, D., 2009. The waterfall model in large-scale development. *International conference on product-focused software process improvement*. Springer, pp.386–400.

Petkov, P., Köbler, F., Foth, M., Medland, R. and Krcmar, H., 2011. Engaging energy saving through motivation-specific social comparison. *Chi'11 extended abstracts on human factors in computing systems*. pp.1945–1950.

Physiopedia, 2021. Positive expiratory pressure (pep) devices [Online]. Accessed on 2021-5-20. Available from: https://physio-pedia.com/Positive_Expiratory_Pressure_(PEP)_Devices.

Prasad, S. and Cerny, F., 2002. Factors that influence adherence to exercise and their effectiveness: application to cystic fibrosis. *Pediatric pulmonology*, 34(1), pp.66–72.

Reedsyblog, 2021. What can rick and morty teach authors about structure? [Online]. Accessed on 2021-8-13. Available from: https://blog.reedsy.com/guide/story-structure/dan-harmon-story-circle/.

Research, A.M., 27-9-2016. Cystic fibrosis: protecting children from life-changing infections [Online]. Accessed on 2021-5-13. Available from: https://action.org.uk/research/cystic-fibrosis-protecting-children-life-changing-infections.

Rollings, Andrew and Adams, E., 2006. *Fundamentals of game design*, Englewood: Prentice Hall, pp.194–204.

Ryan, R.M., 1982. Control and information in the intrapersonal sphere: An extension of cognitive evaluation theory. *Journal of personality and social psychology*, 43(3), p.450.

Schans, C.P. van der, Prasad, A. and Main, E., 2000. Chest physiotherapy compared to no chest physiotherapy for cystic fibrosis. *Cochrane database of systematic reviews*, (2).

Schwaber, K., 1997. Scrum development process. *Business object design and implementation*. Springer, pp.117–134.

Studiobinder, 2020. How the dan harmon story circle can make your story better [Online]. Accessed on 2021-8-15. Available from: https://blog.reedsy.com/guide/story-structure/dan-harmon-story-circle/.

Taddeo, D., Egedy, M. and Frappier, J.Y., 2008. Adherence to treatment in adolescents. *Paediatrics and child health* [Online], 13(1), pp.19–24. Available from: https://doi.org/10.1093/pch/13.1.19. https://academic.oup.com/pch/article-pdf/13/1/19/11124569/pch13019.pdf, Available from: https://doi.org/10.1093/pch/13.1.19.

Thompson and Clive, 2004. Tough love: Can a video game be too hard? [Online]. Accessed on 2021-7-1. Available from: https://web.archive.org/web/20090210210016/http://www.slate.com/id/2100116/.

Thompson, A., 2021. The importance of video game narrative [Online]. Accessed on 2021-4-17. Available from: https://sbanimation.com/the-importance-of-video-game-narrative/.

TO, W.H.E., 2020. Pep integrated videogame to improve young patients' adherence to chest physiotherapy.

Volsko, T.A., 2013. Airway clearance therapy: finding the evidence. *Respiratory care*, 58(10), pp.1669–1678.

WestSuffolkNHSFoundationTrust, 2019. Bubble pep [Online]. Accessed on 2021-4-15. Available from: https://www.wsh.nhs.uk/CMS-Documents/Patient-leaflets/Physiotherapy/6242-1-Bubble-PEP-Bubble-Positive-Expiratory-Pressure.pdf.

Williams, B., Mukhopadhyay, S., Dowell, J. and Coyle, J., 2007. Problems and solutions: accounts by parents and children of adhering to chest physiotherapy for cystic fibrosis. *Disability and rehabilitation*, 29(14), pp.1097–1105.

Worldwide and CysticFibrosis, 2018. What is cystic fibrosis. *Cystic fibrosis worldwide url: https://www. cfww. org/what-is-cystic-fibrosis [accessed 2018-09-10][webcite cache id 72k3312uy]*.

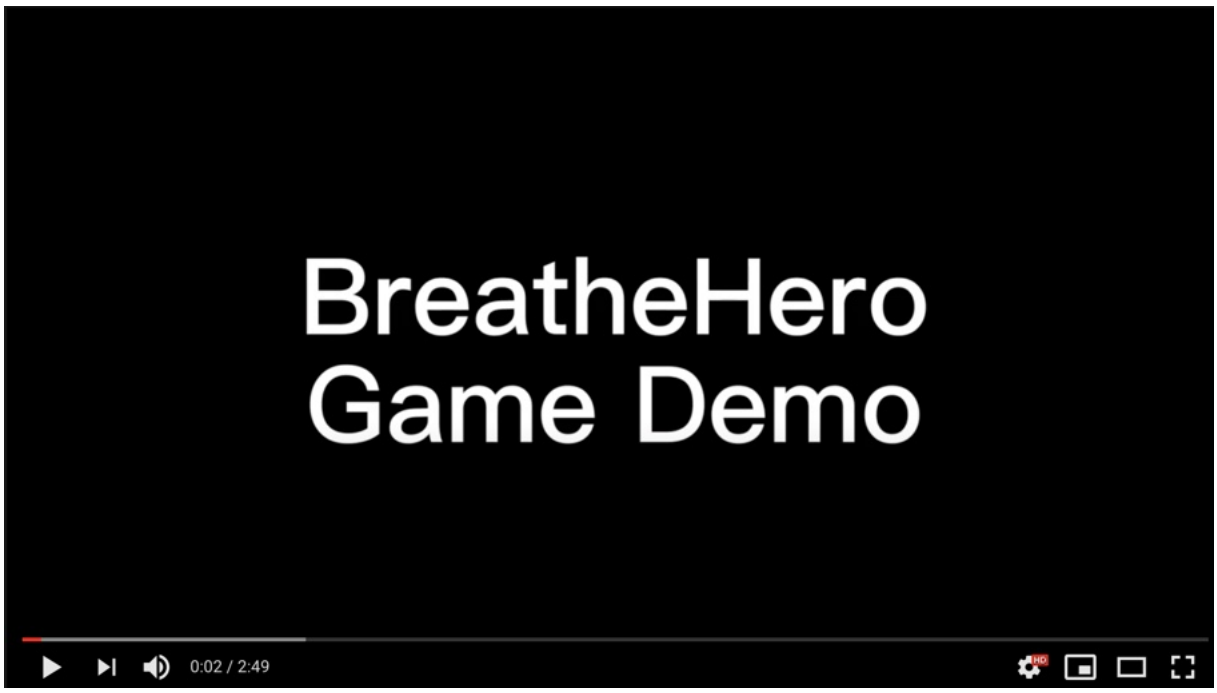
ZHANG, X. and YIBIN, H., 2021. Design guidelines for onboarding experience in narrative-rich games.

Zichermann, G. and Cunningham, C., 2011. *Gamification by design: Implementing game mechanics in web and mobile apps*. " O'Reilly Media, Inc.".

Appendix A

Game Demo

A video for the game demo has been uploaded to Youtube and can be accessed by the link: https://youtu.be/ytvxuhgfQkM.



Appendix B

Story Script

Chapter1: Finding the Flame

Scene One

In A Warm Room

The Mother Dragon is holding a book beside the fireplace and telling a story to the little dragons. The little dragons are sitting in a circle next to their mother, waiting eagerly.

Dialogue Start

Green Little Dragon: (Urgently) Mom! What is the story today! I cannot wait!

Other Little Dragons: (Go along with yelling) Yeah! Yeah! ...

Mother Dragon:(Gently) Oh! Hold on kids. Today is a story on another little dragon, just like you! But he/she was not able to breath fire when he/she was born.

Yellow Little Dragon: (Doubtfully) I can breath fire when I was born! (Try to breathe fire) Why he/she cannot?

Other Little Dragons: (Gossiping) Yeah! We can breath fire! Why? How? ...

Mother Dragon: (Slightly reproachfully) Oh my little guys! Don't be like that rude! There are always little guys be chosen to be different from the others. But the most important thing is that, the little guy was a brave one. He/She was never giving up to look for his/her flame. The little guy overcame difficulties and obstacles, learnt lessons and made friends! And one day, he/she would become the BreathHero! Would protect his/her friends with his/her own flame!

Other Little Dragons: (Apologetically) Sorry mom! We want to be the BreathHero just like he/she! Can we have our story now?

Mother Dragon: (With a smile) Oh, of course! Let the little guy's adventure begin!

Scene Two

In the Forest

Little Red Dragon gets up slowly and dazedly in a forest, cough up smoke. Little Gray Dragon surrounded anxiously.

Dialogue Start

Little Gray Dragon: (With concern) Are you alright?! It is the third time this week you broke into the forest looking for your flame and hit by the monsters. It could be just a tale! A story!

Little Red Dragon: (Gratefully and firmly) Thank you little Gray! But I am still believing in

that! I can feel something warm at the edge of the forest! I know I am special but it is not because I cannot breath fire. I am special because when I find my flame, I would breath out the largest fire ever! And for that, I would be willing to try thousand of times and practise again and again! Would you please trust me one more time?
Little Gray Dragon: (Sigh gently but firmly) I will little red. But promise me take good care of yourself okay? Let's back to the adventure.
–Level 1 Start–

–Level 4 End–
Scene Three
At the Front of the Castle
Dialogue Start
Little Gray Dragon: Look!! A castle! I have never known there is a castle in the deep forest! Maybe you are right! There are so many miracles. And maybe, maybe there would be one for you as well. The flame!
Little Red Dragon: Thank you little gray! It was so lucky that I have you nearby! Yeah, I can feel the flame! It is so close! It must be in the castle!
Little Gray Dragon: A huge slime!! Watch out! It mush be the challenge of us! Go for it little red! I am with you!
–Level 5 Start–

–Level 5 End–
Scene Four
Get into the Castle and Find the Dim Flame
Dialogue Start
Little Red Dragon: Is this the flame?
Little Gray Dragon: I think so, but it is not shining any more!
Little Red Dragon: There must be something happening in the forest!
Little Gray Dragon: You are right, or the flame would protect the forest and I have never saw slimes like that. Little red, don't give up here! Let's go further and explore more and make you flame shining again!
Little Red Dragon: Thank you to be with me little gray! Yeah, as long as I believe the flame in my heart, I would not stop moving on! Please be with me for more adventures!
Little Gray Dragon: I will little red and my pleasure.

