Dr. Abel's Adventures: A Game Based Application to Enhance Knowledge and Awareness About Communicable Diseases

Josef Rex O. Armenta
Bachelor of Science in Information Technology

Jefen B. De Villa
Bachelor of Science in Information Technology

Senior project submitted to the faculty of the
Department of Computer Science
College of Computer Studies, Ateneo de Naga University
in partial fulfillment of the requirements for their respective
Bachelor of Science degrees

Project Advisor: Raphael Henry M. Garay
Adrian Leo T. Pajarillo
Maricel L. Fornaleza
Frederick D. Olaño

Month Day, 2022
Naga City, Philippines

Keywords: Disease Prevention and Control, Communicable Diseases, Game-Based Learning

The Senior Project entitled

Dr. Abel's Adventures: A Game Based Application to Enhance Knowledge and Awareness About Communicable Diseases

developed by

Josef Rex O. Armenta

Bachelor of Science in Information Technology

Jefen B. De Villa

Bachelor of Science in Information Technology

and submitted in partial fulfillment of the requirements of their respective Bachelor of Science degrees has been rigorously examined and recommended for approval and acceptance.

Adrian Leo T. Pajarillo
Panel Member
Date signed: 9 July 2023

Maricel L. Fornaleza
Panel Member
Date signed: 12 July 2023

for:

Frederick D. Olano
Panel Member
Date signed: 17 July 2023

Raphael Henry M. Garay
Project Advisor
Date signed: 04 July 2023

The Senior Project entitled

Dr. Abel's Adventures: A Game Based Application to Enhance Knowledge and Awareness About Communicable Diseases

developed by

Josef Rex O. Armenta

Bachelor of Science in Information Technology

Jefen B. De Villa

Bachelor of Science in Information Technology

and submitted in partial fulfillment of the requirements of their respective Bachelor of Science degrees is hereby approved and accepted by the Department of Computer Science, College of Computer Studies, Ateneo de Naga University.

Adrian Leo T. Pajarillo
Chair, Department of Computer Science
Date signed: 17 July 2023

Joshua C. Martinez, MIT
Dean, College of Computer Studies
Date signed: 17 July 2023

Declaration of Original Work

We declare that the Senior Project entitled

Dr. Abel's Adventures: A Game Based Application to Enhance Knowledge and Awareness About Communicable Diseases

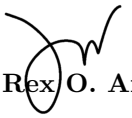
which we submitted to the faculty of the

Department of Computer Science, Ateneo de Naga University

is our own work. To the best of our knowledge, it does not contain materials published or written by another person, except where due citation and acknowledgement is made in our senior project documentation. The contributions of other people whom we worked with to complete this senior project are explicitly cited and acknowledged in our senior project documentation.

We also declare that the intellectual content of this senior project is the product of our own work. We conceptualized, designed, encoded, and debugged the source code of the core programs in our senior project. The source code of third party APIs and library functions used in my program are explicitly cited and acknowledged in our senior project documentation. Also duly acknowledged are the assistance of others in minor details of editing and reproduction of the documentation.

In our honor, we declare that we did not pass off as our own the work done by another person. We are the only persons who encoded the source code of our software. We understand that we may get a failing mark if the source code of our program is in fact the work of another person.



Josef Rex O. Armenta
4 - Bachelor of Science in Information Technology

Jefen B. De Villa
4 - Bachelor of Science in Information Technology

This declaration is witnessed by:

Raphael Henry M. Garay
Project Advisor

Dr. Abel's Adventures: A Game Based Application to Enhance Knowledge and Awareness About Communicable Diseases

by

Josef Rex O. Armenta, and Jefen B. De Villa

Project Advisor: Raphael Henry M. Garay

Department of Computer Science

EXECUTIVE SUMMARY

Health Literacy refers to "the people's ability to find, understand, and use information and services to inform health-related decisions and actions for themselves and others" [9]. With cases of new and highly contagious diseases such as COVID-19 appearing, and a lot of misinformation about the disease to top it, being health literate should be necessary to avoid falling victim to the disease. In the Philippines, initiatives to increase the people's health literacy are already observable as early as elementary, by incorporating health topics in subjects such as Science and health, and MAPEH. However the country's situation during the peak of COVID-19 pandemic shows that the information the people gained from the subjects were not applied properly during the pandemic. This study created the Game-based learning tool titled "Dr. Abel's Adventures" to help in improving the health literacy of the Grade 8 students of Camarines Sur National High School. The result of this study showed that the use of game-based learning tools has helped to improve the players' knowledge significantly, but the learning outcome it produces does not significantly differ with the existing teaching methods that is implemented to this day.

The late Jose P. Rizal once said, "*Ang Kabataan ang Pag-asa ng Bayan*" As a youth, it is our duty to carry on his vision. We dedicate this work not only to ourselves and our families but also to the youth of this nation.

ACKNOWLEDGEMENTS

We would like to express our sincere gratitude to Sir Raphael Henry M. Garay, our thesis advisor, for his support and understanding throughout the past few months. We truly appreciate his invaluable contribution in guiding us throughout the process of creating and writing this project.

We are also grateful to our partner institution, Camarines Sur National High School, led by Mr. Sulpicio C. Alferes and the following individuals: Ms. Catherine Olaes, the guidance counselor, Ms. Nimfa Regalado, the Head Teacher in MAPEH, and Ms. Vikki Neo for providing us with their unwavering help and support since the outset of this project. We are profoundly grateful for their collaborative contributions and assistance throughout the entire project duration.

We would also like to express our gratitude to our esteemed panelists, Mr. Adrian Leo T. Pajarillo, Ms. Maricel L. Fornaleza, and Mr. Frederick D. Olano, for their time and effort in evaluating our work and providing constructive criticism. Their comments and suggestions were extremely valuable and helped us improve the quality of our project.

Finally, we would like to thank our family, friends, and loved ones for their unwavering love, support, and encouragement throughout our academic pursuits. Their belief in us has been a constant source of strength and inspiration.

Thank you all for being a part of this incredible journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

Health plays an important role in the development of a person, a healthy emotional, mental, and physical development of a person provides positive outcomes in their lives as well as to their community [19]. Taking care of one's health is a part of lifestyle, and should not be limited and end after visiting doctors and hospitals [26]. The benefits of being healthy extends to everyone in the community, it reduces absenteeism in schools which increases the children's ability to learn, allows adults to function properly allowing them to fulfill their roles in the society, and decreasing the emergence of diseases in the community [7, 52].

From the COVID-19 pandemic that the world faced, disease prevention and control has become a popular topic all over the world [30]. The rapid spread of the global pandemic Coronavirus disease 2019 (COVID-19) has caused panic all around the world, negatively affecting many aspects such as education, the economy, and most notably, the people. Having multiple infections in a short period overwhelmed the affected countries, leaving them unable to provide healthcare, causing alarming concerns among the people [24].

The Philippines is one of the countries that suffered the effects of the pandemic, which was made worse by public panic. When the pandemic was announced in the country, some people began hoarding supplies (alcohol, medicine, etc.) leaving none for others [17]. Worse yet, hoarders exploit resources by reselling their goods at a higher price, making the situation even more difficult for the populace [40]. Situations such as this show the need for a proper enhancement of health literacy among the people in order to prevent other similar situations from happening. The president of

Philippine Medical Association, Jose P. Santiago identified the low level of health literacy among Filipinos as the primary cause of the spread of diseases. It stressed the importance of advancing people's health literacy. The paper Health Literacy for All Filipinos (2019) identified approaches to improve health literacy which are (1) Preventive and Promotive Approaches, which are done through teaching from schools and barangays, and (2) Curative and Rehabilitative Approaches, where consultations with professionals in the medical field are involved [22]. This study focused on the first approach which is the preventive and promotive approach that is done through teaching in schools. One of the problems in teaching health education is that there is no immediate way to test and prove the concepts taught to the learners. Topics that they have not encountered or experienced yet are hard for them to visualize or imagine and eventually get discarded from their memory as they find the information irrelevant because there are no current practical applications in their lives [45]. One way to address this problem is by implementing Game-Based Learning (GBL). Game-based learning is an instructional method incorporating educational topics into video games [13]. Game-based learning provides a game environment for the learners to interact with to increase learners' engagement and understanding of the topics [28]. Using game-based learning, it is possible to design a game environment to supplement the problem in visualization and relation of newly introduced health topics in real-life situations to help in knowledge retention and understanding the learners.

1.1 Project Context

The term "health literacy" refers to the literacy ability of a person, which is needed in making decisions about one's health in a variety of settings (home, community, and health clinic) as well across the three domains of the health continuum; mainly healthcare, disease prevention, and health promotion [25, 49]. In the Philippines, the initiative to develop and improve health literacy is observable in many subjects as early as kindergarten. The Department of Education (DepEd) even creates standards and competencies for five-year-old Filipino children. Kalusugang Pisikal at Pagpapaunlad sa Kakayahang Motor (Physical Health & Motor Development) was one of the core teachings which aims to know the importance of having a physically fit body to five-year-old kids [27]. Subjects like Science and Health and MAPEH (Music, Arts, Physical Education, and Health) are also one of many subjects that stretch beyond teaching, not just including how to stay fit but

also talks about different concepts involved in teaching health literacy. Despite the continuous efforts made by the government to teach health concepts, the Philippines still suffers greatly when experiencing a global health crisis. For that reason alone, health literacy calls for more than just the dissemination of new knowledge; it also calls for the growth of empowering personal abilities that permit engagement in a variety of actions that can safeguard and enhance health brought by new opportunities presented by health education and emerging technologies [25].

Game-Based Learning (GBL) has been around almost since technology took its leap toward innovation. In a study conducted by Hellerstedt A. & Mozelius P. entitled Game-Based Learning - A Long History (2019), they pointed out that game-based learning has had a high status throughout history. Game-based learning has been used to instruct politicians, military officers, and princesses [12]. Plato, an ancient Greek philosopher, proposed a constructive role for play in education, viewing it as a "first step on a ladder towards true knowledge." Educators such as Vit-torino da Feltre pursued this idea during the Renaissance. In the 17th century, John Comenius' systematic theory of education viewed the game (Ludus) as the ideal form of learning. Finally, in the 20th century, Jean Piaget and Lev Vygotsky introduced GBL as a pedagogical approach at the university level [50]. Today, many fields have adopted game-based learning to see if it helps improve the knowledge retention of its target audience. Game-based learning (GBL) refers to creating learning activities with game characteristics and principles inherent in the learning activities themselves [10]. Furthermore, designing learning activities that may gradually introduce concepts and lead users toward a common objective is another aspect of game-based learning, and it goes beyond simply creating games for students to play. [33]. While GBL does not seek to replace formal education (and should not), it can be an extremely beneficial supplement. By incorporating essential skills and abilities in play and games, GBL may be able to replace arduous 'homework' for more practical-oriented learning [50].

This study created a game-based learning system for Grade 8 students of Camarines Sur National High School (CSNHS) that teaches them about the different concepts of prevention and control of communicable diseases, that integrates the DepED MELC: K to 12 Most Essential Learning Competencies provided by the Department of Education (DepEd) in their MAPEH (Music, Arts, Physical Education, and Health) specifically in the Health component of the subject.

1.2 Purpose and Description

This study served as an educational tool that taught its players the concepts about the prevention and control of communicable diseases by simulating a game environment where they could interact to better visualize the topic being discussed. The reason for this method is to catch the players' focus to create an immersion in learning the topic and to make them understand it better so that they can easily relate their learning from the game if they encounter it in real-life situations.

The main purpose of this study is to help enhance the students' health literacy, particularly on the prevention and control of communicable diseases to avoid, if not lessen, the chances of another global health crisis like the COVID-19 pandemic in the country.

By the end of the study, it aims to determine if game-based learning was effective in teaching Health education, specifically about preventing and controlling communicable diseases, to Grade 8 students of Camarines Sur National High School (CSNHS).

1.3 Objectives

The main objective of the study was to develop a game-based learning application to enhance the knowledge and awareness of the Grade 8 students about the concepts concerning the prevention and control of communicable diseases and to assess whether the implementation of a game-based learning environment was an effective tool in teaching the students who are taking MAPEH subject. To achieve the main objective, the following specific objectives were accomplished;

- Developed a 2D game-based application to serve as a learning tool.
- Incorporated the topics from the Health component of the MAPEH (Music, Arts, Physical Education and Health) subject of the Department of Education (DepEd) MELC: K to 12 Most Essential Learning Competencies about prevention and control of communicable diseases in the developed game.
- Evaluated the usability and the effectiveness of the game-based application through usability testing.

1.4 Scope and Limitations

This study primarily focused on developing a game-based application to assist students in learning and enhance their understanding of the different concepts of prevention and control of communicable diseases. Specifically, the definition of communicable disease, the different kinds of germs and disease agents, e.g., bacteria, viruses, fungi, parasites, etc., the transmission of each type of communicable disease agent, the prevention needed to combat the different kinds of communicable disease agents, factors or causes of communicable diseases, signs and symptoms of common communicable diseases, effects of common communicable diseases and the misconceptions, myths, and beliefs about common communicable diseases.