Appendix C

Development Plans

C.1 Quality Plan

Introduction
The document defines the quality assurance for the project. In all the cases, the quality for the product should be guaranteed.

Documents
The following aspects are covered in the quality assurance including the requirements, development, testing and maintaining of the project.

- Requirement
- Backlogs
- Version Control Plan
- Coding Style
- Test Plan
- Review and Maintains
- Responsibilities

Quality Assurance
During the development process, the following elements are required to be guaranteed and all developers should take responsibility for them.

- **Paperwork:** In all the stages, paperwork would need to be always complete and readable. Development instruction and process should be recorded, which is for later development. All the files would need to be well structured.
- **Codes:** All the code work would need to consider its runnability, robustness and maintainability. Comments would need to be added clearly near the code for the codes readability. More instructions would be given in the code style document.
- **Review:** All the works including both the paperwork and code would need to be reviewed periodically. The trace-ability for the review document would need to be guaranteed.

Software's Identification and Trace-ability
The requirements listed will be used for check if the software meets the deliverable. The review process will ensure if the final product meets all the requirements, design and tests.

Checkpoints and Review
At least one checkpoint would need to be set for each stage of the project, which is called a stage exit. This includes that when a stage is successful completed, all the deliverable is ready, all outstanding issues have the complementary plans and a plan for next stage is created.

At least one review process would happen at the end of each stage. All the deliverable including both document and code would be reviewed. This is for guaranteeing the deliverable meets all the requirements. The further details can be found in Review Plan.

Issue Reporting and Correctness
All the issues would need to be reported during the development. The "issue report" feature in GitHub would be used for recording and tracing all the document on problem resolving.

C.2 Coding Style

Introduction
Coding standards would be defined in the document including identifying, formatting and documentation etc.

Identifiers
All the parameters and definitions including class, method name, variables and constant would need to be meaningful and use letters and numbers only, special symbol "_" is allowed when it is necessary.

Formatting
The coding language used for the project is godot script. The indentation, spacing and layout would strictly follow the coding instructions provided by Godot official (godotofficial, 2021).

Documentation
The documentation including class comment, method comment and code comment. All the comments need to give brief introduction including the code's usage, instruction and logic.

C.3 Test Plan

introduction
The document gives the solution for the testing of the project. After the deliverable, all the features would need to process the regression testing. This is for spotting problems in an early stage and fixing them with a lower cost. During the testing, test cases would be created and result would be recorded.

Testing
For the system, a black-box testing would be suitable for our case. The black-box testing would need to check if each feature lead to the ideal output we expected. All the test cases would be listed in advance according to the backlogs. For a black box testing, two principles would be followed: 1) No need to consider the interactions between bugs, all the cases are

independent. 2) There is no need to test unfinished part of the system, only runnable parts are required.

The test adopts the form of simple comparison between the output and the expected results. For this purpose, the following table is used for logging the test result.

Table C.1: Test Log

ID	...
Test Feature	...
Test Objective	...
Description	...
Result(Pass/Fail)	...
Date	...
Tester	...

Debugging
After and during the testing process, if the result is different from the expected output. A bug report should be recorded. The following table is for the bug reporting and trace. The issue should be closed after the bug is resolved.

Table C.2: Bug Reporting

ID	...
Bug Name	...
Reporter	...
Date	...
Description	...
Priority	...
Close Date	...
Developer	...

C.4 Review Plan

Introduction
The document would give a brief description of review process of the project. As mentioned, all the paperwork and code need to be review at least once at the end of each stage.

Document Review
The documents need to be reviewed with its producing order and includes the following and all the documents need to be well organised and stored in the local folder.

- Requirements
- Backlog
- Sprint Record

- User Stories
- User Cases
- Tests and Test Results
- Exception Handling

Code Review
The code would need to be review at the end of each stage as well. This includes the following elements of the system.

- Runnability
- Output Correctness
- Maintainability
- Test Code
- Debug Code

C.5 Version Control

Introduction
Version control is a significant part for software development. Save points are created during the development and so that the contributions can be saved and synchronized at the point. The purpose for the document is for formally layout the version control process for the project. Github is used as the version control tool for the project.

Central Branch
A main branch is set as the central repository for the project and managed by Dr. Christof. All the members should avoid coding directly in the main branch in case any break of the code, which means the main branch is pull and merge allowed. All the pull requests to the main branch would need to be reviewed by Dr. Christof.

Development Branches
For all the developers, they would be able to create branches belongs to them or specify with some specific features. The development branches are used for distributed realization of different functions. After the feature is completed, a pull request can be created for merging the work to the main branch.

Commit
For all the commits, a brief description of the save point is required. As far as there is no crash for the codes, the commit action can happen at anytime during the development. For the safety of the code, developers should not work on the branches belongs to others.

Documentation
The documentation of the commit, pull request and merge action etc. should be well recorded for trace-ability of the versions. Such documents provide the version histories and could be useful in case roll-back is required.

Appendix D

Design Diagrams

D.1 Whiteboard and Brainstorming

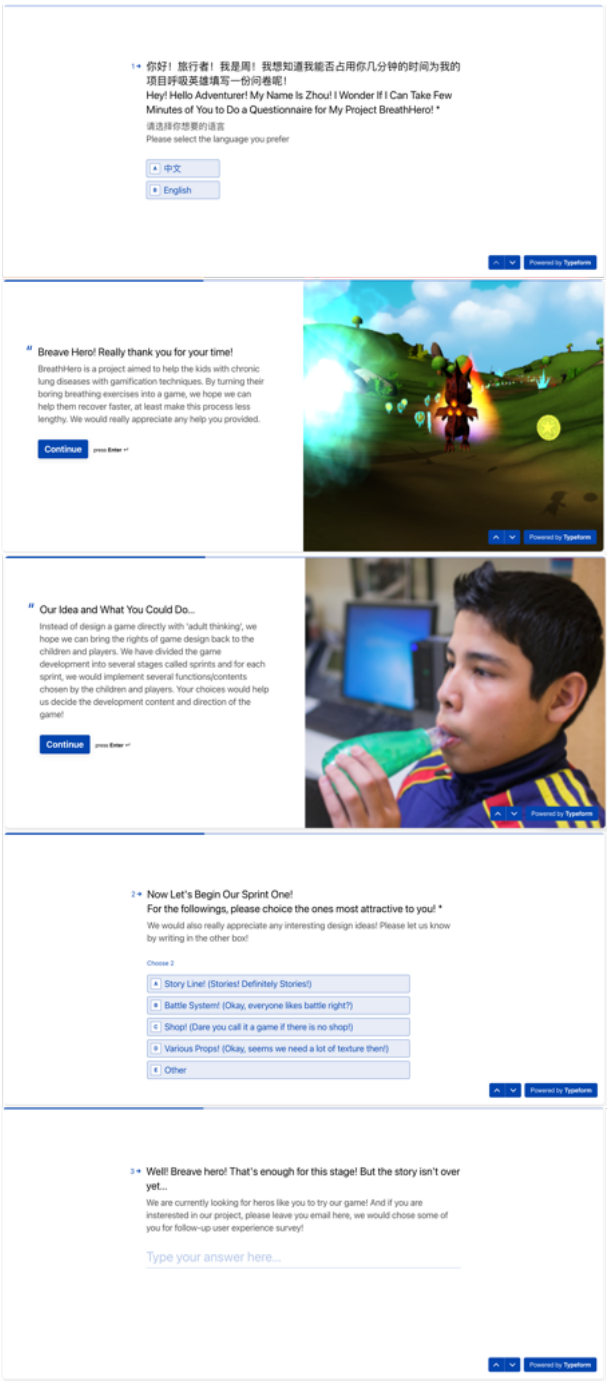
D.2 Development Road-map for BreatheHero

D.3 Participant Survey, Result and Receipt

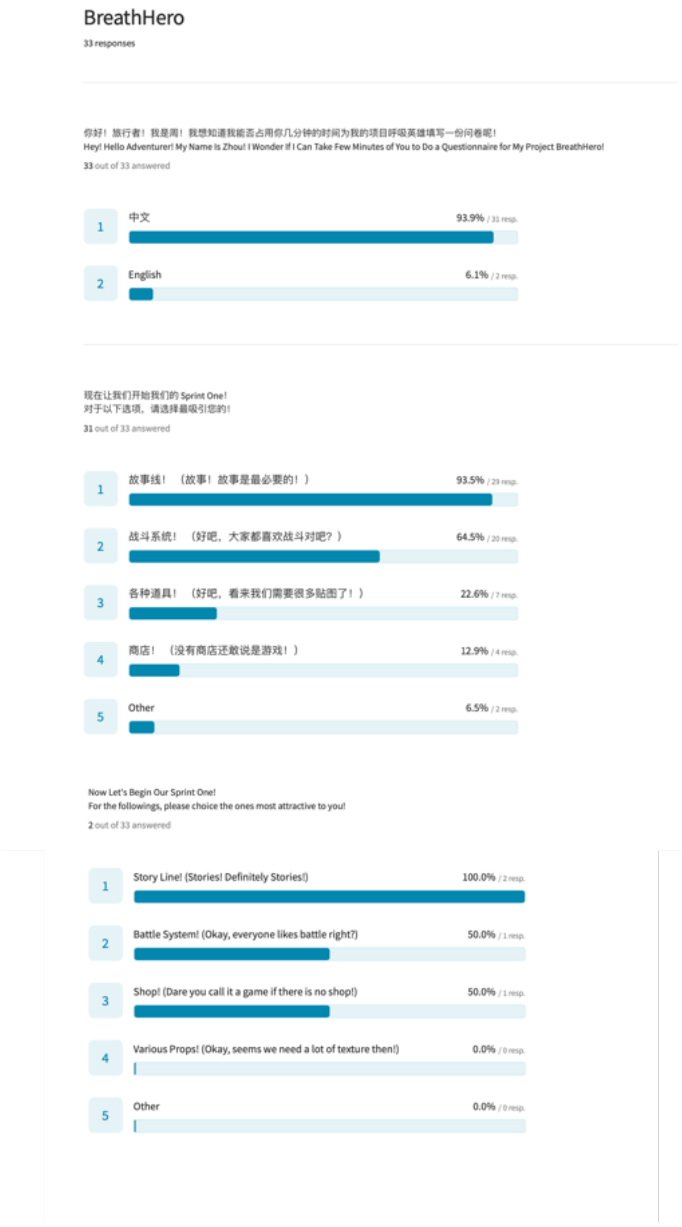
For a better understanding and convenient for the participants, the survey has been designed into bilingual, one-to-one correspondence between Chinese and English options and results have been recalculated according to the final result.

D.3.1 Sprint 3

Sprint 3 Questionnaire



Sprint 3 Vote Result



BreathHero

33 responses

你好！旅行者！我是周！我想知道我能否占用你几分钟的时间为我的项目呼吸英雄填写一份问卷呢！
Hey! Hello Adventurer! My Name is Zhou! I Wonder if I Can Take Few Minutes of You to Do a Questionnaire for My Project BreathHero!

33 out of 33 answered

1	中文	93.9% / 31 resp.
2	English	6.1% / 2 resp.

现在让我们开始我们的 Sprint One!
对于以下选项，请选择最吸引您的!

31 out of 33 answered

1	故事线！（故事！故事是最必要的！）	93.5% / 29 resp.
2	战斗系统！（好吧，大家都喜欢战斗对吧？）	64.5% / 20 resp.
3	各种道具！（好吧，看来我们需要很多贴图了！）	22.6% / 7 resp.
4	商店！（没有商店还敢说是游戏！）	12.9% / 4 resp.
5	Other	6.5% / 2 resp.

Now Let's Begin Our Sprint One!
For the followings, please choice the ones most attractive to you!

2 out of 33 answered

1	Story Line! (Stories! Definitely Stories!)	100.0% / 2 resp.
2	Battle System! (Okay, everyone likes battle right?)	50.0% / 1 resp.
3	Shop! (Dare you call it a game if there is no shop!)	50.0% / 1 resp.
4	Various Props! (Okay, seems we need a lot of texture then!)	0.0% / 0 resp.
5	Other	0.0% / 0 resp.

Sprint 3 Receipt

Braves Heroes,
I am very honoured to have you involved in the project. Please allow me to introduce our project in detail. There may be information you are interested in here, but please don't have to read all of it, just select the part you need.

Overview: This email will contain an **introduction** to our project, the **shortcomings we are facing**, the **current progress, plans, and the results of the questionnaire**. At this stage, **you don't need to do anything. We are very honoured to have you here. If you are interested in the trial of the game in the future, we will provide a Demo version.** We will only send update emails when Sprint is updated. If you wish to no longer receive project updates, please reply unsubscribe.

Project introduction: As mentioned before, the project aims to help children with congenital lung diseases to carry out breathing training more easily. In game design, we need to focus more on **combining the process of breathing training with gameplay**. Another focus is on the **motivational effect of the story on the children**. The project itself is relatively simple, but as far as I know, we also hope that the project can be really used for treatment. For this reason, we are also trying to expand the academic nature of the project, and you will be an indispensable part of this process. For me, you are also one of the valuable members of the development team!

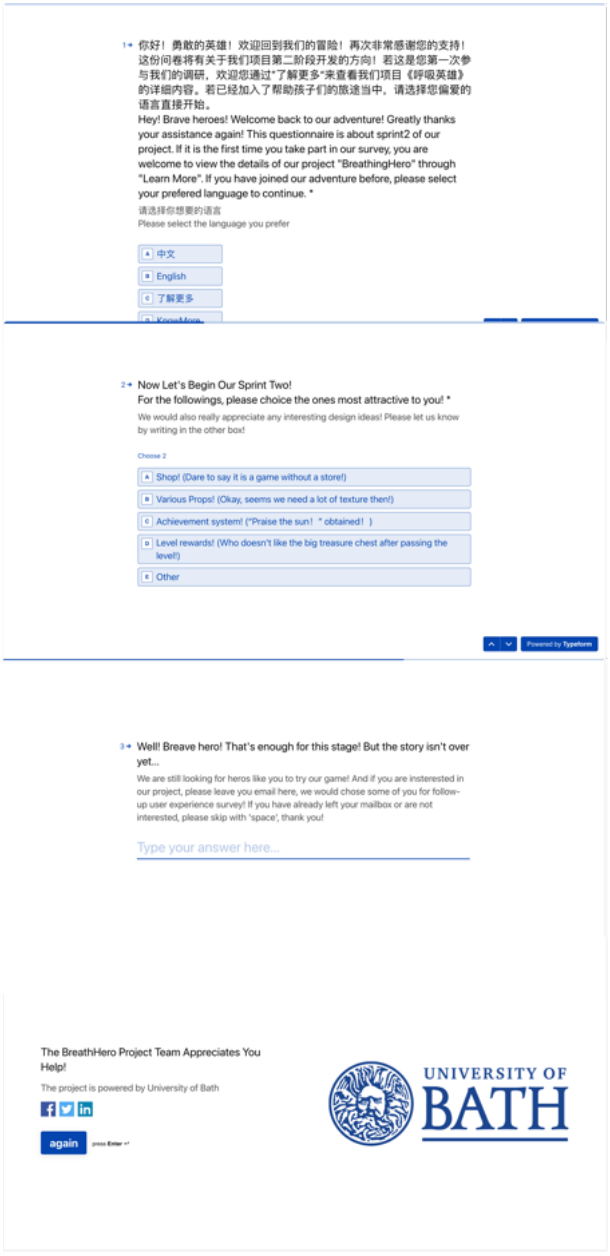
Shortcomings: The project was refactored this year, migrating from the originally messy Unity engine to the Godot engine, which caused the game to have to be developed from scratch. On the other hand, due to the emerging engine, we have to spend more time exploring and learning. At this point, we may face the following shortcomings: **First of all**, as the first batch of developers after the migration, we are very happy to leave more development documents, comments and design concepts for subsequent developers, but it is difficult for us to guarantee the integrity and super high quality of the game since we lack materials as well as the development instructions. **Furthermore**, the age of our actual users will fall between 5-15 years old. For this reason, our game content may be simpler than imagined, or even just a development framework. **Finally**, our number of developers is limited, and the actual results may be far from expected, but we will try our best to optimize the content. **We hope to get understanding in the follow-up research and experience.**

Current progress, plan, and questionnaire results: As mentioned before, we will advance the development process through sprints. In this survey, we have established two development direction: **story lines** (93.5%) and **battle system** (64.5%). Other interesting ideas mentioned by you will also be included in the next round of sprint voting. Thank you again for your participation. Sprint One will last about 3 days, during this period we will develop the corresponding content, and then usher in the next Sprint. **Really thank you for your support again!**

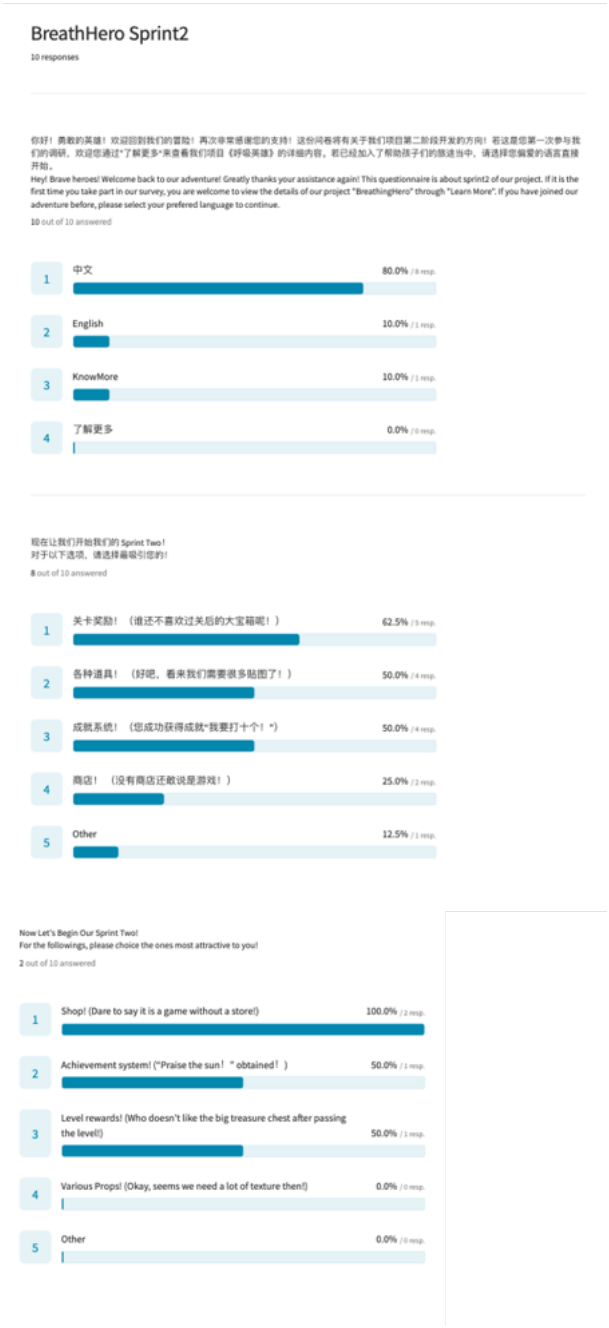
Sincerely,
BreathHero Team

D.3.2 Sprint 4

Sprint 4 Questionnaire



Sprint 4 Vote Result



BreathHero Sprint2

10 responses

你好！勇敢的英雄！欢迎回到我们的冒险！再次非常感谢您的支持！这份问卷将有关于我们项目第二阶段开发的方向！若这是您第一次参与我们的调研，欢迎您通过"了解更多"来查看我们项目《呼吸英雄》的详细内容，若已经加入了帮助孩子们的旅途当中，请选择您偏爱的语言直接开始。
Hey! Brave heroes! Welcome back to our adventure! Greatly thanks your assistance again! This questionnaire is about sprint2 of our project. If it is the first time you take part in our survey, you are welcome to view the details of our project "BreathingHero" through "Learn More". If you have joined our adventure before, please select your prefered language to continue.