The information integrated into the game-based learning environment is based on the Health component of MAPEH (Music, Arts, Physical Education, and Health) subject from the Department of Education's (DepEd) K to 12 Most Essential Learning Competencies (MELC) about prevention and control of communicable diseases, that is being taught in the 3rd Quarter which is in February of the school year 2022-2023. Furthermore, this study only focused on the prior content standard, which is the demonstration of understanding and principles in the prevention and control of communicable diseases for the attainment of individual wellness, and did not extend towards other content standards found in the DepEd MELC: K to 12 Most Essential Learning Competencies [36]. The Grade 8 students of Camarines Sur National High School are the subjects of this study and are limited to this grade level. Other Grade 8 students outside Naga City adopting the same curriculum guide and content standard provided by the Department of Education are also not included in this study. Finally, the game-based learning application is only supported on a mobile platform running an Android operating system.

Chapter 2

Review of Related Systems and Related Literature

This chapter presents various ideas, techniques and conclusions of the different literature and systems related to this capstone project. Those included in this chapter contribute to familiarizing the reader with significant and comparative data to the current study. Furthermore, example literature and systems intended to perform the same functions are discussed to form a solid foundation and acquire a better knowledge about the study.

2.1 The Need for Proper Health Literacy

The COVID 19 pandemic has brought some light to a major issue that the world is facing today, specifically how it demonstrated that low health literacy among a community is still and continuing an underappreciated public health issue. In the study conducted by Healio in the Philippines, which describes the health literacy levels of Filipinos at a national and sub-national level during 2018 and 2019, highlights that the nationwide prevalence of limited health literacy sits at 51.5% (with the confidence interval of 95%), while sub-national prevalence is at an estimated range of 48.2% to 65.4%. The study concluded that Filipino people still had limited health literacy, which varies across health dimensions, sub-national groupings, and socio-demographic characteristics [49]. According to Spring H. in his article Health Literacy and COVID-19 (2020), ever since the COVID 19 outbreak happened,

various information dissemination sources have been bombarded with information regarding COVID-19, and a substantial amount was incorrect [43]. The World Health Organization has also taken some initiative to combat this misinformation by creating a website with downloadable posters that helps strengthen public health information [29]. Though the government has taken some precautionary steps in minimizing the issue at hand, people are still holding out about the necessary precautions one must follow when dealing with the situation that the world is currently experiencing. Because of this, it is conclusive that there are still health literacy deficiencies in the general public. Given that, if a certain individual lacks a critical understanding of the problem, one can't distinguish between reality and fiction and may allow erroneous information to influence their actions. This may be destructive to the individual and society as a whole [43].

In a recent study made by Paakkari, L. & Okan, O. entitled COVID 19: Health Literacy is an Underestimated Opinion (2020), they identify a striking aspect when COVID-19 started to emerged, according to them;

- First, globally, health literacy is as important for the prevention of communicable diseases as it is for non-communicable diseases.
- Second, along with system preparedness, individual preparedness is key for solving complex real-life problems.

It's never been more important to stress applying a greater value when it comes to more critical health literacy to the masses, especially now, where bodies of knowledge regarding health, safety, and the pandemic are abundant and, at the same time, uncertain. In general, public health personnel assumes that by knowing about the risk factors of these infectious diseases, one can control and prevent epidemic consequences from occurring [1]. Health literacy, as we know, is a crucial tool to stay healthy and take informed and appropriate actions and prevent communicable diseases like COVID-19 from spreading. It is a good investment in education, communication, and society because it seeks to be sustainable and long-term [31].

2.2 Use of Technology in Education

The technology industry is one of the fastest growing and evolving fields in our society as time goes by. Technology is being developed continuously by different experts and used for various

purposes like educational means and distribution of information for certain topics. Because of these technological advancements, many changes also happened in education methods, like the integration of Information and Communication Technologies (ICT) for instructions [13]. The computer is one of the products of the advancement of technology, which also allowed the spreading of information to be much more accessible and easier through different gadgets. With the advancement of mobile device capabilities, many students cannot imagine their studies at the university without gadgets [14].

With this growth in technology, and devices being more accessible to the public, new teaching methods like video games have been introduced. They are being used for students as an educational tool. During the previous year, the use of gadgets as a learning tool at school has grown because many students are always using their mobile devices [14]. In addition to mobile devices, computers, and computer games are also being used for teaching the elementary students of this generation. According to Mayer (2019), researchers suggest that using first-person shooting games for training perceptual attention skills and using spatial puzzle games to train two-dimensional mental rotation skills shows promising results in the student's skill [20]. However, as technology improves, some consequences can be felt by students with low economic status. Additional descriptive analyses suggested that game-based learning may be less beneficial for students with low socioeconomic status than students with high socioeconomic status [4].

2.3 Game-Based Learning

Video games have become an important part of our culture especially to the children [16]. Unlike before, where it was identified as a distraction from doing more "significant" things or works, it is now recognized by many organizations as a powerful medium that can potentially help in learning and giving out information to the masses [16]. Lately, "game-based learning" has become a favorite subject for academic research and development (R&D), since it is considered a promising future in an educational approach [11]. As games provide "instant gratification", they are agreed by a wide demographic, from children to the elderly, differentiated between women and men, so video gaming is increasingly preferred as an information medium [18]. A lot of studies have been made as of late, and most of these studies find that the game-based processes and activities in the context of education or giving information prove to affect subjects positively, increasing their motivation and

engagement in the subject while also being entertained as they take in the given information [15].

However, although there is a lot of positive feedback to a game-based approach in education, there are still cases wherein the idea of game-based learning is considered or viewed negatively, and issues such as cheating and privacy are yet to be addressed in the game-based systems. In a study conducted by Armando M. Toda et al., they identified and discussed the negative effects of game-based systems in educating people. The study found four major negative effects of game-based systems in the learning process of the selected subjects of the study, namely (1) Indifference, where the learners did not feel any difference from playing the game compared to the traditional way of learning and opting to continue with the traditional way of learning since it is more familiar to them, (2) Loss of performance, where the learners who did not understand the rules of the game and chose to not perform out of fear of being penalized for doing the tasks wrong, (3) Undesired behavior, where the game caused a different effect (may it be negative or positive) to the learners from its expected result that may be due to the lack of having none of the planning made from the development team, and finally (4) Declining effects, where the players experience a gradual loss of interest in the game as time goes on. With this information, the developers need to be mindful in designing a game-based learning system to address such problems or negative effects to avoid making the created system a failure or a useless system [48]. There are five (5) important elements in an interesting and powerful game: the Challenges, Levels, Storyline, Rewards, and Consequences. Also, ensure the developed games support the traditional problem-based learning cycle where students look at a specific situation, analyze the problem, gather resources, test their thinking, and present their findings. The game is an activity and a part of the learning process. Additionally, it allows teachers to meet various student learning styles and differentiate; it is highly engaging and often visual and can provoke both independence and teamwork; finally, it engages students in authentic problem-solving where they're empowered to make their own choices.

2.4 Related Systems

Below are the discussions of some related systems that have been developed from the past that are deemed to be helpful references in the development of this study.

2.4.1 Unlock Me: A Real-World Driven Smartphone Game to Stimulate COVID-19 Awareness

Unlock Me is a game that aims to teach and increase people's awareness about Coronavirus. The goal of the players is to catch violators of COVID-19 protocols within a certain time by creating checkpoints on the map. The game teaches topics regarding COVID-19 norms such as social distancing, use of sanitizers, wearing face masks, vaccination, cleaning, and disinfection as the game progresses through question banks, info blocks, and game interaction. The game shows positive results in teaching the topics regarding Coronavirus having significant improvements in the test scores of the subjects compared to the pretest they took before playing the game. Unlock Me used pretest and post-test questionnaires to evaluate the learning outcome of the players before and after playing the game, which will be adopted for this research study. Using pretest and post-test questionnaires will help the researchers determine the impact of using a game-based learning system on the students' learning outcome.

2.4.2 A Game-Based Learning System for Improving Student's Learning Effectiveness in System Analysis Course

This study by Ching-Hsue Cheng and Chung-Ho Su created a Game-based learning system that aims to teach topics about system analysis. The system mapped the course content from the system analysis course into the game to provide a scenario environment to teach the students. It used a role-playing genre where players are to take on roles from an IT department (project manager, system analyst, and programming staff) and complete the game by answering questions through a form of question and answer dialogues with NPCs where they are provided with multiple choices for their answers. This research study will use dialogue and question-and-answer mechanics similar to this system. The dialogue-type mechanic, where the players will interact with NPCs to answer their questions, will create an ongoing interaction and feedback between the players and the game, which will help increase the subjects' engagement with the game.

Chapter 3

Theoretical and Technical Background

The theoretical foundation of the system that was developed is presented in this chapter and consist of the guiding principles and description of the learning theories used, as well as the software and hardware technologies that were used in the systems development and implementation, as well as other relevant information.

3.1 Simulation Approach

Sometimes, questions about something cannot be investigated directly, making it hard or even impossible to teach about it in a real situation. In the context of health education, some topics cannot be taught easily because some topics are situational. For example, topics like disease prevention teach about how to prevent or minimize the occurrence of diseases through preventive measures. A way to verify the information taught in this subject is through referencing previous cases or scenarios. Still, in cases where there is no data available yet to base from, the choice is to simulate a scenario where the knowledge given to them is either followed or not and see its corresponding result, but doing so can put the learners in danger just to prove or disprove the information. To address this problem, educators can resort to gaming through Simulation Approach. The Simulation Approach translates the characteristics of the subject to be taught into elements of the game, and by letting the learners

play the created game, they will be able to translate their acquired knowledge and experience into the real world, but to ensure that this will be the case, the game model needs to be accurate. Peters et al. named three principles in the design process of a game, namely (1) *Reduction*, where elements included in the game are reduced to those which are relevant, (2) *Abstraction*, which states that the elements of the game do not necessarily need to be as detailed as in the real world to make the model less complex, and (3) *Symbolization*, where elements of the topics are molded into the elements of the game (scenarios, roles, rules, etc.).

3.2 Learning Theories in Health Education

Learning theories are the foundation of how learning is designed for learning to happen, providing information sources with models to develop instructions or ways to achieve better learning outcomes [41]. It is important to incorporate the appropriate learning approach/es into the intended system to fit the learners' needs and ensure the system's effectiveness. This study will adopt David Kolb's Experiential Learning Theory (ELT).

3.2.1 Experiential Learning Theory

Experiential Learning Theory (ELT) argues that having experience/s is the best way to retain information and facts. This type of learning puts focus on the transformation of experience/s into knowledge. ELT works in four stages, namely (1) *Concrete Learning*, where learners get a new or new interpretation of their experiences, (2) *Reflective Observation*, where learners reflect on their experiences, (3) *Abstract Conceptualization*, where learners form knowledge from their experience, and (4) *Active Experimentation*, where learners apply their newly formed knowledge. Using this learning approach will help retain knowledge for the learners as they reflect upon their learning and think of its applications in real-world scenarios [54].

In conclusion, this study made use of the Simulation Approach together with Experiential Learning Theory to provide the learners with a simulated experience to teach topics about disease prevention and control to help in increasing their health literacy and retain the information taught in their Health subject better and relate it in a real-world setting.

3.3 Characteristics of Game-Based Learning

Below are the discussion of some of the characteristics of game-based learning that the proponents found helpful in the development of the game.

3.3.1 Motivation Enhancement

A game in and of itself is a challenge. Game-based learning provides an extrinsic motivation enabled by adding game features like scores, badges, or trophies, and the competition inside the game will serve as an intrinsic motivator for the more aggressive/active students. Game mechanics must be implemented or created to make them engaging and entertaining for players. As a result of this, students will be more willing to repeat levels over and over again (practice). In particular, games can facilitate self-directed study and research; when students appreciate a certain topic in a game, they are more likely to investigate it online or in the real world, read a book about it, or watch a documentary on it [38]. This can boost the students' motivation and attentiveness [2].

3.3.2 Problem Solving

Game-based learning allows students to immerse themselves in the game, allowing them to form new connections and solve problems accurately. While Aristidis Protopsaltis noted that "the real promise of games as educational tools are in their ability to demonstrate the complexity and interconnectedness of issues" [37], it is hoped that students will rise to the occasion and become enthusiastic learners who are driven to solve the games and subsequently learn from them. Implementing a difficulty level will challenge students to think critically depending on the gravity of the problem/situation that they are in.

3.3.3 Intellectual Building

Game-based learning can provide continuous opportunities to use the proximal development zone (the area between what a learner can accomplish without help and what they can accomplish with adult supervision) [2]. Games can promote active learning in which students can engage in critical thought while still feeling the joy of learning new things.

3.3.4 Ingenuity Abilities Creating

Good learning enables a student to be a creator rather than a passive consumer of his learning. Game-based learning can provide a platform for "modernization, improvisation, adaptation, and adjustment" [38]. Games are interactive in that "when a player acts, the game reacts in a way that motivates the player to act again. As a result, the player is forced to think outside the box, think about better choices, and come up with good theories to solve the problem. According to Clark Abt in his book "serious games," traditional education views students as passive recipients. In contrast, games let them participate actively in their learning, resulting in more self-directed, imaginative, and engaged learning.

3.3.5 Challenge Anticipation

(Jose Rizal) Youth is the hope of the future. Since the students of this age will be in charge of making decisions about their future, game-based learning can allow them to practice self-discipline by pushing themselves. For instance, a game must be designed to progressively increase the difficulty level of the in-game challenges to promote the practice and the formation of positive habits. This keeps gamers interested and motivates them to improve their talents [51].

3.3.6 Graceful Failure

With game-based learning, students can make mistakes and not be severely penalized. Games give students a setting where failure has fewer repercussions, encouraging them to take chances and try new things.

3.4 System Development Tools

To create a game-based application, various tools are required to achieve the study's objectives. Analyzing the various aspects that must be considered in designing the game is necessary to better understand what must be designed and what tools will be used in the development process. In this section, the proponents listed the diagramming tools and the development tools that were used to help in the development and implementation of the game Dr. Abel's Adventures.

3.4.1 Diagramming Tools

- **Lucidchart** - Lucidchart is an online web application which allows collaboration to create charts and diagrams that are needed in the study. The proponents utilized this web application to create diagrams in order to provide a clear and unambiguous explanation to the processes in the system

3.4.2 Development Tools

- **Unity** - The proponents utilized Unity in order to accomplish different types of tasks related to the game production process. Since Unity provides a 2d and 3d platform for the developers, it is perfect for the development of the system since the game will be running on a 2d platform.

- **Unity Asset Store** - The Unity Asset Store is utilized by the proponents in the game development process since there are a lot of free assets available. Unity Asset Store also has various types of Assets in the store ranging from textures, animations and models to entire project examples, tutorials and editor extensions that can be useful in the project development.

- **Microsoft Visual Studio** - The proponents used Microsoft Visual Studio as an IDE in game development since it brings a premium debugging experience to the Unity game engine. Visual Studio can also identify issues quickly by debugging your Unity games in Visual Studio—set breakpoints and evaluate variables and complex expressions which are advantageous for the proponent in order to save time and make the development period productive.

- **Third Party Asset Websites** - The proponents utilized third party websites like opengameart.org, itch.io, and kenney.io to download game assets such as characters, and music and sound assets that is used to make the game more fun and lively.

- **LibreSprite** - LibreSprite is a free and open source program for creating and animating sprites. The proponents utilized this application to create game assets (Characters, Objects, Tilemaps) that are used in the game development process.

Chapter 4

Methodology

The definitions of technical terms used in this study are presented in this chapter. This section also includes a brief discussion of the procedures for developing and implementing the system, and other important factors for its development, as well as the procedures for conducting the experiment and testing.

4.1 Definition of Terms

- **Role Playing Game** - is a type of game where players assume the roles of a character/s in order and follow a storyline to complete the game.
- **Simulation game** - is a type of game which copies activities from real life to the game. It is mainly used for purposes such as training, analysis, prediction, and entertainment.
- **Non Player Characters or NPCs** - are characters that are programmed to do specific functions in the game instead of being controlled by the players. It is mainly used as a plot device to advance the storyline of the game, or guides to assist the players complete the game.
- **Trust System** - a game mechanic of Dr. Abel's Adventures that is represented by a heart where the players need to maintain in order to complete the game.

4.2 Development Model

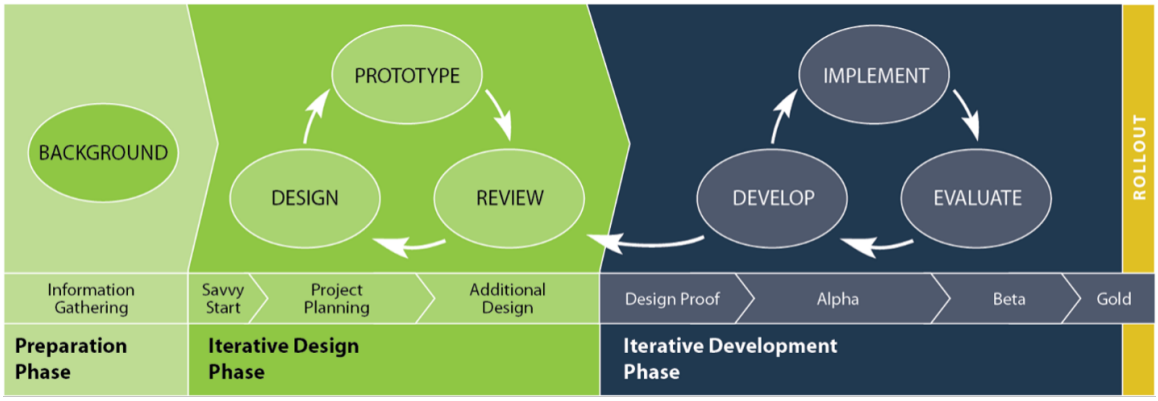


Figure 4.1: SAM Model

This study developed the system using the Successive Approximation Model (SAM) or SAM Method. SAM is an agile, iterative development that will allow the developers to assess the system and find areas that need improvement and be able to fix them quickly. This method will help improve the system to create the desired system quickly and efficiently.

SAM is composed of three parts, namely (1) the Preparation Phase, where the necessary data and context for the project are gathered, (2) the Iterative Design Phase, where the prototype is designed so it can be evaluated; and lastly (3) Iterative Development, where the prototype is continuously developed until the desired outcome is achieved.

4.3 UML Class Diagram

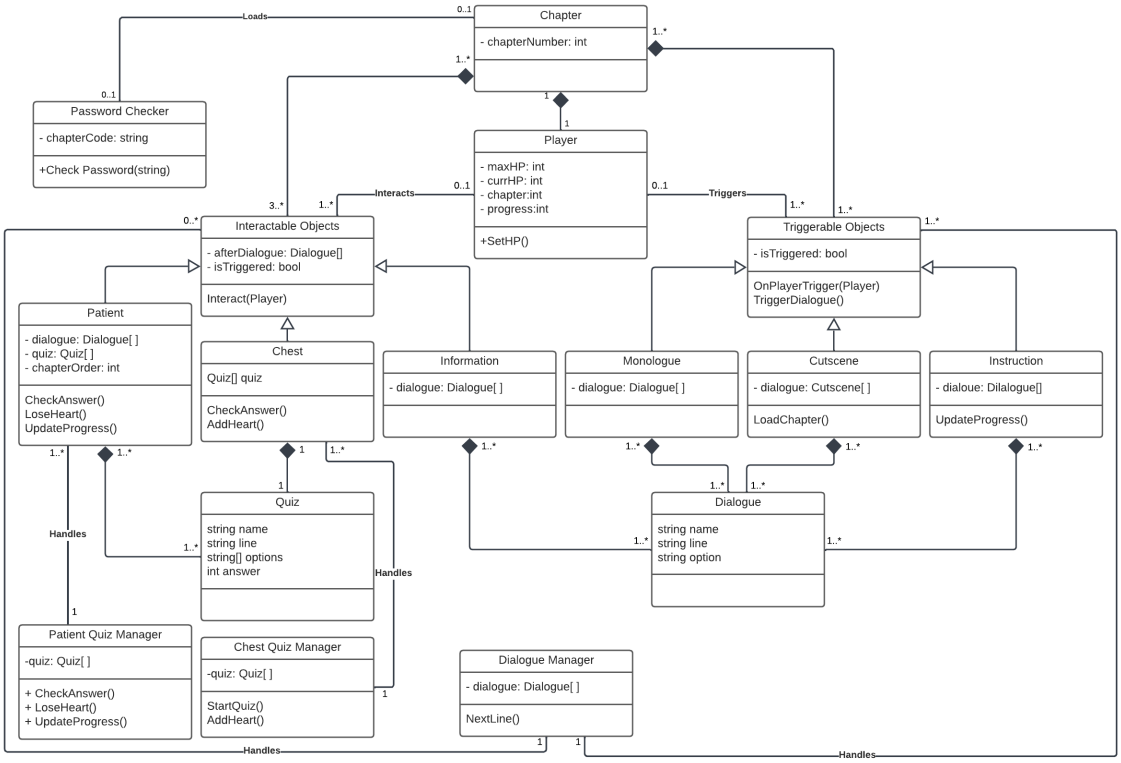


Figure 4.2: UML Class Diagram

The game is divided into different CHAPTERS assigned with specific pathogens, which are loaded through chapterCode from the PASSWORD CHECKER Class. CHAPTERS consists of a PLAYER, which will be the character that the players will control; one or more TRIGGERABLE OBJECTS which are objects within the game that are interactable by stepping onto; and three or more INTERACTABLE OBJECTS, which are objects that are interactable by the use of the interact button in the game.

TRIGGERABLE OBJECTS has three types, namely INSTRUCTION, which gives the players instruction to help in the progression of the game, MONOLOGUE which provides the player with recaps from previous chapters played; and CUTSCENE, which triggers the cutscene of the game and loads the chapter it is assigned to and is handled by a DIALOGUE MANAGER which runs the

dialogue of the game.

INTERACTABLE OBJECTS has three types, namely PATIENT, which are NPCs that asks players questions in the game, CHEST which is an object that contains questions and gives players a chance to regain lost hearts; and INFORMATION which are NPCs that provides players with information regarding the chapter, and is handled by a DIALOGUE MANAGER. PATIENT and CHEST consists of one or more QUIZ, the class used to create quizzes in the game, and are handled by their respective QUIZ MANAGER, which checks the players' answers and performs their functions.

4.4 Use Case Diagram

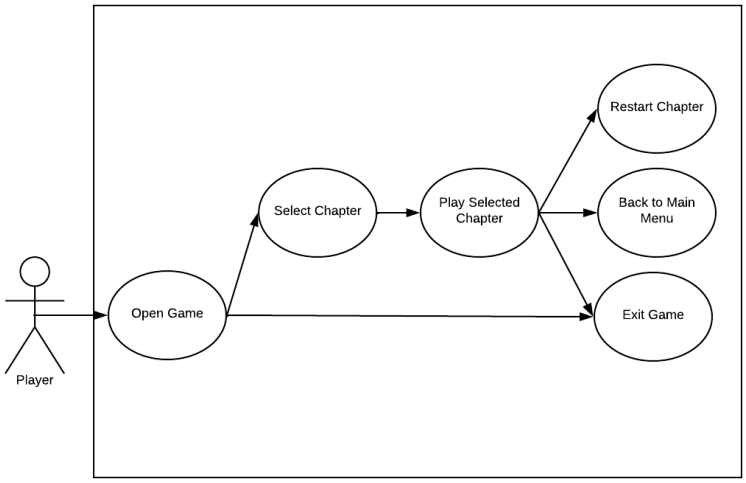


Figure 4.3: Use Case Diagram

Upon opening the game, the players will be presented on the home screen with a chapter select button; If they Select a chapter, the chapter will load, and they will have the options to: (1) Restart the chapter, (2) Go back to Home Screen, or (3) Exit the game.