10 out of 10 answered

1	中文	80.0% / 8 resp.
2	English	10.0% / 1 resp.
3	KnowMore	10.0% / 1 resp.
4	了解更多	0.0% / 0 resp.

现在让我们开始我们的 Sprint Two！
对于以下选项，请选择最吸引您的！

8 out of 10 answered

1	关卡奖励！（谁还不喜欢过关后的大宝箱呢！）	62.5% / 5 resp.
2	各种道具！（好吧，看来我们需要很多贴图了！）	50.0% / 4 resp.
3	成就系统！（您成功获得成就"我要打十个！"）	50.0% / 4 resp.
4	商店！（没有商店还敢说是游戏！）	25.0% / 2 resp.
5	Other	12.5% / 1 resp.

Now Let's Begin Our Sprint Two!
For the followings, please choice the ones most attractive to you!

2 out of 10 answered

1	Shop! (Dare to say it is a game without a store!)	100.0% / 2 resp.
2	Achievement system! ("Praise the sun！" obtained！)	50.0% / 1 resp.
3	Level rewards! (Who doesn't like the big treasure chest after passing the level!)	50.0% / 1 resp.
4	Various Props! (Okay, seems we need a lot of texture then!)	0.0% / 0 resp.
5	Other	0.0% / 0 resp.

Sprint 4 Receipt

来自呼吸英雄团队的问候!/Greetings from BreathHero Team!

Mon 02/08/2021 15:09

To: Xiaoyan Zhou

2 attachments (5 MB) Download all Save all to OneDrive - University of Bath

勇敢的英雄，

非常高兴的告诉您，在上一个Sprint中我们完成了**简单的故事系统和boss战的框架，您可以在附件中查看其大致设计（抱歉现阶段没能提供更精美的资源处理）**。在此我们的游戏的chapter1已经成为了一个完整的闭环，我们将对游戏个细节和扩展功能进行修饰！**现在我们将开启我们的新的旅程Sprint2**，如果可以，我们希望占用您一点点的时间填写一份类似的问卷！再次感谢您的关注和支持！问卷链接：https://kd6r8zoeppb.typeform.com/to/kVoBMFz8

呼吸英雄团队，

谨上

Brave Hero,

I am very happy to tell you that in the last Sprint we completed **a simple story system and a framework for boss battles**. **You can view its rough design in the attachment (sorry for not being able to provide more exquisite resource processing at this stage).** Here, chapter1 of our game has become a complete closed loop, and we will modify the details and extended functions of the game! **Now we will start our new journey Sprint2**, if possible, we hope to take a little bit of your time to fill out a similar questionnaire! Thank you again for your attention and support!
Questionnaire link: https://kd6r8zoeppb.typeform.com/to/kVoBMFz8

Sincerely,
BreathHero Team

Appendix E

Participants Information Sheets

Project Name: Boosting Chest Physiotherapy's Positive Effect with Gamification Technologies

Experimenter: Xiaoyan Zhou
Experimenter's Email: xz2539@bath.ac.uk
Project Supervisor: Dr. Christof Lutteroth
Supervisor's Email: c.lutteroth@bath.ac.uk

A Letter for Participants:

Dear Participants,

It is my great honor to invite you to join in the research BreatheHero. It is a proud project for creating games for children with lung disease who would need to do breathing therapy daily. With the gamification technology, we aims to build game prototypes that can assist children with their breathing exercise and make it less painful. In the research, you would have the opportunity to play the PEP integrated videogame prototype and give us valuable feedback. The research is a meaningful adventure and it is important that you make an informed decision on whether you want to join in. Please take some to read the particifants information sheet carefully and discuss with anyone you want if you wish. You are welcomed to ask any questions on the project before and in the research and of course have the rights to withdraw from the study at anytime and without any reasons. The data collected would be protected and anonymity and you can withdraw the data within two weeks after the research by emailing the experimenter. All information will be subject to the current conditions of the Data Protection Act. Whether you join in the research or not, we appreciate you attention and help, you are the hero of the children.

Sincerely,

BreatheHero Team

Background:

Breathing exercises is an airway clearance technique for treating chronic lung diseases by assisting the patients to expel thick sputum. The patients with chronic lung diseases would need to do such exercises daily to prevent the worsening of lung situation. However, low adherence situation has always been a difficult problem to solve. We believe that the situation

happens because the young patients could easily feel the exercise is boring and exhausted. In this case, we thought gamification technology could be used and developed a game prototype for the patient. The research aims to make the breathing exercise more easier and entertaining and help the young patients to maintain their interest and adherence to it.

What you could do to help us?

In the study, you would be invited to play our PEP videogame prototype BreatheHero for a course of treatment (which is last for 7 days and a session per day, each session would last for 10 minutes). This involves using a special PEP controller for breathing and out. The breath signal would then be converted into character's movement in the game. The game has the following features: storyline and morality system, reward system, shop, achievements system and boss battle. We hope you can enjoy the game play circle at the same time give us feedback. We would collect the data for your breath activities during the playing such as breath pressure and breath frequency etc. And at the end of the research, you would be invited to fill a questionnaire on you overall experience, opinions on the system and suggestions for further developments.

Are there any risks?

The game is friendly to the participants. Please always take a break if you are breathing too hard when playing the game. And feel free to decide whether to resume the gameplay or not. You can always switch back to the normal breathing exercise at any time.

Do you have to take part in?

The answer is no. It is your right to withdraw the study at anytime without any reasons. And if you withdraw the study, all the data collected would be destroyed and your name would be removed from the study list. A consent form would be filled by you after reading the participants information sheet. You may stop you participation anytime if you feel discomfort. If you feel any discomfort, please inform the supervisor and experimenter immediately.

The right for you to complain.

We would be really sorry for bringing you any discomfort and inconvenience during the study. Please contact project supervisor Dr. Christof Lutteroth for making any complaint for the research.

How your data would be used and protected?

All the data collected will be in conjunction with University of Bath GDPR data policies. The data would be confidential throughout the whole study and post-study assessment. The data would be anonymously stored in digital storage and only participant ID would be used to recognize them, no personal information would be included. The anonymous data might be published in academic journals or conferences. The data would additionally be stored for 12 months after the study ends and would be destroyed immediately if you required.

Further Information:

We really appreciate your participation and you are welcomed for any queries before, during and after the study. Thank you for be the children's hero.

Appendix F

Consent Form

Project Name: Boosting Chest Physiotherapy's Positive Effect with Gamification Technologies

Experimenter: Xiaoyan Zhou
Experimenter's Email: xz2539@bath.ac.uk
Project Supervisor: Dr. Christof Lutteroth
Supervisor's Email: c.lutteroth@bath.ac.uk

The part for experimenter:
I, the study experimenter,
Confirm that I have explained the nature, needs, purpose and potential risks carefully with my participants of the proposed research. I promise to protect the confidentiality of my participants and will only collect the data for research purposes for the study.

Signature: **Date:**

The part for participant:
I, the participant,
Have been informed and totally understand the following points:

- ☐ I have read and understand the participant information sheet.
- ☐ The experimenter has explained the nature, needs, purpose and potential risks for me.
- ☐ All my quires have been answered satisfactorily before the study.
- ☐ I agree with using my personal information for the research purpose.
- ☐ I understand my data would be collected and strictly protected by Data Protection Act.
- ☐ I agree to be the volunteer of the described study and give full consent to my participation.

Signature: **Date:**

Appendix G

Field Study Questionnaires

Project Name: Boosting Chest Physiotherapy's Positive Effect with Gamification Technologies

Experimenter: Xiaoyan Zhou
Experimenter's Email: xz2539@bath.ac.uk
Project Supervisor: Dr. Christof Lutteroth
Supervisor's Email: c.lutteroth@bath.ac.uk

Daily Questionnaire

The questionnaire requires to be filled daily in the session.

1. Does the session be fully completed (10 minutes)? If not, please write down the situation happened.

   ---

2. How many levels you have completed today?

   ---

3. How many times you have breathed (attacked with fire) in the session? And how long the average breath length is?

Total Breath No.	Average Breath Length (s)
...	...

4. How many times you have coughed (break the treasure box) in the session?

   ---

5. From score 0-5, please mark your entertainment level for today's game play.

Participant No.: **Date:**

Section I: Participant's Background Information & Game History

- **Participant's Background Information**

  1. What is your gender? Male/Female
  2. What is your age?

     ---

  3. Do you have any breathing or lungs disease history or medical history? If so what is it?

- **Previous Game History**

  1. How long (years) you have played video games? How often do you play video games?

     ---

  2. what kind of games do you prefers? (Multiple choice)

     - ☐ Action games
     - ☐ Action-adventure games
     - ☐ Adventure games
     - ☐ Role-playing games
     - ☐ Simulation games
     - ☐ Strategy games
     - ☐ Sports games
     - ☐ Puzzle games
     - ☐ Other

  3. What kind of devices are you usually used for playing? (e.g. PC, Game Console and Mobile etc.)

     ---

  4. Have you had any experience on any gamified physiotherapy games? If so, what was it and what do you think of its effectiveness?

     ---

  5. Please select one or more following play styles that attract you.

     - ☐ I enjoy interacting with other players in the game instead of playing alone.
     - ☐ I enjoy exploring the game world more than battling with other players.
     - ☐ I enjoy the feeling that when I gain achievements after finishing challenges.
     - ☐ I enjoy the feeling of win the game and like other players to admire me.
     - ☐ I enjoy the tension and excitement brought by the game.
     - ☐ I enjoy the game which brings a sense of ease.
     - ☐ I enjoy honing my game skills.

Section II: Game Features & Feedback

The game system has the following main features: reward system, achievement system, boss battle and story line and its morality system. The section would mainly marking and asking questions based on these features.

- **Score the game**: please rate 1 (least enjoy) - 5 (most enjoy) for the game features/objects.