4.5 Game Flow Diagram

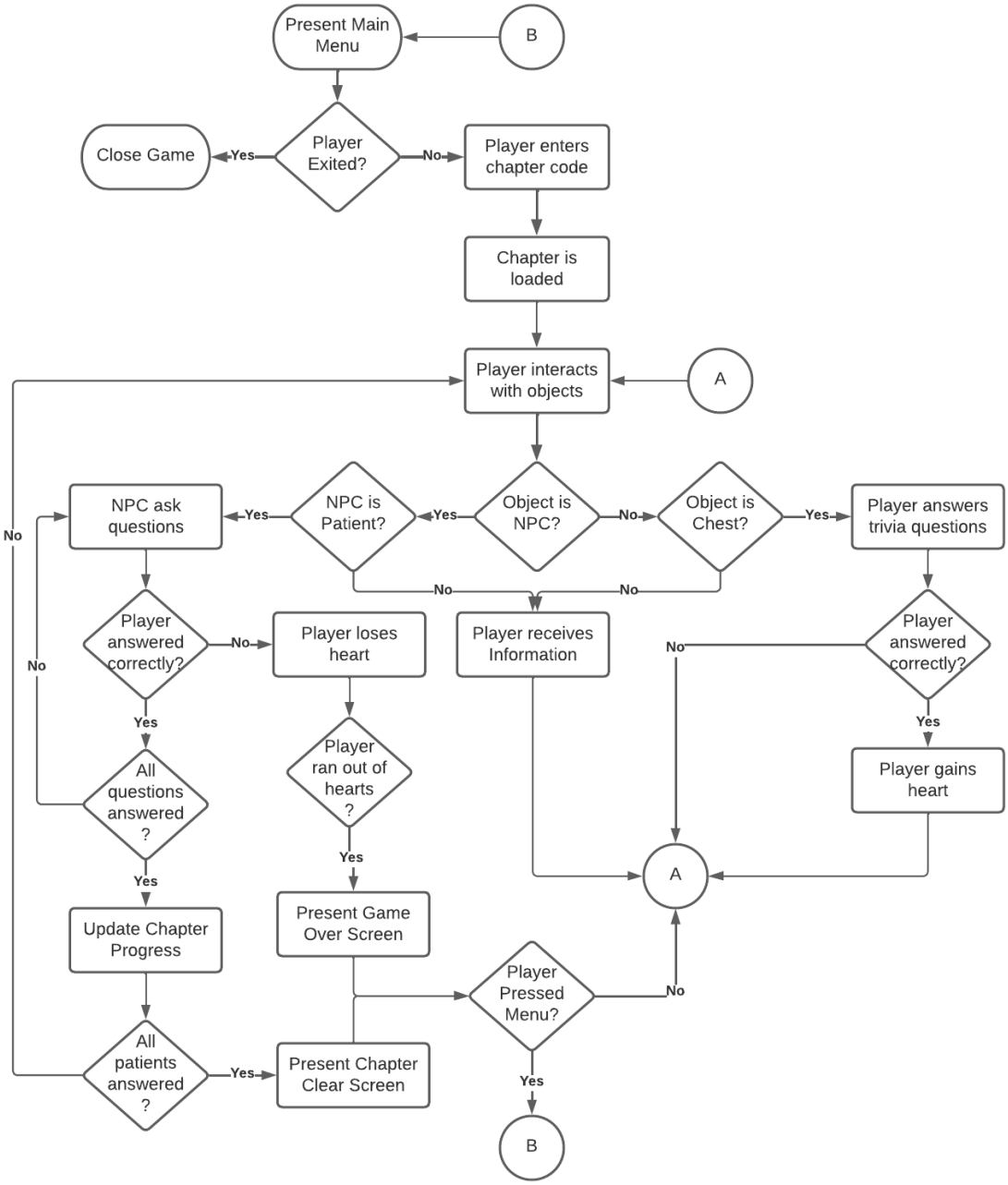


Figure 4.4: Game Flow Diagram

Upon opening the game, the players will be presented with the main menu, where they will have the option to close the game, which will terminate the game, or select a chapter to play by entering the chapter code given to them, which will load the chapter that corresponds to the code entered. While playing, the players will interact with different objects within the game. If the player interacts with a Patient NPC, they will be engaged in a Question and Answer dialogue with it, to answer all of its questions about its health problem correctly; if the players can answer all of the Patient NPCs questions correctly their Progress will be updated so they can interact with the remaining Patients in the map and if they can correctly answer all of the Patient NPCs questions the current chapter will be cleared, However in the case of the players answering incorrectly, they will lose a heart, and if they ran out of hearts the game would be over where they can choose to go back to the main menu or retry the chapter. In another case, if the player interacts with an Information NPC, they will also be engaged in a dialogue with it. They will receive information about the current chapter, which includes directions, instructions, fun facts, and knowledge about the chapter, and Lastly, if the player interacted with a chest, they will be engaged in a quick Question and Answer dialogue where the chest will present them with a single question related to the current chapter, if the player answered it correctly they would be able to regain a heart they lost during their interaction with the Patient NPCs.

4.6 Systems Design

The game Dr. Abel's Adventures was developed to assess the effectiveness of game-based learning in enhancing the learning outcomes of Grade 8 students of Camarines Sur National High School about prevention and control of communicable diseases.

4.6.1 Integration of Lecture

For the players to learn, the proponents followed the book that was provided by Camarines Sur National High School (CSNHS) entitled Physical Education and Health 8, which is in line with the curriculum guide presented by the Department of Education (DepEd) MELC: K to 12 Most Essential Learning Competencies (2022-2023) in the Health component of the MAPEH Subject of Grade 8 Students of Camarines Sur National High School. This will be done in a quiz-type environment where the players must successfully choose the correct answer from the given choices.

Topics that are included in the game are; what is communicable diseases, the different kinds of germs and disease agents, e.g., bacteria, viruses, fungi, parasites, etc., the transmission of each type of communicable disease agent, the prevention needed to combat the different kinds of communicable disease agents, factors or causes of communicable diseases, signs and symptoms of common communicable diseases, effects of common communicable diseases and the misconceptions, myths, and beliefs about common communicable diseases

3rd Quarter	demonstrates understanding of principles in the prevention and control of communicable diseases for the attainment of individual wellness	consistently demonstrates personal responsibility and healthful practices in the prevention and control of communicable diseases	discusses the stages of infection	Week 1	**H8DD-IIIa-15**
			analyzes the leading causes of morbidity and mortality in the Philippines		**H8DD-IIIa-16**
			*discusses the most common communicable diseases • signs and symptoms of common communicable diseases • effects of common communicable diseases • misconceptions, myths, and beliefs about common communicable diseases • prevention and control of common communicable diseases	Week 2 to Week 3	**H8DD-IIIb-c-17** **H8DD-IIIb-c-18** **H8DD-IIIb-c-19**
			analyzes the nature of emerging and re-emerging diseases	Week 4 to Week 5	**H8DD-IIId-e-20**
			demonstrates self-monitoring skills to prevent communicable diseases	Week 6 to Week 8	**H8DD-IIIf-h-21**

Table 4.1: Health Subject's Most Essential Learning Competencies

4.6.2 Game Title Inspiration

The title Dr. Abel's Adventures was inspired by a Filipino figure named Dr. Abelardo Aguilar, who contributed something to medicine. Dr. Aguilar was a Physician from Iloilo who discovered that *Streptomyces Erythreus* bacteria had antibiotic-producing properties when he was

working on the soil samples found in his backyard back in 1949. Dr. Aguilar submitted his works to Eli Lilly and Company's research department, where he worked as a researcher, which later announced the discovery of a brand-new antibiotic in 1952, which is generically named *Erythromycin* [8]. When Erythromycin was registered, they used the patent name *"Ilosone"* and *"Ilotycin"* which is derived from the place where it was first discovered (Iloilo City). Today, Erythromycin is a widely used antibiotic to treat and prevent various infections in the body. These infections can affect the skin and respiratory system. Patients allergic to other antibiotics, such as penicillin, are also given Erythromycin as a replacement medication [42]. Although Dr. Abelardo Aguilar was not directly credited with discovering Erythromycin, his notable medical work is enough to make him well known.

The proponents used this Filipino figure as an inspiration not just because of his notable contribution to the field of medicine but also because his work was not properly credited. It is important, especially for young learners, to remember and value the great contributions our past generations have made. Furthermore, the proponents also utilized this kind of strategy to create an impact and bring significant and relevant information to the table.

4.6.3 Game Storyline

After studying medicine for four years at Ateneo de Naga University, Abel decided to complete his residency at the local hospital. He spent three years training at the same medical center and became a licensed doctor. However, before he could even begin practicing, Abel received devastating news. His beloved grandfather had passed away due to lung complications caused by a virus and poor lifestyle choices. Heartbroken, Abel decided to put his career on hold and tend to his grandfather's house, which had been left to him. When he arrived back in his hometown, he was greeted by the province's mayor, who expressed his condolences and shared his concerns about the state of the town's healthcare system. The mayor explained that many people in the area were struggling with health issues due to the lack of proper services and medical personnel. Curious about Abel's background, the mayor asked about his past jobs and hobbies. Learning that Abel was a recent medical graduate, the mayor saw an opportunity to improve the town's healthcare services. He offered Abel a plot of land to build his clinic and asked for help solving the town's health problems. Although initially hesitant, Abel accepted the mayor's offer and vowed to do his best to provide the town with the healthcare it deserved.

4.6.4 Discussion of Sprites Used

The game Dr. Abel's Adventure utilized the assets on the website itch.io, specifically the Mystic Woods asset pack created by Game Endeavor, which the proponents bought for 168 pesos. The proponents modify the sprites that are present in the asset pack. This is allowed under the license of the asset pack, which permits modification and use in commercial projects.

- **Doctor Abel** is the main character in the game "Dr. Abel's Adventure." He graduated from the Ateneo de Naga University with a degree in Medicine, and after that, he completed a four-year residency in a local hospital before becoming a licensed doctor. As a doctor, Abel always practices proper precautions when treating patients, so he is always seen wearing his PPE (Personal Protective Equipment), such as a mask and a white doctor's coat.
- **Mr. Mayor** is a supporting character in the game "Dr. Abel's Adventure," who holds a deep connection with the family of Doctor Abel. He is a helpful and supportive character, who helps the player kick start their adventure. An old man with white hair who wears a brown coat and carries a red cane.

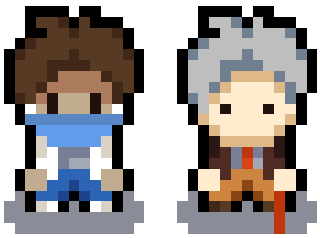


Figure 4.5: Doctor Abel and Mr. Mayor

Chapter 1 Characters: BACTERIA

- **Architect JC** is a talented architect who enjoys listening to music. He has green hair and is always seen wearing his red headphones. JC is the first NPC (non-playable character) the player needs to encounter to start the story. During the interaction, the player may notice that JC has a chronic cough that becomes *Tuberculosis*. To complete the story, the player must identify the disease, its cure, prevention, and common myths and misconceptions surrounding the disease.

- **Engineer John** is the second NPC the player needs to interact with in Chapter 1. He is a friend of Architect JC and a talented engineer. He wears a yellow construction hat and is experiencing stomach pain, which is later identified as *Typhoid Fever*. Engineer John is the second NPC the player must interact with to progress through the chapter.

- **Foreman Carlo** is the last NPC the player must interact with to finish Chapter 1. He is the trusted foreman of Engineer John, and they have worked together for five years. Carlo is a hard-headed person who values wealth over his well-being. He can be seen wearing a shade and a red scarf. During the interaction, Carlo is often distracted by his urge to urinate, which turns out to be a *Urinary Tract Infection*.

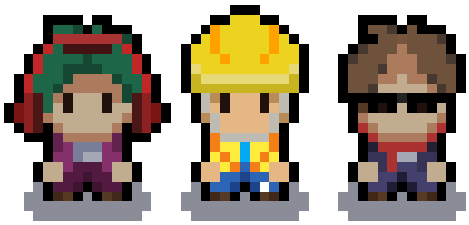


Figure 4.6: JC, John and Carlo

Chapter 2 Characters: VIRUS

- **Electrician Kaloy** is a skilled electrician who enjoys spending most of his free time playing computer games at a nearby computer shop. He has a dark complexion and always wears safety glasses. In Chapter 2 of the game, Kaloy is the first NPC the player must interact with to progress through the game. During the player's interaction with Kaloy, they may notice that he has a chronic cough and frequently sniffs and sneezes, which turns out to be *Influenza*.

- **Plumber Juan** is the second NPC the player interacts with in Chapter 2, and he is a plumber who has a hearing problem. He can be seen wearing a hat with the letter P on it. During the game, the player may notice that, unlike the other NPCs, Juan is not experiencing any major complications besides a bump on his hand, which turns out to be a *Wart*.

- **Cleaner Sef** is the last NPC the player must interact with in Chapter 2. He is a cleaner who always wears his safety goggles and brings his broom. Sef's character is a pessimistic

person who always question government efforts. During the interaction, the player may notice that Sef has a dry cough and complains about being unable to taste or smell anything, later identified as *COVID-19.*

Figure 4.7: Kaloy, Juan and Sef

Chapter 3 Characters: FUNGI

- **Gardener Fred** is the first NPC the player must interact with in Chapter 3 of "Abel's Adventure." He is a diligent gardener and an old man who is also a friend of Mr. Mayor and a loving father of Painter Daisy. He wears a straw hat and a green jumper. During the interaction, the player may notice that Fred constantly complains about his itchy foot, which is later identified as *Athlete's Foot.*

- **Painter Daisy** is the second NPC the player must interact with to progress through Chapter 3. She is the lovely daughter of Gardener Fred and the town's most brilliant painter. She has orange hair with flower decorations and wears a pink dress. During the interaction, the player may notice that she constantly scratches near her groin area, which is later identified as *Jock Itch* caused by sharing clothes with her father.

- **Designer Jessie** is the last NPC the player encounters in Chapter 3. She is a friend of Painter Daisy and a pet lover. She has yellow hair and wears a pink suspender. During the interaction, Jessie can be noticed scratching her head a lot, which is later identified as *Ringworm* that was transferred to her by her pets.

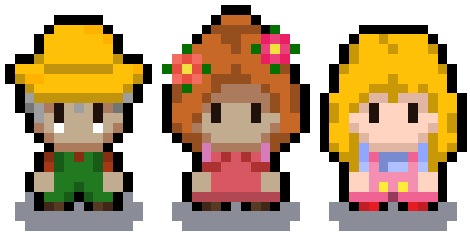


Figure 4.8: Fred, Daisy and Jessie

Chapter 4 Characters: PARASITE

- **Guard Magnus** is the first NPC the player needs to encounter in the last chapter. He is a former bodyguard of Mr. Mayor who quit his job to take care of his kids when his wife passed away. He can be seen wearing a guard uniform. During the interaction, the player may notice that Magnus complains about his stomach ache, later identified as *Trichinosis* caused by eating raw pork during his son's birthday celebration.