Game Features/Objects	**Score**
Reward System	
Collecting coins	
Breathing fire	
Melt ice of the coin	
Having regular levels	
Story Line & Morality	
The story has presented	
The morality the story presented	
Achievement System	
The content of the achievements	
The way of the achievements achieved	
Boss Battle	
The design of the boss fight	
The way to fight the boss	
The image of the boss	
The difficulty of the boss fight	
The reward after the boss fight	
World View	
Environment	
The way maps breathing with breathing fire	
Level design (the combination of collection, dodging, attacking elements)	
Game Control	
Overall System	
Whole Game Play Circle	

- **Open questions**:

1. Have you joined in the previous development direction survey of the game? If so, do you like the way the project were developed? Were you inspired by the interaction with the development team?

---

2. Have you experienced the whole game play circle (All the designed features)? Which feature did you enjoyed the most? Please give brief explanation.

---

3. What do you think about the design of the story line, did you feel be inspired and want to know what would happen next?

---

4. Have you checked the credit cookies in the achievement system and if so do you like the content and were you inspired by them?

---

5. After playing the whole game circle, compare with single game feature, do you think the combination of all elements (the full game circle) brings you better game experience and longer term interest?

---

6. Does the web version of game bring you convenient and positive effect on playing the game?

---

7. Do you have any suggestions and feedback for the game and further development?

Section III: System Usability Scale (SUS)

This is the section for evaluating the system's usability.

- **Score the system**: please rate 1 (Strongly disagree) - 5 (Strongly agree) for the system's usability.

System Object	Score
The system is interesting and I would like to use it frequently.	
I think the system is cumbersome.	
I think the system is easy to use.	
I think I would need someone to assist with me to use the system.	
I think all the elements in the system are well integrated.	
I think the system has a lot of inconsistency with its guidance and description.	
I feel very confident when I use the system.	
I need to learn a lot of instructions before I start to use the system.	
The difficulty of the system increases sensibly with my gaming skill	
The online platform version brings me convenient on playing the game	

- **Fill the checkbox if you understand the object's meaning in the game**:
  - ☐ Coins
  - ☐ Coins with ice
  - ☐ Little slime
  - ☐ Slime King
  - ☐ Treasure box
  - ☐ Achievements
  - ☐ How to melt the ice
  - ☐ How to choose levels
  - ☐ How to unlock achievements
  - ☐ How to watch the story line

Section IV: Intrinsic Motivation Inventory (IMI)

This is the section for evaluating the system's motivation effects. Please rate 1 (Strongly disagree) - 5 (Strongly agree) for the system's intrinsic motivation.

- **The system's interest/enjoyment level**:

System Object	Score
I enjoyed playing the whole game.	
I feel fun when I play the single level.	
I think the game is boring.	
The game was not attractive to me at all.	
I feel all the single game element is fun.	

- **The system's pressure/tension level**:

System Object	Score
I feel very nervous while playing the game.	
I feel very tense while playing the game.	
The game inspired me a lot.	
I feel very relax when I playing the game.	
I feel very pressured when I playing the game.	
I was anxious when I playing the game.	

Appendix H

Raw Results Output

H.1 Analysed Result for Daily Questionnaire

Participant No.	Date	Session's Completeness	Levels Completed	Total Breath No.	Average Breath Length	Total Cough No.	Entertainment Score
Participant1	24-Aug	Y	2	7	3	24	4
	25-Aug	Y	2	10	5	28	4
	26-Aug	Y	2	12	6	39	5
	27-Aug	Y	3	10	4	26	3
	28-Aug	Y	2	13	3	42	4
	29-Aug	Y	2	10	6	35	3
	30-Aug	Y	2	10	5	25	3
Participant2	25-Aug	Y	3	12	6	36	4
	26-Aug	Y	3	17	5	54	4
	27-Aug	Y	4	18	6	60	4
	28-Aug	Y	4	20	6	67	4
	29-Aug	Y	3	14	5	50	3
	30-Aug	Y	4	24	6	70	3
	31-Aug	Y	3	15	3	47	2
Participant3	23-Aug	Y	2	8	4	21	4
	24-Aug	Y	3	17	5	30	4
	25-Aug	Y	3	10	4	34	3
	26-Aug	Y	2	15	6	37	3
	27-Aug	Y	3	14	4	36	3
	28-Aug	Y	2	17	6	47	4
	29-Aug	Y	2	13	4	36	3
Participant4	23-Aug	Y	3	13	5	39	3
	24-Aug	Y	2	11	3	37	5
	25-Aug	Y	4	17	5	32	3
	26-Aug	Y	3	12	5	38	3
	27-Aug	Y	3	13	6	33	3
	28-Aug	Y	3	15	3	48	2
	29-Aug	Y	3	15	5	45	3
Participant5	23-Aug	Y	4	20	5	56	4
	24-Aug	Y	3	23	5	53	5
	25-Aug	Y	3	14	6	46	4
	26-Aug	Y	4	27	5	67	4
	27-Aug	Y	3	17	5	54	4
	28-Aug	Y	4	25	5	76	4
	29-Aug	Y	3	16	6	47	4
Participant1_Average			2.142857143	10.28571429	4.571428571	31.28571429	3.714285714
Participant2_Average			3.428571429	17.14285714	5.285714286	54.85714286	3.428571429
Participant3_Average			2.428571429	13.42857143	4.714285714	34.42857143	3.428571429
Participant4_Average			3	13.71428571	4.571428571	38.85714286	3.142857143
Participant5_Average			3.428571429	20.28571429	5.285714286	57	4.142857143
Total Average			2.885714286	14.97142857	4.885714286	43.28571429	3.571428571
Standard Diviation			0.718308003	4.755713061	1.022437358	13.71682399	0.739065956

H.2 Analysed Result for System Questionnaire

Game Features							
Each Feature Average Score	4.10833333	**Standard Diviation**	0.80748425	**System Average Score**	81.838	**Standard Diviation**	9.56482723
SUS							
SUS	63.2	**Standard Diviation**	7.29383301				
Enjoyment Level							
Enjoyment Average Score	69.6	**Standard Diviation**	9.20869155				
Pressure Level							
Pressure Average Score	52.614	**Standard Diviation**	19.0422433				

Appendix I

Code

I.1 File: Login.gd

```
# @author Dr. Christof
extends Control

export(bool) var TEST_GAME = false
export(String) var CREDENTIALS_PATH = "user://credentials.txt"


func _ready():
  if TEST_GAME:
    get_tree().change_scene("res://Game/Game.tscn")

  Firebase.Auth.connect("signup_succeeded", self,
    "_on_FirebaseAuth_signup_succeeded")
  Firebase.Auth.connect("login_succeeded", self,
    "_on_FirebaseAuth_login_succeeded")
  Firebase.Auth.connect("login_failed", self,
    "_on_FirebaseAuth_login_failed")

  # if saved credentials available, show them
  var file = File.new()
  if not file.file_exists(CREDENTIALS_PATH):
    # No saved credentials
    $VBoxContainer/GridContainer/Email.grab_focus()
    return
  file.open(CREDENTIALS_PATH, File.READ)
  if file.eof_reached():
    print("Error: credentials file empty")
    return
  var credentials : Dictionary = parse_json(file.get_line())
  if not credentials.has("email") or not
    credentials.has("password"):
    print("Error: credentials file does not contain email or
    password")
    return
  $VBoxContainer/GridContainer/Email.text = credentials["email"]
  $VBoxContainer/GridContainer/Password.text =
    credentials["password"]
  file.close()
  $VBoxContainer/HBoxContainer/LoginButton.grab_focus()

func _on_LoginButton_pressed():
  Firebase.Auth.login_with_email_and_password(
    $VBoxContainer/GridContainer/Email.text,
    $VBoxContainer/GridContainer/Password.text)


func _on_RegisterButton_pressed():
  Firebase.Auth.signup_with_email_and_password(
    $VBoxContainer/GridContainer/Email.text,
    $VBoxContainer/GridContainer/Password.text)

func _on_FirebaseAuth_signup_succeeded(auth_info : Dictionary):
  print("signup succeeded: %s" % auth_info)
  get_tree().change_scene("res://Game/Game.tscn")


func _on_FirebaseAuth_login_succeeded(auth_info : Dictionary):
  print("login succeeded: %s" % auth_info)

  # save credentials for easier login next time
  var file = File.new()
  file.open(CREDENTIALS_PATH, File.WRITE)
  var credentials := {
    "email": $VBoxContainer/GridContainer/Email.text,
    "password": $VBoxContainer/GridContainer/Password.text
  }
  file.store_line(to_json(credentials))
  file.close()

```

```
  # get user record and role, and store it in User singleton
  # NOTE: This will crash if no user record has been added in collection users.
  User.id = auth_info["localid"]
  var users : FirestoreCollection = Firebase.Firestore.collection("users")
  users.get(User.id)
  var document : FirestoreDocument = yield(users, "get_document")
  User.role = document["doc_fields"]["role"] if document["doc_fields"].has("role") else "patient"
  User.first_name = document["doc_fields"]["first_name"] if document["doc_fields"].has("first_name") else ""
  User.last_name = document["doc_fields"]["last_name"] if document["doc_fields"].has("last_name") else ""

  # get extra data & start next scene depending on role
  match User.role:
    "clinician":
      get_tree().change_scene("res://Clinician/Clinician.tscn")
    "patient":
      # load patient specific fields
      User.dob = document["doc_fields"]["dob"] if document["doc_fields"].has("dob") else ""
      User.cpt_duration = document["doc_fields"]["cpt_duration"] if document["doc_fields"].has("cpt_duration") else 10
      User.cpt_pressure = document["doc_fields"]["cpt_pressure"] if document["doc_fields"].has("cpt_pressure") else 15
      User.game_state = document["doc_fields"]["game_state"] if document["doc_fields"].has("game_state") else {}
      get_tree().change_scene("res://Game/Game.tscn")
    _:  # any other role can play the game
      get_tree().change_scene("res://Game/Game.tscn")


func _on_FirebaseAuth_login_failed(code, message : String):
  print("login failed: %d %s" % [code, message])
  match message:
    "INVALID_EMAIL":
      $VBoxContainer/ErrorLabel.text = "Invalid email. Please correct it and try again."
    "INVALID_PASSWORD":
      $VBoxContainer/ErrorLabel.text = "Invalid password. Please change it and try again."
    "EMAIL_NOT_FOUND":
      $VBoxContainer/ErrorLabel.text = "Email not found. Please register an account first."
    "USER_DISABLED":
      $VBoxContainer/ErrorLabel.text = "This user account has been disabled."
    "WEAK_PASSWORD : Password should be at least 6 characters":
      $VBoxContainer/ErrorLabel.text = "Weak password. Please make it at least 6 characters long and try again."
    _:
      $VBoxContainer/ErrorLabel.text = message

func _on_developer_mode_pressed():
  get_tree().change_scene("res://Game/Game.tscn")
  pass # Replace with function body.
```