- **Clerk Mariane** is the second NPC the player needs to interact with in the last chapter. She is a 'friend' of Guard Magnus who helped him raise his children when his wife passed away. Clerk Mariane has purple hair and wears an orange headband, a pink dress, and a washed-out blue coat. During the interaction, the player may notice that Mariane complains about feeling ill, later identified as *Malaria* caused by the dirty pond near her house that was a breeding ground for mosquitoes.

- **Nurse Joy** is the last NPC the player needs to interact with in the last chapter. She is a graduate of medicine and a licensed nurse who decided to leave her town to pursue her passion in medicine abroad for practical reasons. Upon learning that a healthcare facility was being built in her town, she decided to take the chance and leave her work abroad to return to her hometown. Nurse Joy wears a nurse's outfit. During the interaction, the player might notice that she always scratches her head; this is later identified as *Pediculosis.*

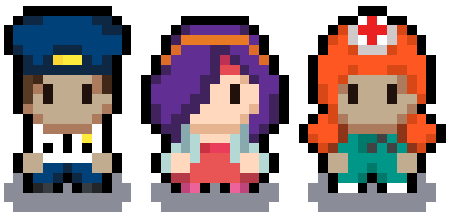


Figure 4.9: Magnus, Mariane and Joy

4.6.5 Gameplay and Mechanics

The game is inspired by Pokemon and Stardew Valley, where players wander off to different places and interact with different people/NPCs.

In "Dr. Abel's Adventures," Dr. Abel, who recently graduated from Ateneo de Naga University and became a doctor, builds a clinic in his hometown to provide proper healthcare to the people after learning that health systems and health services are a problem in his town. To complete the clinic, he needs help from various NPCs to gather resources and services. However, for the NPCs to help him, the player needs to help the NPCs first by finding out the proper diagnosis for their health problems.

The game contained multiple chapters containing different scenarios categorized by disease agents (pathogens) of communicable diseases; in each scenario, it also included the characteristics of the pathogen, its types, mode of transmission, factors or causes of it, myths and misconceptions surrounding the disease as well as its prevention measures and/or cure. The game consists of four chapters in total, Chapter 1 contains scenarios discussing topics solely about bacteria, Chapter 2 for viruses, Chapter 3 for fungi, and Chapter 4 for parasites. This was done to help moderate the amount of information given to the students to avoid information overload and help them recall the topics from the scenarios better.

Topics and concepts incorporated in the game were administered using a quiz-type environment through interactions with objects in the game. In the game, two types of NPCs are present, one that asks for help or Patient NPCs and the other which gives information to the player or Information NPCs. The Information NPCs will provide the players with necessary information related to the chapter the players are in (instructions, fun facts, and necessary information), while the Patient NPCs will state the details as to how they arrived at their health problem, the player then are

presented with options in the interface where they will need to correctly identify the correct answers to some of the Patient NPCs questions, such as what causes their health problem, how to cure or prevent their health problem, etc. After the player successfully identifies the correct answer to all of the Patient NPCs' questions in every chapter, the Patient NPCs will help them back by lending the player some services or resources in his objective to build a clinic.

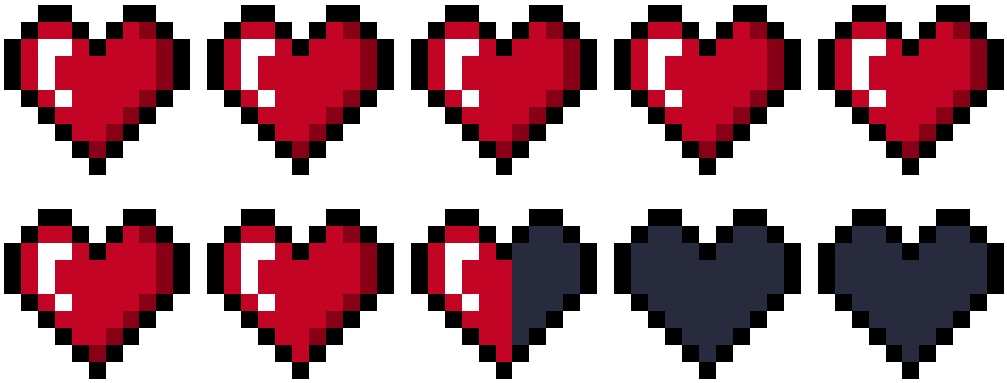


Figure 4.10: Patient NPCs Trust System

If, for example, the player didn't get all the right choices, the game implemented a *trust system* where the player will be presented with five hearts to begin to represent the level of trust of all the Patients NPCs to the player in each chapter, every wrong choice the player picks a heart will lose, if the player loses all five hearts the game will end and the player loses. Additionally, some Chests are scattered throughout the map, which the players can interact with within the game. These *chests* contain trivia questions that the players can answer, and upon giving a correct answer, gives the player an additional heart (trust) that they have lost from their previous interactions with Patient NPCs. For the players to win and clear the game, the player needs to build the clinic by gathering help and resources from every Patient NPCs by correctly identifying/answering the Patient NPCs questions in each chapter, mainly as to how he/she leads to his/her health problem.

4.6.6 Game Storyboard

Figure 4.11: **Scene 1:** Shows Abel arriving into town to tend to his grandfather's house.

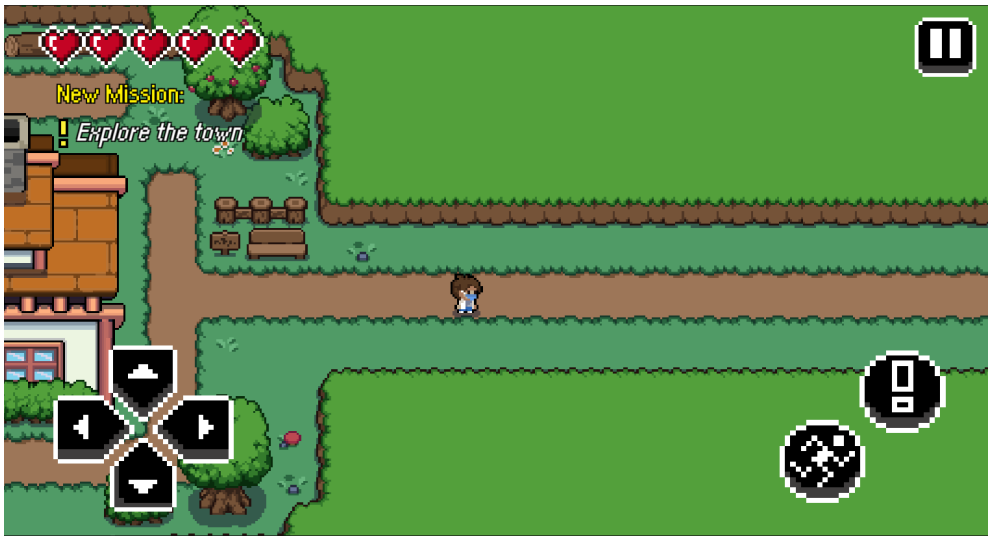


Figure 4.12: **Scene 2:** Shows Abel going into town to explore what has changed.

Figure 4.13: **Scene 3:** Shows Abel and Mr. Mayor's interaction.

Figure 4.14: **Scene 4:** Shows Mr. Mayor offering Abel a plot of land.

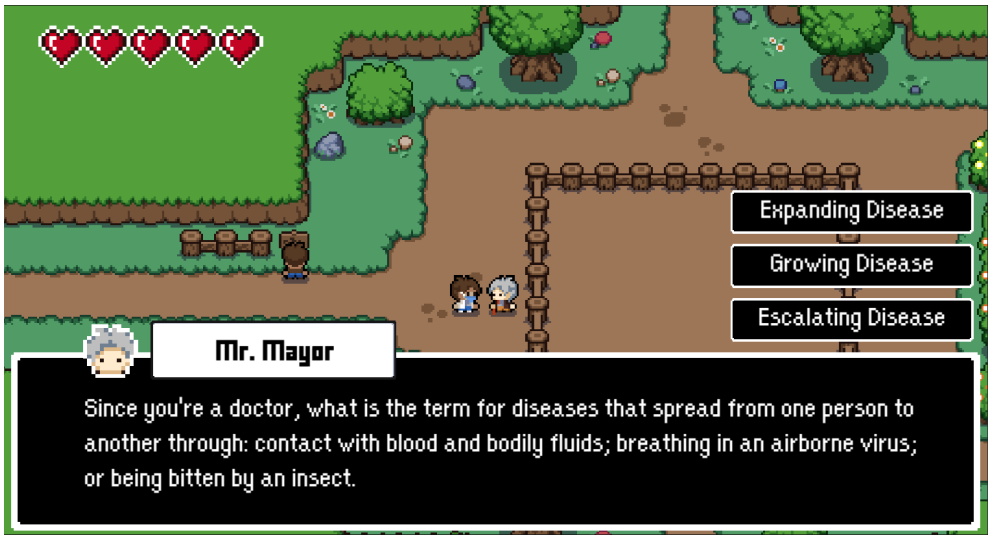


Figure 4.15: **Scene 5:** Shows Mr. Mayor demonstrating the trust system.

Figure 4.16: **Scene 6:** Shows a loss of heart when a player answers incorrectly.

Figure 4.17: **Scene 7:** Shows Mr. Mayor explaining how to gain hearts.

Figure 4.18: **Scene 8:** Shows Mr. Mayor instructing Abel what to do first.

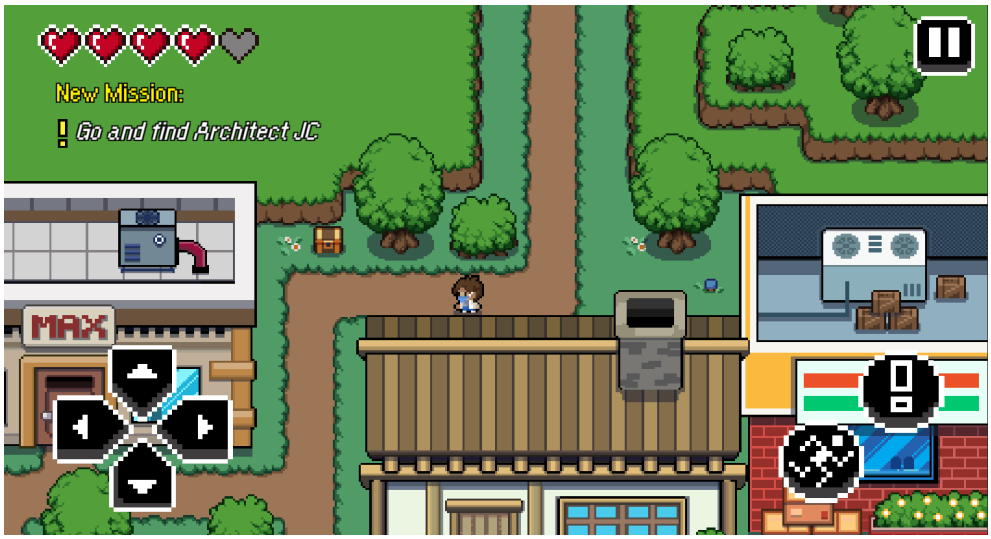


Figure 4.19: **Scene 9:** Shows the location of the chest.

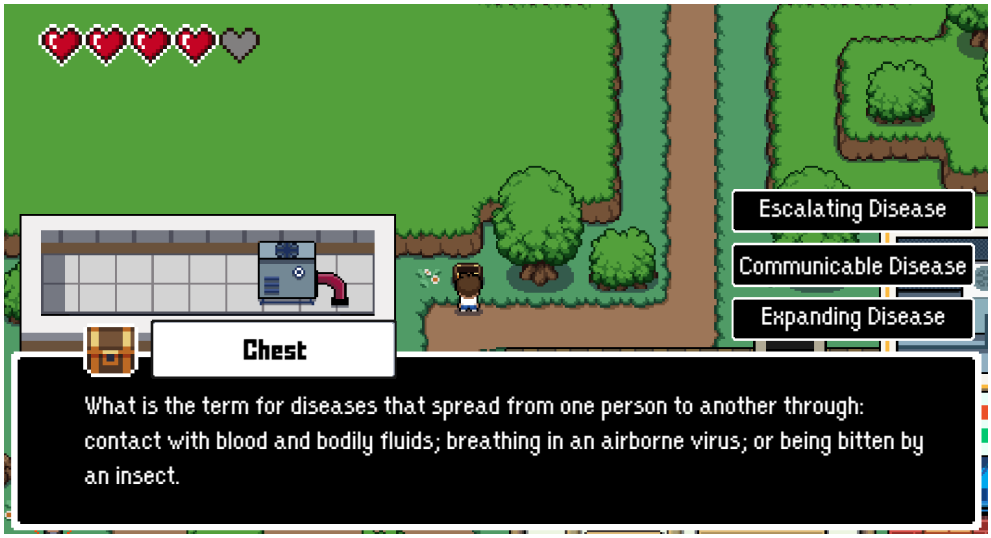


Figure 4.20: **Scene 10:** Shows Abel interacting with the chest.

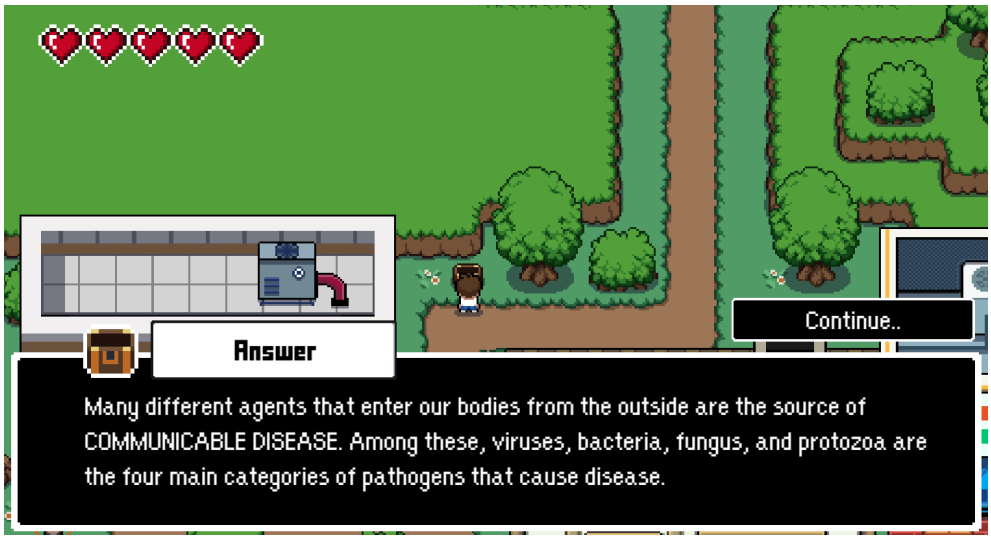


Figure 4.21: **Scene 11:** Shows the explanation when answering the chest correctly

Figure 4.22: **Scene 12:** Shows the location of Architect JC as per Mr. Mayors instruction.

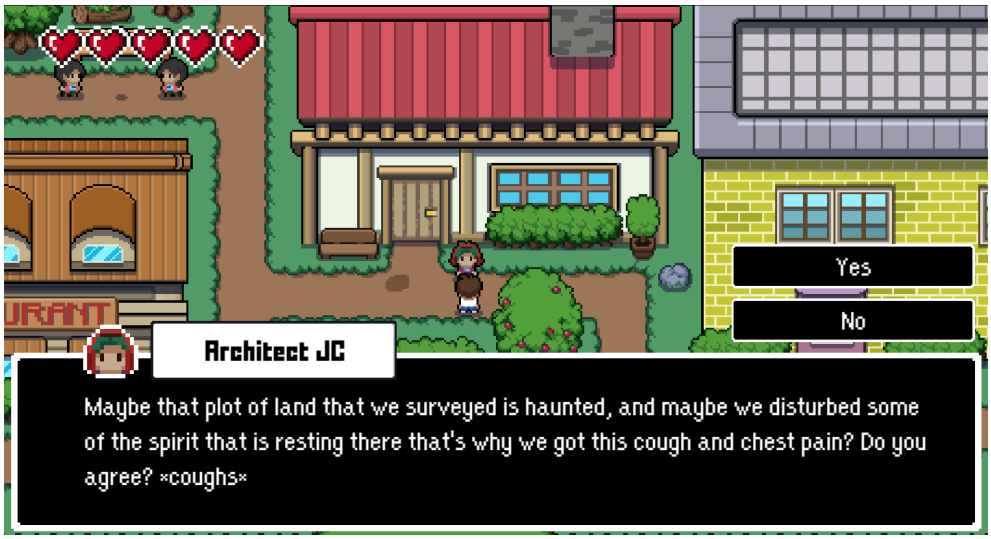


Figure 4.23: **Scene 13:** Shows the question that Abel must need to answer correctly.

Figure 4.24: **Scene 14:** Shows Architect JC giving instructions on who to visit next.

Figure 4.25: **Scene 15:** Shows the continuation of Architect JC giving instructions where to go next.

Figure 4.26: **Scene 16:** Shows the question that Abel must need to answer correctly.

Figure 4.27: **Scene 17:** Shows Engineer John giving instructions on who to visit next.

Figure 4.28: **Scene 18:** Continuation of Engineer John giving instructions where to go next.

Figure 4.29: **Scene 19:** Shows the question that Abel must answer to clear the chapter.

Figure 4.30: **Scene 20:** Shows the chapter clear screen after clearing all the 3 NPC's.

Figure 4.31: **Scene 21:** Shows Abel's monologue at the start of Chapter 2.

Figure 4.32: **Scene 22:** Shows Abel going into town to check the progress of his clinic.

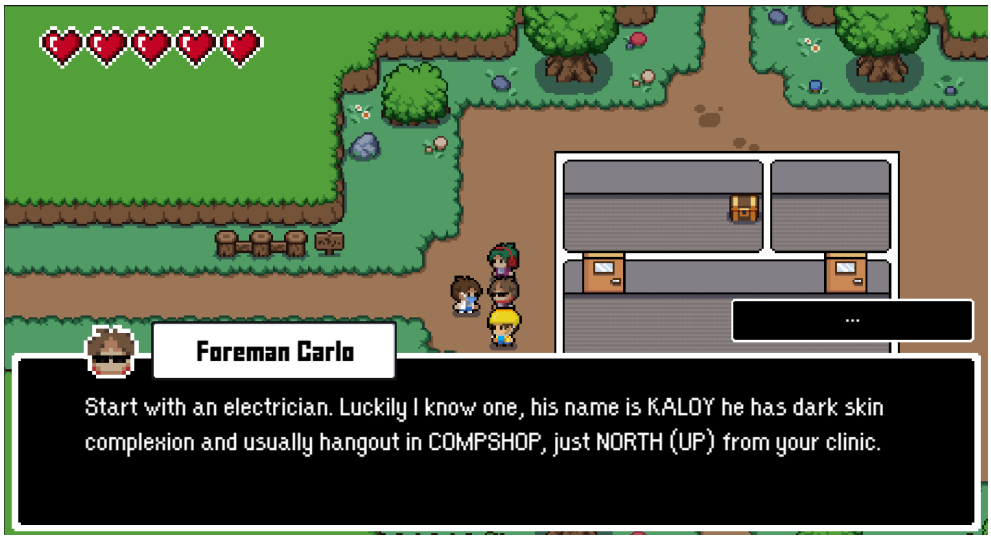


Figure 4.33: **Scene 23:** Shows Foreman Carlo giving instructions where and who to go next. After Abel successfully diagnoses the three (3) NPC's present in Chapter 2, the player will be presented with a 'Chapter Cleared' screen.

Figure 4.34: **Scene 24:** Shows Abel's monologue at the start of Chapter 3.

Figure 4.35: **Scene 25:** Shows Abel asking Mr. Mayor if he knew someone that can provide services.

Figure 4.36: **Scene 26:** Shows Mr. Mayor suggesting his gardener friend. After Abel successfully helps and diagnoses all of the Patient NPCs present in the Chapter 3, the player will be presented with a 'Chapter Cleared' screen.

Figure 4.37: **Scene 27:** Shows Abel's monologue at the start of Chapter 4.

Figure 4.38: **Scene 28:** Shows the continuation of the monologue for Chapter 4.

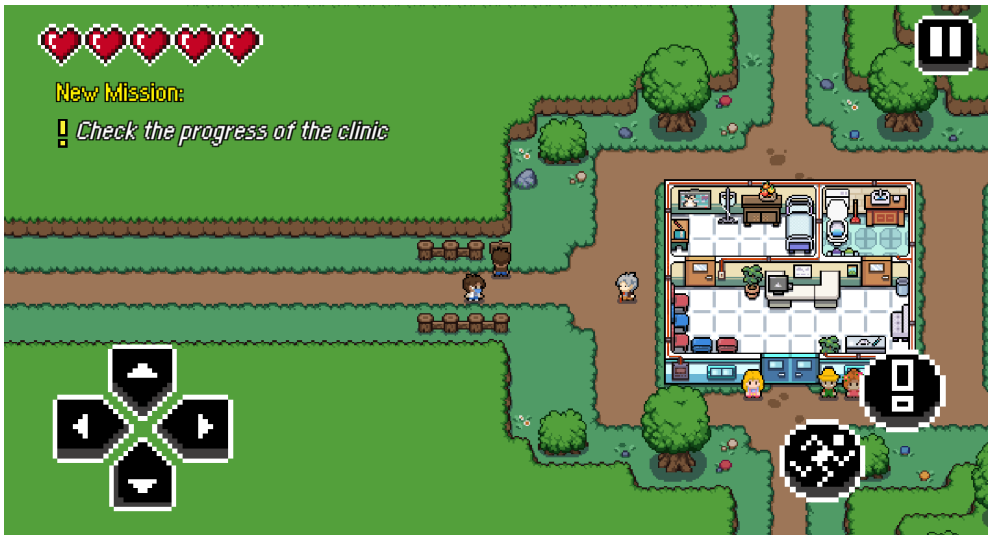


Figure 4.39: **Scene 29:** Shows Abel going into town to check the progress of his clinic.

Figure 4.40: **Scene 30:** Shows Abel asking Mr. Mayor if he knew someone he can hire as a staff

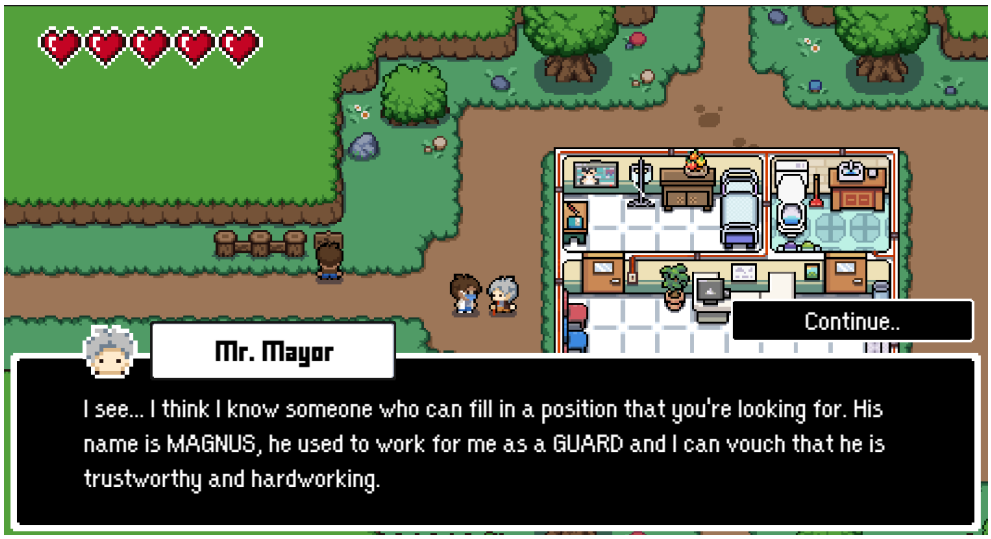


Figure 4.41: **Scene 31:** Shows Mr. Mayor suggesting Guard Magnus as the first staff for his clinic. After Abel successfully helped all the three (3) NPCs in the chapter they will be presented with a concluding cut scene and a 'Chapter Clear' screen.

Figure 4.42: **Scene 32:** Shows the fully constructed clinic, after Abel helped all the NPCs in town.

4.6.7 Game Controls

The game was designed to be played from a landscape perspective, and the control system is based on mobile touch input. The left side of the screen is intended for the character's movement (left, right, up, and down), which comes in the form of a dynamic touch screen joystick. At the same time, players can also use the right side of the screen to interact with the NPC by pressing the button with the "Exclamation Icon" and run by pressing the button with the "Run Icon ."The upper-right button is meant for pausing, restarting, saving the player's progress, or exiting the game.