I.2 File: Clinician.gd

```
# @author Dr. Christof
extends Control

onready var patient_table : Tree =
    $MarginContainer/VBoxContainer/HBoxContainer/PatientTable
var patient_id : Dictionary = {}

func _ready():
  # query all patients
  var query : FirestoreQuery = FirestoreQuery.new().from("users").
    \
    where("role", FirestoreQuery.OPERATOR.EQUAL,
    "patient").order_by("first_name",
    FirestoreQuery.DIRECTION.ASCENDING).limit(100)
  var patient_docs : Array =
    yield(Firebase.Firestore.query(query), "result_query")

  # populate PatientTable and patient_id Dictionary with query
    results
  var root = patient_table.create_item()
  for patient_doc in patient_docs:
    var patient = patient_doc["document"]["fields"]
    var first_name : String = patient["first_name"]["stringValue"]
    if patient.has("first_name") else ""
    var last_name : String = patient["last_name"]["stringValue"]
    if patient.has("last_name") else ""
    var dob : String = patient["dob"]["stringValue"] if
    patient.has("dob") else ""

    var row = patient_table.create_item(root)
    row.set_text(0, first_name)
    row.set_text(1, last_name)
    row.set_text(2, dob)

    var name_parts : PoolStringArray =
    patient_doc["document"]["name"].split("/")
    patient_id[row] = name_parts[name_parts.size() - 1]

  # TODO load session data from sessions and populate Activity grid


func _on_PatientTable_item_selected():
  # enable change button, disable save button
  $MarginContainer/VBoxContainer/Buttons/ChangeButton.disabled =
    false
  $MarginContainer/VBoxContainer/Buttons/SaveButton.disabled = true

  # load user document for selected patient and update text fields
  var user_id : String = patient_id[patient_table.get_selected()]
  var users : FirestoreCollection =
    Firebase.Firestore.collection("users")
  users.get(user_id)
  var document : FirestoreDocument = yield(users, "get_document")

  var first_name : String = document["doc_fields"]["first_name"]
    if document["doc_fields"].has("first_name") else ""
  var last_name : String = document["doc_fields"]["last_name"] if
    document["doc_fields"].has("last_name") else ""
  var dob : String = document["doc_fields"]["dob"] if
    document["doc_fields"].has("dob") else ""
  var cpt_duration : int = document["doc_fields"]["cpt_duration"]
    if document["doc_fields"].has("cpt_duration") else 10
  var cpt_pressure : int = document["doc_fields"]["cpt_pressure"]
    if document["doc_fields"].has("cpt_pressure") else 15


    $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/First
    = first_name

    $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/LastN
    = last_name

    $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/Dob.
    = dob

    $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/CptD
    = String(cpt_duration)

    $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/CptP
    = String(cpt_pressure)


func _on_PlayButton_pressed():
  # if patient selected, update User
  if patient_table.get_selected():
    User.id = patient_id[patient_table.get_selected()]
    var users : FirestoreCollection =
    Firebase.Firestore.collection("users")
    users.get(User.id)
    var document : FirestoreDocument = yield(users, "get_document")

    User.role = document["doc_fields"]["role"] if
```

```
    document["doc_fields"].has("role") else "patient"
    User.first_name = document["doc_fields"]["first_name"] if
    document["doc_fields"].has("first_name") else ""
    User.last_name = document["doc_fields"]["last_name"] if
    document["doc_fields"].has("last_name") else ""
    User.dob = document["doc_fields"]["dob"] if
    document["doc_fields"].has("dob") else ""
    User.cpt_duration = document["doc_fields"]["cpt_duration"] if
    document["doc_fields"].has("cpt_duration") else 10
    User.cpt_pressure = document["doc_fields"]["cpt_pressure"] if
    document["doc_fields"].has("cpt_pressure") else 15
    User.game_state = document["doc_fields"]["game_state"] if
    document["doc_fields"].has("game_state") else {}

  get_tree().change_scene("res://Game/Game.tscn")


func _on_ChangeButton_pressed():
  # disable change button, enable save button
  $MarginContainer/VBoxContainer/Buttons/ChangeButton.disabled =
    true
  $MarginContainer/VBoxContainer/Buttons/SaveButton.disabled =
    false

  # make text fields editable

    $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/Patient
    = true

    $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/Patient
    = true

    $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/Patient
    = true

    $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/Patient
    = true

    $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/Patient
    = true


func _on_SaveButton_pressed():
  # enable change button, disable save button
  $MarginContainer/VBoxContainer/Buttons/ChangeButton.disabled =
    false
  $MarginContainer/VBoxContainer/Buttons/SaveButton.disabled = true


# make text fields non-editable

  $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/First
  = false

  $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/LastN
  = false

  $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/Dob.
  = false

  $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/CptD
  = false

  $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/CptF
  = false

# store patient document
var selected_row : TreeItem =
  $MarginContainer/VBoxContainer/HBoxContainer/PatientTable.get_selected()
var user_id : String = patient_id[selected_row]
var users : FirestoreCollection =
  Firebase.Firestore.collection("users")
var up_task : FirestoreTask = users.update(user_id, { \
  'first_name':
  $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/First
  \
  'last_name':
  $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/LastN
  \
  'dob':
  $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/Dob.
  \
  'cpt_duration':
  int($MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/
  \
  'cpt_pressure':
  int($MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/
  \
  })
var document : FirestoreDocument = yield(up_task,
  "task_finished")

# update row in PatientTable
selected_row.set_text(0,
  $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/First
```

```
selected_row.set_text(1,
  $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/LastName.text)
selected_row.set_text(2,
  $MarginContainer/VBoxContainer/HBoxContainer/VBoxContainer/PatientDetails/Dob.text)
```

I.3 File: User.gd

```
extends Node

# singleton object to access the details of the current user
var id : String = "test001"
var role : String = "patient"
var first_name : String = "Player"
var last_name : String = "Test"
var dob : String = "01/01/2012"
var cpt_duration : int = 10
var cpt_pressure : int = 16

var game_state : Dictionary = {
  "Achievements" : {
    "all_collect" : false,
    "woops": false,
    "go_away": false,
    "more_level": false,
    "boss_fight": false,
    "breath": false
 },
    "coins" : 0,
    "time_left" : 600,
    "stats" : { "sideways_speed": 2.5, "forward_speed": 2.0,
    "fire_damage": 1.0 },
    "items" : { "FireUpgrade1": true, "PinkDragonSkin": true }
}


# game state, to be stored in gameState after gameplay
var coins : int = 0
var time_left : int = 0# play time left for today in sec
var hit_points : int = 3

# Achievement System
#var all_collect : bool = false
#var woops : bool = false
#var go_away: bool = false
#var more_level: bool = false
#var boss_fight: bool = false
#var breath: bool = false
```

I.4 File: Player.gd & Controller.gd

```
# @author Dr. Christof
extends KinematicBody

export(float) var sideways_speed : float = 2.5
export(float) var forward_speed : float = 1.5
export(float) var fire_damage : float = 1.0  # damage dealt per
    second
export(bool) var stopped : bool = false
var fire_is_on : bool = false

func _ready():
  #fire_damage = User.game_state["sideways_speed"]
  #fire_damage = User.game_state["forward_speed"]
  #fire_damage = User.game_state["fire_damage"]
  pass



func _process(delta):
  # if playe ris stopped, e.g. at HuffCoughHurdle, then deactivate
    fire and ignore Controller input
  if stopped:
    fire_is_on = false
    $Mouth/Fire.emitting = false
    $FireCollisionTimer.stop()
    $FireSFX.stop()
    return

  move_and_slide(Vector3(Controller.movement * sideways_speed, 0,
    -forward_speed), Vector3(0, 1, 0))

  if Controller.breath > 5 and not fire_is_on:
    User.game_state["Achievements"]["breath"] = true
    fire_is_on = true
    $Mouth/Fire.emitting = true
    $FireCollisionTimer.start()
    $FireSFX.play()

  if not Controller.breath > 5 and fire_is_on:
    fire_is_on = false
    $Mouth/Fire.emitting = false
    $FireCollisionTimer.stop()
    $FireSFX.stop()
  pass

var FireCollider = preload("res://Game/Player/FireCollider.tscn")

func _on_FireCollisionTimer_timeout():
  var c = FireCollider.instance()
  c.init($Mouth.global_transform, 0.3,
    $Mouth/Fire.initial_velocity, $Mouth/Fire.lifetime,
    fire_damage * $FireCollisionTimer.wait_time)
  get_parent().add_child(c)
```

```
# @author Dr. Christof
extends Control

export(float) var movement : float = 0.0
export(float) var breath : float = 0.0


func _ready():
  # TODO add fallback for using controller in non-web context for
    testing
  if not OS.has_feature('JavaScript'):
    return

  # first requestDevice to allow user to connect it through
    browser dialogue
  # use this to set JavaScript variable "device"
  # TODO set filters to request only the BreatheHero device, e.g.
    filters: [{ vendorId: 0x2341 }]
  var resp = JavaScript.eval("""
    var device;

    navigator.usb.requestDevice({ filters: [] })
    .then(selectedDevice => {
    device = selectedDevice;
    })
    """, true)

  # waiting for device to be connected
  var try := 0
  while not resp:
    yield(get_tree().create_timer(1.0), "timeout")
    resp = JavaScript.eval("device.toString()", true)
    $Label.text = "Try %d, controller says: %s" % [try, resp]
    try += 1

  #TODO finish implementation, see https://web.dev/usb/
```

```
36 func _process(delta):
37   # input movement
38   if Input.is_action_pressed("move_left"):
39     movement = -1.0
40   elif Input.is_action_pressed("move_right"):
41     movement = +1.0
42   else:
43     movement = 0
44
45   # input breath
46   if Input.is_action_pressed("breathe_fire"):
47     breath = 10
48   else:
49     breath = 0
```

I.5 File: Pickup.gd

```
# @author Dr. Christof
tool
extends Area

export(float, 0, 20, 0.5) var hit_points : float = 0.0
export(int) var coin_value : int = 0
onready var player : Node =
    get_tree().get_root().find_node("Player", true, false)

func _physics_process(delta):
  # if pickup doesn't have hit_points then it is not frozen and
    can be picked up
  if hit_points == 0:
    $Ice.visible = false
    collision_layer = 4  # on "pickup" layer
    return

  # add Ice to frozen coin
  collision_layer = 8  # on "enemy" layer
  $Ice.scale.x = clamp(hit_points / 10, 0.5, 1)
  $Ice.scale.y = clamp(hit_points / 10, 0.5, 1)
  $Ice.scale.z = clamp(hit_points / 10, 0.5, 1)
  $Ice.visible = true


func _on_Pickup_body_entered(body):
  if body.get_instance_id() == player.get_instance_id():
  # cannot pick up pickup with HPs as it is "frozen"
    if hit_points > 0:
      return

    # collect the pickup
    $SFX.play()
    visible = false
    User.coins += coin_value

    # refresh UI
    var ui = get_tree().get_root().find_node("UI", true, false)
    if(ui):
      ui.find_node("CoinsCounter").text = "Coins: %d" % User.coins
  else:
    pass

func _on_Pickup_area_entered(area):
  #print("_on_Pickup_area_entered: Pickup %s on layer %d with mask
    %d; %s on layer %d with mask %d" \
  # % [name, collision_layer, collision_mask, area.name,
    area.collision_layer, area.collision_mask])

  # if hit by fire, then reduce hit_points
  if area.collision_layer == 16:
    hit_points = max(0, hit_points - area.damage_points)

    # while hit_points > 0 ice is visible and pickup is on "enemy"
    layer rather than "pickup" layer
    $Ice.visible = hit_points > 0
    collision_layer = 8 if hit_points > 0 else 4
```

I.6 File: HuffCoughHurdle.gd

```
# @author Dr. Christof
extends Area


onready var player : Node =
    get_tree().get_root().find_node("Player", true, false)
onready var message_label : Label =
    get_tree().get_root().find_node("Message", true, false)

var waiting_for_cough := false
var cough_in_progress := false
var num_coughs := 0

func _physics_process(delta):
  if not waiting_for_cough:
    return

  var volume : float = AudioServer.get_bus_peak_volume_left_db(1,
    0)

  if volume > -10.0 or Input.is_action_pressed("breathe_fire"):
    print("cough volume=%f" % volume)
    cough_in_progress = true
  elif cough_in_progress:
    cough_in_progress = false
    num_coughs += 1

  if num_coughs == 3:
    #TODO dismantle hurdle, release reward
    waiting_for_cough = false
    player.stopped = false

    #set a user achievement
    User.game_state["Achievements"]["go_away"] = true


  #TODO add timeout to continue game in case no coughing is
    detected for a while


func _on_HuffCoughHurdle_body_entered(body):
  if body.get_instance_id() == player.get_instance_id():
    player.stopped = true
    waiting_for_cough = true
    $Microphone.play()

    message_label.text = "Cough 3x to break the obstacle!"
    message_label.visible = true
    yield(get_tree().create_timer(4.0), "timeout")
    message_label.visible = false
  else:
    pass


```

I.7 File: Level.gd

```gdscript
# @author Dr. Christof, Xiaoyan Zhou
extends Spatial

export(String) var chapter_filename : String =
    "res://Game/Chapter/Chapter1.tscn"
export(Resource) var background_music : Resource
export(int) var goal_coins : int = 0
onready var player : Node =
    get_tree().get_root().find_node("Player", true, false)
onready var pickup : Node =
    get_tree().get_root().find_node("Pickups", true, false)

var timer : int = 1
var start_coins : int = User.coins
var level_coins : int

#a special case achievement
var breath : bool = User.game_state["Achievements"]["breath"]
var boss_fight : bool =
    User.game_state["Achievements"]["boss_fight"]
var woops : bool = User.game_state["Achievements"]["woops"]
var all_collect : bool =
    User.game_state["Achievements"]["all_collect"]
var go_away : bool = User.game_state["Achievements"]["go_away"]
var more_level : bool =
    User.game_state["Achievements"]["more_level"]


func _ready():
  User.hit_points = 3
  #count the total level coins
  _level_coin_counter()

  BackgroundMusic.stream = background_music
  BackgroundMusic.play()

  $UI/CoinsCounter.text = "Coins: %d" % User.coins

  # show goal
  # if enter it as a number smaller than 0, boss fight!
  if goal_coins > 0:
    $UI/Message.text = "Goal:\nCollect %d coins!" % goal_coins
  if goal_coins == 0:
    $UI/Message.text = "Goal:\nReach the finish line!"
  if goal_coins < 0:
    $UI/Message.text = "Goal:\nFight the boss!"

  # hide message after 4 seconds
  yield(get_tree().create_timer(1.0), "timeout")
  $UI/Message.visible = false
  #set timer
  while true:
    yield(get_tree().create_timer(1.0), "timeout")
    $UI/TimerCounter.text = "Timer: %d" % timer
    timer += 1




# check if the achievements is completed
func _process(delta):
  # set player hp
  _get_hp()
  _achievements_handler()
  pass

func _on_FinishLine_body_entered(body):
  # check level completion
  if body.get_instance_id() == player.get_instance_id():
    var coins_collected : int = User.coins - start_coins
    if coins_collected >= goal_coins:
      User.game_state["Achievements"]["more_level"] =  true
      $UI/Message.text = "Level complete!"
    else:
      $UI/Message.text = "Level incomplete:\nYou collected %d out
    of %d coins" % [coins_collected, goal_coins]
    $UI/Message.visible = true
  # go back to main menu after 4 seconds
    player.stopped = true
    #set play time
    User.time_left += timer
    yield(get_tree().create_timer(3.0), "timeout")
    get_tree().change_scene(chapter_filename)
  else:
    pass

#count the full collect coins number in the level
func _level_coin_counter():
  for i in range (0, pickup.get_child_count()):
    var coin_v = pickup.get_child(i).coin_value
    level_coins += coin_v
```

```
  pass

#check if the achievement achieved
func _all_achievement(user_achievement, achievement_status):
  # all collect, it is a bit special so it is here
  if user_achievement && !achievement_status:
    if achievement_status == woops: woops = true
    if achievement_status == breath: breath = true
    if achievement_status == boss_fight: boss_fight = true
    if achievement_status == go_away: go_away = true
    if achievement_status == all_collect: all_collect = true

    $UI/Message.visible = true
    $UI/Message.text = "Achievement Unlocked!"
    yield(get_tree().create_timer(0.5), "timeout")
    $UI/Message.visible = false

func _achievements_handler():
  #dynamic claculating the pickup for achievements
  var pickedup = User.coins - start_coins
  User.game_state["Achievements"]["all_collect"] = pickedup ==
    level_coins
  #achievements all collect
  _all_achievement(User.game_state["Achievements"]["breath"],
    breath)
  _all_achievement(User.game_state["Achievements"]["boss_fight"],
    boss_fight)
  _all_achievement(User.game_state["Achievements"]["woops"], woops)
  _all_achievement(User.game_state["Achievements"]["go_away"],
    go_away)
  _all_achievement(User.game_state["Achievements"]["all_collect"],
    all_collect)
  _all_achievement(User.game_state["Achievements"]["more_level"],
    more_level)
  pass

func _get_hp():
  if User.hit_points == 3:
    pass
  elif User.hit_points == 2:
    $UI/GridContainer/hp3.visible = false
  elif User.hit_points == 1:
    $UI/GridContainer/hp2.visible = false
  elif User.hit_points == 0:
    $UI/GridContainer/hp.visible = false
    player.stopped = true
    $UI/Message.text = "Oh No!\nYou have caught by the Slime!"
    $UI/Message.visible = true
    player.stopped = true
    yield(get_tree().create_timer(1.0), "timeout")
    get_tree().change_scene("res://Game/GameOver.tscn")
  pass
```

I.8 File: LittleSlime.gd

```
# @author Xiaoyan Zhou
extends KinematicBody

export(bool) var left : bool = true
export(bool) var right : bool = false
onready var message_label : Label =
    get_tree().get_root().find_node("Message", true, false)
onready var player : Node =
    get_tree().get_root().find_node("Player", true, false)

func _ready():
  pass

func _process(delta):
  _move_left_right()

# let the slime move left and right, change direction when it
    collided
func _move_left_right():
  if left:
    move_and_slide(Vector3(-4,0,0), Vector3(0,0,0), false, 1)
    if is_on_wall():
      _game_over()

      move_and_slide(Vector3(4,0,0), Vector3(0,0,0), false, 1)
      left = false
      right = true

  if right:
    move_and_slide(Vector3(4,0,0), Vector3(0,0,0), false, 1)
    if is_on_wall():
      _game_over()

      move_and_slide(Vector3(-4,0,0), Vector3(0,0,0), false, 1)
      left = true
      right = false

# if the slime touch the player, change scene and game over
func _game_over():
  if get_slide_collision(0).get_collider_id() ==
    player.get_instance_id():
    #player.stopped = true
    #message_label.text = "Oh No!\nYou have caught by the Slime!"
    #message_label.visible = true
    #player.stopped = true
    #yield(get_tree().create_timer(1.0), "timeout")
    User.hit_points -= 1

    if User.hit_points == 0:
      User.game_state["Achievements"]["woops"] = true
    #yield(get_tree().create_timer(1.0), "timeout")
    #get_tree().change_scene("res://Game/GameOver.tscn")
  else:
    pass
```