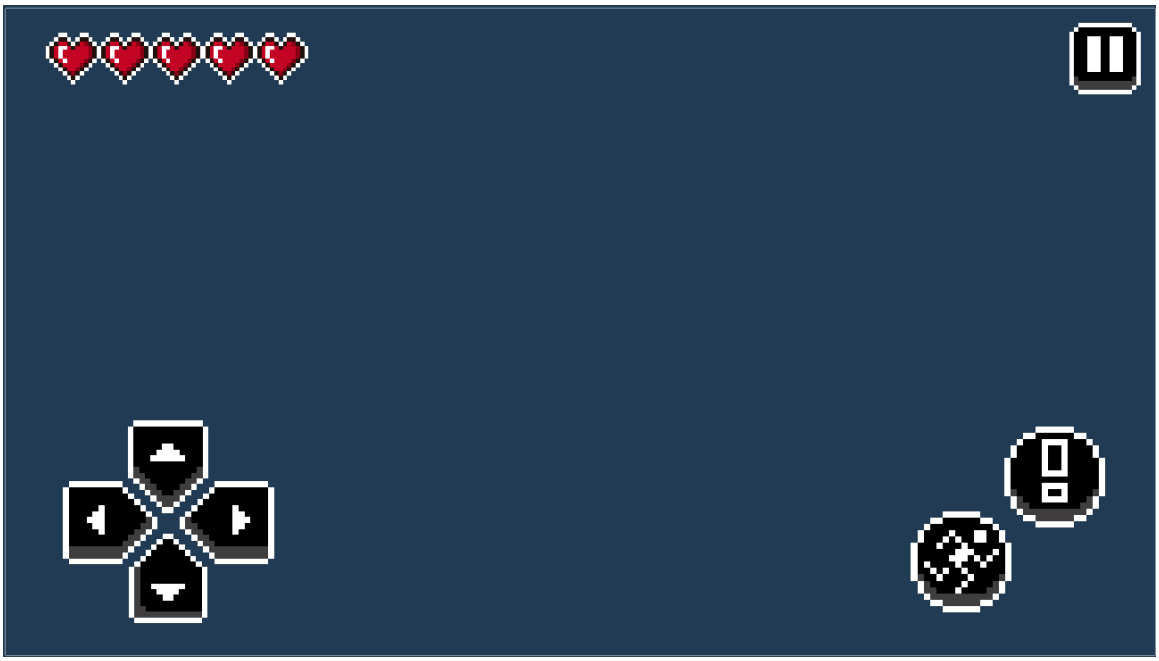


Figure 4.43: Game Controls

4.6.8 Art Style and Game Type

The game was designed to be a 2D style mobile game since the audience is younger, and also, there are varieties of 2D sprites and assets available on the internet. The genre of the game was decided to be a role-playing simulation game which will make the players act the role of a doctor helping NPCs figure out their health problems and give them appropriate remedies to earn resources and help from NPCs in finishing the clinic. Through this design, it will help the learners to learn the topic

better, and experience the learning in a simulated environment, making them feel more comfortable experiencing things without worrying much about making mistakes.

4.6.9 Music

Regarding music, the proponents used websites like Open Game Art and Kenney. Open Game Art and Kenney are websites offering free and open-source video game assets where the proponents will get the music and sound effects assets that will be used in the game's development.

4.6.10 Language

The dialogues and other elements in the game were delivered using English as the language. The reason for this is that English—both as a language and as a subject—is taught beginning in Grade 1 and is further developed in Grade 6. The Department of Education (DepEd) MELC: K to 12 Most Essential Learning Competencies (2022-2023) in English subject states that by the time a student completes the Elementary phase, he/she should be able to communicate feelings and ideas orally and in writing with a high level of proficiency in English, listen critically in English, and read various English text types materials to serve learning needs in meeting a wide range of life's needs [35]

4.7 Testing Approach

The participating school, Camarines Sur National High School, provided this study with two sections from grade 8 level, 8-Mercury, and 8-Mars, who are taking the MAPEH subject, which covers the topic of communicable diseases.

4.7.1 Preliminary Survey

The researchers surveyed to gather information on the population of the sections that will participate in the study. The total number of participating students will be 106, having 54 students (50.9%) from 8-Mercury, and 52 students (49.1%) from 8-Mars.

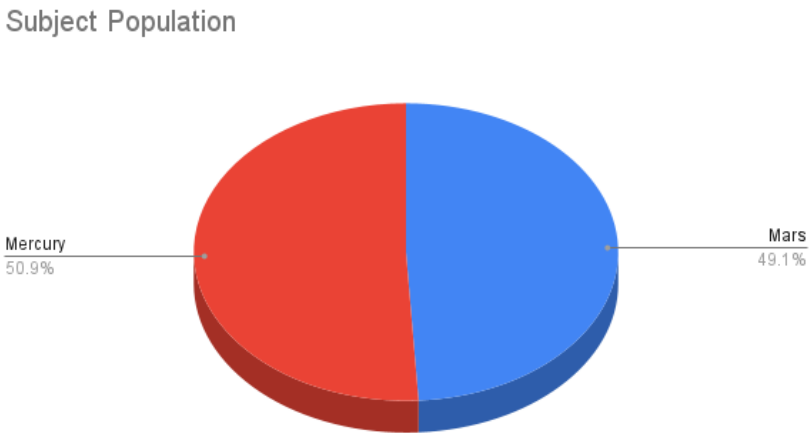


Figure 4.44: Total Population of subjects

The survey also showed that 98.1% of the respondents own a phone, while the remaining 1.9% do not.

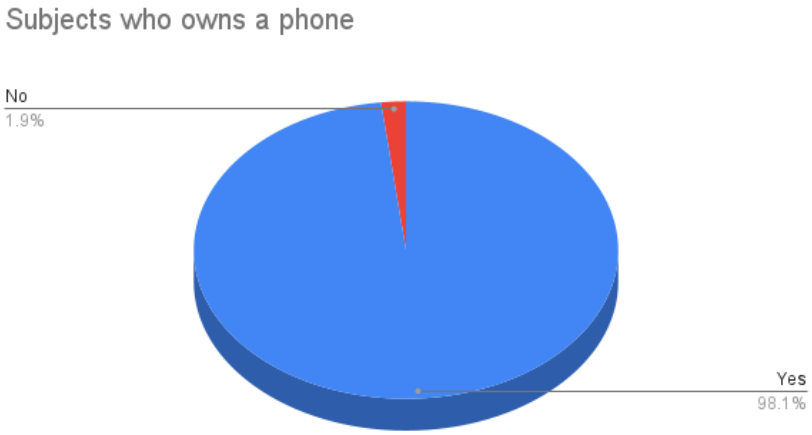


Figure 4.45: Subjects Who Own A Phone

The result shows that all (100%) of the respondents from 8-Mars own a phone, while for 8-

Mercury only 52 (96.2%) own a phone, while the remaining 2 (3.8%) do not.

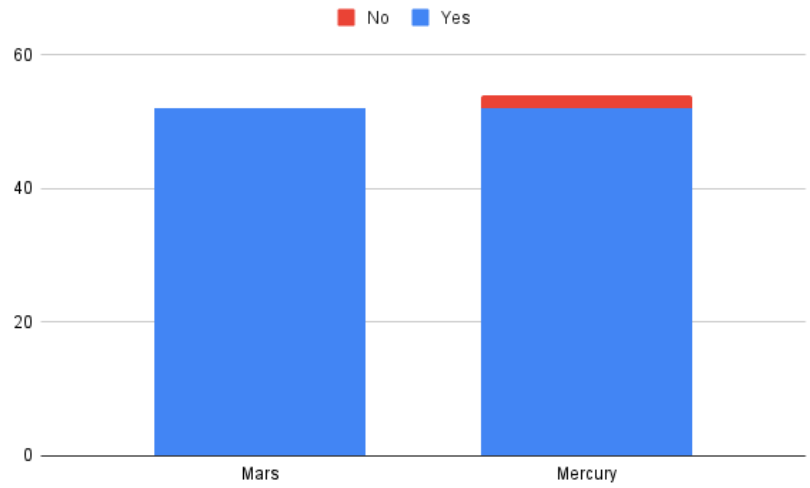


Figure 4.46: Subjects Who Own A Phone by Section

From the results of the survey, the majority (97.2%) runs on Android, and the remaining (2.8%) is shared between iOS (0.9%) and "-" or none (1.9%) for those who do not own a phone, and all (100%) of the android users (97.2%) have their android phones running on android seven or higher.

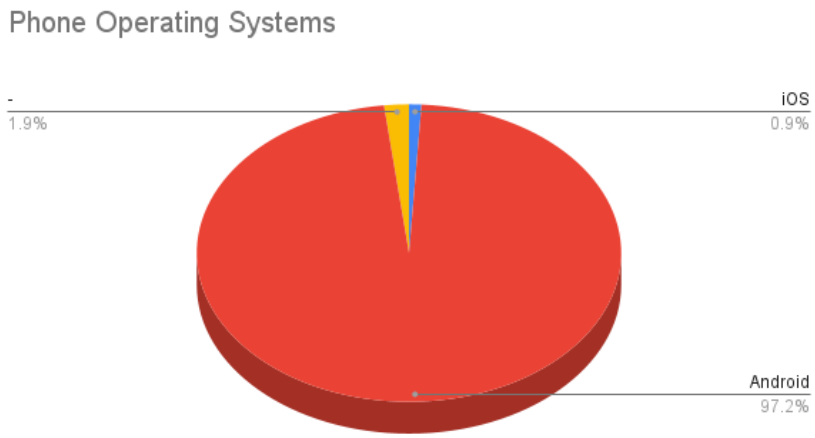


Figure 4.47: Phone Operating Systems

With the data gathered from the survey, the researchers decided to create an Android based mobile game, as most of the respondents use the Android phone. Also, the study will take 8-Mars as the section to test the game, as all of the students in the section own a phone, most of which runs on Android.

4.7.2 Research Design

This research study used the Non-Equivalent Group type of Quasi-Experimental design. This type of design chooses groups that appear to be similar, but only one of the chosen groups will be testing the study [46]. In the case of this study, 8-Mars was the experimental group that used the game to learn the concepts regarding communicable diseases. At the same time, the 8-Mercury was the control group that followed the teacher's lesson plan to learn the topics. Following this approach allowed the researchers to compare the learning outcomes between the two sections and evaluate if the implementation of the game-based learning tool to teach the students about the topic significantly improved the learning outcome of the students.

4.7.3 Conduct of Study

The testing of the system created by this study was done by using pre-test and post-test questionnaires. For the experimental group (8-Mars), a pre-test regarding the topics per chapter was conducted every time before playing the game, then proceeded to play the game, and finally, took a post-test after to evaluate their learning from playing the game. Meanwhile, the control group (8-Mars) followed the teacher's lesson plan flow to complete the topic. On the Final Day of testing, the researchers conducted a summative test about the topics discussed and were taken by both sections (8-Mars and 8-Mercury).

Date	Activities

March 2, 2023	• Conduct pre-test about the topic bacteria
	• Play game chapters containing topics about bacteria
	  ○ Definition of bacteria
	  ○ Modes of Transmission
	  ○ Prevention and/or Cure
	  ○ Factors and Causes of the disease/s
	  ○ Signs and Symptoms of disease/s
	  ○ Common misconceptions, myths and beliefs about the disease/s
	• Conduct post-test about the topic bacteria

March 3, 2023	• Conduct pre-test about the topic viruses • Play game chapters containing topics about viruses • Conduct post-test about the topic viruses
March 17, 2023	• Conduct pre-test about the topic viruses • Play game chapters containing topics about viruses • Conduct post-test about the topic viruses
March 22, 2023	• Conduct pre-test about the topic viruses • Play game chapters containing topics about viruses • Conduct post-test about the topic viruses
March 23, 2023	• Conduct Summative Assessment

Table 4.2: Schedule of Activities for Testing

After taking the pre and post-tests, the researchers compared the results of the students' pre-test and post-test scores to find out their learning outcomes from playing the game and using the traditional way of teaching. Then afterward, the experimental group's results were compared to the control group to find out if the use of the system has indeed improved the students' learning outcome compared to the traditional way of teaching.

4.7.4 Statistical Analysis Tools

The study made use of Independent and Paired T-Test to evaluate data that were gathered from the study.

Independent T-Test

The study's data were evaluated using the Independent T-test to compare the mean results of summative tests between two sections, 8-Mercury, and 8-Mars. This test helps identify significant differences in learning outcomes between the experimental group, which tested the game, and the controlled group, which received traditional lessons.

The formula for Independent T-Test is:

$$t = \frac{M_1 - M_2}{\sqrt{(s^2(\frac{1}{n_1} + \frac{1}{n_2}))}}$$

Where $M_!$ and M_2 are the means of the two groups being compared; s^2 is the pooled standard error of the two groups; and n_1, and n_2 is the sample size of each of the groups.

Paired T-test

The data from the experimental group's pre-tests and post-tests were evaluated using Paired T-Test. Paired T-Test is used when comparing results from groups from a single population. In this case, the Paired T-Test was used to compare the pre-test and post-test results of the experimental group. This is to identify the significance of using the system to the learning outcome of the students who played the game (8-Mars). Using this analysis tool helped the researchers evaluate the system's effectiveness to the players.

The formula for Paired T-Test is:

$$t = \frac{\overline{d} - 0}{\sigma/\sqrt{n}}$$

Where $\overline{d}$ is the sample mean difference; σ is the sample standard deviation of the differences; and n is the sample size.

Chapter 5

Results and Discussions

This chapter provides the findings of relevant results and discussion about the test conducted on Grade 8 students of Camarines Sur National High School. The main goal was to test whether the implementation of a game-based learning environment is an effective tool in teaching students who are taking MAPEH subject, using Paired T-Test and Independent T-Test, as well as to evaluate the usability and effectiveness of the game-based application through a usability testing.

5.1 Paired T-Test for Pre- and Post Test Results

		n	Mean	SD	p-value (one-tailed)	Interpretation (alpha=0.05)
Day 1	Pre-test	36	7.5833	1.6966	0.00000077	Significant
	Post test	36	9.0278	1.1081		
Day 2	Pre-test	47	7.9574	1.4136	0.10322759	Not Significant
	Post test	47	8.3404	1.7229		
Day 3	Pre-test	44	7.2727	1.7962	0.03100291	Significant
	Post test	44	7.9545	1.5243		
Day 4	Pre-test	46	6.4130	1.6677	0.00076814	Significant
	Post test	46	7.3261	2.0446		

Table 5.1: Paired t-test results of experimental group

The test conducted on the experimental group aimed to identify if the use of the system Dr. Abel's Adventure provided a significant difference in the students' learning outcome. The researchers

hypothesized that using the system would provide a greater Post-test score than a Pre-test score. To test this hypothesis, paired t-test was used to analyze the pre-test and post-test results of the students.