I.9 File: Achievement.gd

```
# @author Xiaoyan Zhou
extends Control
onready var achievements : Node =
    get_tree().get_root().find_node("GridContainer", true, false)


# Called when the node enters the scene tree for the first time.
func _ready():
  for i in range (0,achievements.get_child_count()):
    achievements.get_child(i).visible = false

  pass # Replace with function body.

func _process(delta):
  _achievements_check()
  pass

#check achievements
func _achievements_check():

  if User.game_state["Achievements"]["all_collect"]:
    get_node("VBoxContainer/GridContainer/AllCollect").visible =
    true
  if User.game_state["Achievements"]["boss_fight"]:
    get_node("VBoxContainer/GridContainer/BossFight").visible =
    true
  if User.game_state["Achievements"]["woops"]:
    get_node("VBoxContainer/GridContainer/Woops").visible = true
  if User.game_state["Achievements"]["go_away"]:
    get_node("VBoxContainer/GridContainer/GoAway").visible = true
  if User.game_state["Achievements"]["more_level"]:
    get_node("VBoxContainer/GridContainer/MoreLevel").visible =
    true
  if User.game_state["Achievements"]["breath"]:
    get_node("VBoxContainer/GridContainer/Breath").visible = true
# Called every frame. 'delta' is the elapsed time since the
    previous frame.
#func _process(delta):


func _on_BackButton_pressed():
  get_tree().change_scene("res://Game/Game.tscn")
  pass # Replace with function body.
```

I.10 File: BossSlime.gd & LittleSlime_bossfight.gd

```gdscript
# @author: Xiaoyan Zhou
extends KinematicBody

export(bool) var left : bool = true
export(bool) var right : bool = false
export(bool) var bossFight : bool = true
var LittleSlime_bossfight =
    preload("res://Game/enemies/Slime/LittleSlime_bossfight.tscn")
onready var UI : Node = get_tree().get_root().find_node("UI",
    true, false)
onready var player : Node =
    get_tree().get_root().find_node("Player", true, false)
onready var pick_up : Node =
    get_tree().get_root().find_node("Pickups", true, false)
var coins = preload("res://Game/Pickup/Coin/GoldCoin.tscn")
# set boss fight
func _ready():
  if bossFight:
    $Generate_littleSlime.start(10)
    player.forward_speed = 0
  pass

# boss would move left and right
func _process(delta):
  _move_left_right()
  if !bossFight && !User.game_state["Achievements"]["boss_fight"]:
    queue_free()
    _generate_coins()

    #achievements
    User.game_state["Achievements"]["boss_fight"] = true

# move left and right, same logic as the other slime
func _move_left_right():
  if left:
    move_and_slide(Vector3(-3,0,0), Vector3(0,0,0), false, 1)
    if is_on_wall():
      move_and_slide(Vector3(3,0,0), Vector3(0,0,0), false, 1)
      left = false
      right = true

  if right:
    move_and_slide(Vector3(3,0,0), Vector3(0,0,0), false, 1)
    if is_on_wall():
      move_and_slide(Vector3(-3,0,0), Vector3(0,0,0), false, 1)
      left = true
      right = false

# generate a little slime in the scene
func _generate_little_slime():
  var slime = LittleSlime_bossfight.instance()
  get_parent().add_child(slime)
  var parent_position = get("translation")
  slime.set("translation", Vector3(parent_position.x,
    parent_position.y-1.5, parent_position.z+5))
  pass

func _on_Generate_littleSlime_timeout():
  _generate_little_slime()
  pass # Replace with function body.

# generate coins
func _generate_coins():
  pick_up.visible = true
  pass
```

```gdscript
# @author: Xiaoyan Zhou
extends KinematicBody
export(float, 0, 20, 0.5) var hit_points : float = 0.0
onready var player : Node =
    get_tree().get_root().find_node("Player", true, false)
onready var boss_slime : Node =
    get_tree().get_root().find_node("BossSlime", true, false)
onready var message_label : Label =
    get_tree().get_root().find_node("Message", true, false)
onready var pick_up : Node =
    get_tree().get_root().find_node("Pickups", true, false)

var speed : int = 3
var waiting_for_cough := false
var cough_in_progress := false
var num_coughs := 0
var deleted: bool = false

# Called when the node enters the scene tree for the first time.
func _ready():
  collision_layer = 8
  collision_mask = 2
  pass # Replace with function body.
```

```

func _physics_process(delta):
  _cough_handler(delta)

  if hit_points == 0:
    rotate_y(0.1)

  if is_on_wall() && hit_points > 0:
    if get_slide_collision(0).get_collider_id() ==
    player.get_instance_id():
      _game_over()
      pass

func _process(delta):
  #free the resource
  var player_position = player.get("translation")
  var slime_position = translation
  if slime_position.z > (player_position.z+5) || slime_position.z
    < (player_position.z-35):
    queue_free()


  move_and_slide(Vector3(0,0,speed))

# the func for hit the slime with flame
func _on_Area_area_entered(area):
  # if hit by fire, then reduce hit_points
  if area.collision_layer == 16:
    hit_points = max(0, hit_points - area.damage_points)
  pass # Replace with function body.

# let the player cough
func _cough_handler(delta):
  if not waiting_for_cough:
    return

  var volume : float = AudioServer.get_bus_peak_volume_left_db(1,
    0)

  if volume > -10.0 or Input.is_action_pressed("breathe_fire"):
    print("cough volume=%f" % volume)
    cough_in_progress = true
  elif cough_in_progress:
    cough_in_progress = false
    num_coughs += 1

  if num_coughs == 3:
    #TODO dismantle hurdle, release reward
    waiting_for_cough = false
    player.stopped = false
    speed = -10
  #TODO add timeout to continue game in case no coughing is
    detected for a while

func _on_Area_body_entered(body):
  if is_on_wall() && get_slide_collision(0).get_collider_id() ==
    player.get_instance_id() && hit_points == 0:
    speed = 0
    player.stopped = true
    waiting_for_cough = true
    $Microphone.play()

    message_label.text = "Cough 3x times!\nShoot the little slime
    back!"
    message_label.visible = true
    yield(get_tree().create_timer(4.0), "timeout")
    message_label.visible = false

  if is_on_wall() && get_slide_collision(0).get_collider_id() ==
    boss_slime.get_instance_id() && hit_points == 0:
    boss_slime.bossFight = false
    player.forward_speed = 2
    pick_up.visible = true
    queue_free()

# if the slime touch the player, change scene and game over
func _game_over():
  if get_slide_collision(0).get_collider_id() ==
    player.get_instance_id():
    #player.stopped = true
    #message_label.text = "Oh No!\nYou have caught by the Slime!"
    #message_label.visible = true
    #player.stopped = true
    #yield(get_tree().create_timer(1.0), "timeout")
    User.hit_points -= 1
    queue_free()
    #yield(get_tree().create_timer(1.0), "timeout")
    #get_tree().change_scene("res://Game/GameOver.tscn")
  else:
    pass
```

Appendix J

Ethical Checklist

This form must be attached to the dissertation as an appendix.

UNIVERSITY OF BATH

Department of Computer Science

12-Point Ethics Checklist for UG and MSc Projects

Student XIAOYAN ZHOU

Academic Year or Project Title BREATHEHERO

Supervisor CHRISTOF LUTTHEROTH

Does your project involve people for the collection of data other than you and your supervisor(s)? YES / ~~NO~~

If the answer to the previous question is YES, you need to answer the following questions, otherwise you can ignore them.

This document describes the 12 issues that need to be considered carefully before students or staff involve other people ('participants' or 'volunteers') for the collection of information as part of their project or research. Replace the text beneath each question with a statement of how you address the issue in your project.

1. *Will you prepare a Participant Information Sheet for volunteers?* YES / ~~NO~~

   This means telling someone enough in advance so that they can understand what is involved and why – it is what makes informed consent informed.

2. *Will the participants be informed that they could withdraw at any time?* YES / ~~NO~~

   All participants have the right to withdraw at any time during the investigation, and to withdraw their data up to the point at which it is anonymised. They should be told this in the briefing script.

3. *Will there be any intentional deception of the participants?* YES / ~~NO~~

   Withholding information or misleading participants is unacceptable if participants are likely to object or show unease when debriefed.

4. *Will participants be de-briefed?* YES / ~~NO~~

   The investigator must provide the participants with sufficient information in the debriefing to enable them to understand the nature

of the investigation. This phase might wait until after the study is completed where this is necessary to protect the integrity of the study.

5. *Will participants voluntarily give informed consent?* YES / ~~NO~~
Participants MUST consent before taking part in the study, informed by the briefing sheet. Participants should give their consent explicitly and in a form that is persistent –e.g. signing a form or sending an email. Signed consent forms should be kept by the supervisor after the study is complete. If your data collection is entirely anonymous and does not include collection of personal data you do not need to collect a signature. Instead, you should include a checkbox, which must be checked by the participant to indicate that informed consent has been given.

6. *Will the participants be exposed to any risks greater than those encountered in their normal work life (e.g., through the use of non-standard equipment)?* ~~YES~~ / NO
Investigators have a responsibility to protect participants from physical and mental harm during the investigation. The risk of harm must be no greater than in ordinary life.

7. *Will you be offering any incentive to the participants?* ~~YES~~ / NO
The payment of participants must not be used to induce them to risk harm beyond that which they risk without payment in their normal lifestyle.

8. *Will you be in a position of authority or influence over any of your participants?* ~~YES~~ / NO
A position of authority or influence over any participant must not be allowed to pressurise participants to take part in, or remain in, any experiment.

9. *Will any of your participants be under the age of 16?* YES / ~~NO~~
Parental consent is required for participants under the age of 16.

10. *Will any of your participants have an impairment that will limit Their understanding or communication?* ~~YES~~ / NO
Additional consent is required for participants with impairments.

11. *Will the participants be informed of your contact details?* YES / ~~NO~~
All participants must be able to contact the investigator after the investigation. They should be given the details of the Supervisor as part of the debriefing.

12. ***Will you have a data management plan for all recorded data?*** YES / ~~NO~~

Personal data is anything which could be used to identify a person, or which can be related to an identifiable person. All personal data (hard copy and/or soft copy) should be anonymized (with the exception of consent forms) and stored securely on university servers (not the cloud).