Figure 5.1 shows the pre-test and post-test results of the four-day experiment the researchers conducted. Day 1 results show sufficient evidence that the Post-test (Mean=9.0278, SD=1.1081) is greater than the Pre-test (Mean=7.5833, SD=1.6966) scores of the students. Day 2 results show insufficient evidence that the Post-test (Mean=8.3404, SD=1.7229) is greater than the Pre-test (Mean=7.9574, SD=1.4136) scores of the students. Day 3 results show sufficient evidence that the Post-test (Mean=7.9545, SD=1.5243) is greater than the Pre-test (Mean=7.2727, SD=1.7962) scores of the students. Day 4 results show sufficient evidence that the Post-test (Mean=7.3261, SD=2.0446) is greater than the Pre-test (Mean=6.4130, SD=1.6677) scores of the students. These findings show that using Dr. Abel's Adventure has significantly improved the learning outcome of the students in the Experimental group three out of four times (75%) which is satisfactory or average performance.

5.2 Independent T-Test for Summative Test Results

		n	Mean	SD	p-value (two-tailed)	Interpretation (alpha=0.05)
Summative Test	8 – Mars	51	20.3333	2.7031	0.27335885	Not Significant
	8 – Mercury	45	19.8000	1.9141		

Table 5.2: Independent t-test result of the experimental and controlled group

The study also aimed to determine if there is a difference in the learning outcome of the students who played the game versus those who received the current teaching method that the school is implementing. The researchers hypothesized that the Experimental group (8-Mars) who played the game would have higher scores than the Controlled group (8-Mercury). To test the researchers' hypothesis, an independent t-test was conducted to analyze the summative test scores of both groups.

Figure 5.2 shows the result of the independent t-test that was conducted. The results show that even though the score of the Experimental group (8-Mars) is higher than the Controlled group (8-Mercury), there is no sufficient evidence that the summative test scores of 8-Mars (Mean=20.3333, SD=2.7031) is statistically different from the summative test scores of 8-Mercury (Mean=19.8000,

SD=1.9141).

5.3 System Usability Scale

The System Usability Scale (SUS) was used to determine the game's usability from the perspective of students and the teacher involved in the experimental group and identify areas in the system that need improvement. The SUS survey was administered using Google Forms, which consists of ten (10) questions with five responses, 1 being 'strongly disagree' and 5 being 'strongly agree' [5], which provides an efficient way of obtaining user feedback on their experience playing the game 'Dr. Abel's Adventure'.

Items	Questions	1	2	3	4	5
1	I think that I would like to play this game frequently.					
2	I found this game unnecessarily complex.					
3	I thought this game was easy to play					
4	I think that I would need assistance to be able to play this game					
5	I found the various functions in this game were well integrated.					
6	I thought there was too much inconsistency in this game.					
7	I would imagine that most people would learn to use this system very quickly.					
8	I found this game very difficult to use.					
9	I felt very confident playing this game.					
10	I needed to learn a lot of things before I could get going with this system					

Table 5.3: System Usability Scale

5.3.1 Scoring SUS

First, to calculate the SUS score, sum the score contributions from each item. Each item's score contribution will range from 1 to 5. For items 1, 3, 5, 7, and 9, the score contribution is the scale position minus 1. For items 2, 4, 6, 8, and 10, the contribution is 5 minus the scale position. Finally, multiply the sum of the scores by 2.5 to obtain the overall value of SUS [47] [6].

5.3.2 SUS Result From the Teacher's Perspective

After creating a 100% working prototype of the game, the researchers asked the teacher who handles the two sections involved in the study to play and evaluate the game by answering a SUS questionnaire. The survey was done to allow the teacher to assess the game before it gets released to our target users, which are the students, and also to gain insights into which features of the game the teacher has found satisfactory or need improvement. From the teacher's perspective, the game received a 52.5 SUS score (Table 5.4), which equates to a "poor" rating [44]. Based on this result, the researchers further improved on the areas identified by the teacher during our face-to-face consultation: Chapter code system, Introduction of Game Elements (Trust system, Chest, etc.), and some Game designs.

Items	Questions	Responses
1	I think that I would like to use this game frequently.	3
2	I found this game unnecessarily complex.	3
3	I thought this game was easy to use.	4
4	I think that I would need assistance to be able to play this game.	4
5	I found the various functions in this game were well integrated.	4
6	I thought there was too much inconsistency in this game.	4
7	I would imagine that most people would learn to use this system very quickly.	4
8	I found this game very difficult to use	3
9	I felt very confident playing the game	3
10	I needed to learn a lot of things before I could get going with this system.	3
	Average	52.5

Table 5.4: SUS report for the teacher handling the experimental group

5.3.3 SUS Result from the Students' Perspective

After the revisions based on the client's feedback were completed, the game was released to the target users (8-Mars) to be played and tested. After playing the game, a SUS survey was also conducted to the users to get insight into their experience playing Dr. Abel's Adventures.

	Rater																	
Question	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
1	5	4	5	5	5	5	5	4	4	5	3	4	5	3	4	4	4	3
2	1	5	4	5	2	1	1	3	3	4	1	3	3	3	2	2	3	2
3	3	4	4	5	5	5	4	3	4	3	1	5	3	2	3	3	3	3
4	1	4	3	5	3	1	1	2	5	2	1	3	2	4	2	3	2	1
5	5	3	2	5	3	5	5	4	4	3	5	5	3	3	4	3	3	5
6	1	4	3	5	3	2	2	2	3	2	5	2	3	3	2	2	4	2
7	5	4	5	3	5	5	4	4	5	5	4	5	3	2	5	4	5	4
8	1	5	2	5	2	1	1	1	3	1	1	1	3	3	1	2	2	1
9	5	4	4	3	4	5	5	5	2	4	4	5	3	3	4	3	5	4
10	1	4	4	4	5	1	1	4	4	1	1	5	3	5	1	4	4	2
SUS Score	95	42.5	60	42.5	67.5	97.5	92.5	70	52.5	75	70	75	57.5	37.5	80	60	62.5	77.5
Average	67.5																	

Figure 5.1: SUS Report for Students

Of the 52 students from 8-Mars who played the game, 18 students (players) responded to the System Usability Scale Survey. Based on the results, the average SUS score is 67.5, which suggests that the usability performance of the proposed game based on the student is 'poor' [44].In order to gain additional insights into specific aspects of the users' experience, the researchers calculated the scores for individual questions. The researchers deemed that individual scores lower than the overall SUS score (67.5) indicate areas of concern.

Question	Score	Difference from Teacher's SUS score
I think that I would like to use this system frequently.	81.9	+31.9
I found the system unnecessarily complex.	58.3	+8.3
I thought the system was easy to use.	62.5	-12.5
I think that I would need the support of a technical person to be able to use this system.	62.5	+37.5
I found the various functions in this system were well integrated.	72.2	-2.8
I thought there was too much inconsistency in this system.	55.5	+30.5
I would imagine that most people would learn to use this system very quickly.	81.9	+6.9
I found the system very cumbersome to use.	75	+25
I felt very confident using the system.	75	+25
I needed to learn a lot of things before I could get going with this system.	50	0

Table 5.5: SUS individual questions Report for Students

5.3.4 Learning How To Play Dr. Abel's Adventures can be difficult

Learnability is a component of usability that looks at how easy it is for the users to accomplish basic tasks the first time they interact with the system. The result shows that Items (2, 3, 4, and 10) related to the Learnability of Dr. Abel's Adventures have received scores lower than the overall SUS score (67.5).

2. I found the system unnecessarily complex.	
Students SUS score	58.3
Difference from teacher's SUS score	+8.3

4. I think that I would need the support of a technical person to be able to use this system.	
Students SUS score	62.5
Difference from teacher's SUS score	+37.5

3. I thought the system was easy to use.	
Students SUS score	62.5
Difference from teacher's SUS score	-12.5

10. I needed to learn a lot of things before I could get going with this system.	
Students SUS score	50
Difference from teacher's SUS score	+0

Figure 5.2: SUS scores for items 2, 3, 4, and 10

The respondents identified features that the game is lacking and that contributed to their low rating, which mainly are: Lack of in-game instructions and The use of chapter codes. The respondents commented that the implementation of chapter codes made the use of the game restrictive and only playable whenever the researchers gave them codes for each chapter. Additionally, the respondents pointed out that the lack of instructions during their first game played made it difficult to identify what each button did.

Chapter 6

Contributions and Recommendations

6.1 Summary of Contributions

The major contribution of this research was to develop a game-based learning system that could enhance the knowledge and awareness of Grade 8 students at Camarines Sur National High School regarding the topics and concepts of Prevention and Control of Communicable Diseases and Disorders, which are particularly relevant nowadays due to the ongoing COVID-19 pandemic. Additionally, this research aimed to assess the effectiveness of implementing the game-based learning system as a teaching tool for students taking MAPEH subjects. This assessment helps determine whether a new mode of teaching and learning can effectively impart knowledge to the students. Furthermore, one significant contribution of the mobile game, 'Dr. Abel's Adventure', is that it does not require an internet connection, and it requires low mobile system requirements making it more accessible for students. Finally, this project can serve as a reference for the Department of Education, Naga City (DepEd Naga), the Camarines Sur National High School (CSNHS), and other researchers interested in further studies and teaching and learning innovation concerning 'Game-Based Learning.'

6.2 Recommendations

The study showed that using a game-based learning tool in education contributed positively to the players' learning outcomes. Even though its results do not differ from the existing teaching methods currently being implemented, further improvements in the field can still be made to achieve greater results. Given that this study received a "poor" rating in terms of usability, specifically in the area of Learnability, The researchers recommend future studies consider creating a better design on instructions and introduction of game elements such as the implementation of a "help" module where users can look up how specific buttons in the game work, or a more thorough tutorial level to introduce the users about the game controls, game elements, and gameplay can help address the issue of the system's learnability. Also, focusing on improving the game design to make the game more interesting to the players and keep them engaged while playing will also help success for future studies. Based on the feedback from the players who tested Dr. Abel's Adventure, having a linear design on NPC interactions takes too long, and makes their game experience boring, resulting in loss of interest over time. Improvements on the design of the game by creating the dialogues as concise as possible, and adding a mechanic where players can leave the conversation temporarily, to find clues in the map should be considered to avoid losing the interest of the players towards the game.

Additionally, since Dr. Abel's Adventure was developed to be played in a fixed timeline, chapter codes were used to load chapters of the game to control the players' game time. This was done to avoid bias in their knowledge during testing days. Instead of making this approach, designing a chapter select module to allow the players to have easier access to the chapters to make the replayability of the game easier and hassle-free. Finally, creating an activity log to track how the players interact with the objects within the game can help future studies identify if the game elements placed in the game were effective and helpful to the players' completion of the game.

Chapter 7

Conclusion

With the poor health literacy that the Philippines is facing despite the current efforts of the government to integrate health-related subjects in the DepEd curriculum to teach the students as early as possible, finding other means to educate the people can be helpful to increase the reach of their efforts in improving the country's health literacy. This study aimed to use the popularity of video games to the masses to create an accessible game-based learning tool and determine its effectiveness in teaching topics about the Prevention and Control of Communicable Diseases. By creating Dr. Abel's Adventure and testing it on the students of Camarines Sur National High School, the study has found that using Game-based learning in teaching topics about Prevention and Control of Communicable diseases helped improve the learners' knowledge and awareness about the subject. Although the learning outcome of the Experimental group that tested the game does not significantly differ from the learning outcome of the Control group, the study was still able to serve its original purpose, which was to help enhance its players' health literacy. Additionally, the findings from this study have shown the importance of having thorough instructions and introduction of the game to the users in providing a good user experience. Implementing a rundown on each element in the game can help the users accomplish actions in the game easily, even on their first try.

Appendix A

GANTT Chart

A.1 Preparation Phase

Successive Approximation Model (SAM)	Task Name	August 2022		September 2022		October 2022		Nov 2022	...	Status
PREPARATION PHASE	*Project Conceptualization and Project Revision*	Aug.3 - 28								Completed
	Introduction and Related System and Literature Design		Aug.29 - Sep.11							Completed
	Theoretical Framework and Methodology Design			Sep.12 - 25						Completed
	Finalization of Storyline and Game Mechanics			Sep.19 - 25						Completed
	Find and Approach Clients (Central and Camhigh)				Sep.29 - Oct.14					Completed

A.2 Design Phase

Successive Approximation Model (SAM)	Task Name	September 2022		October 2022		November 2022		Dec 2022	...	Status
DESIGN PHASE	*Diagramming*		Sep.19-29							Completed
	Creating Game Storyboard		Sep. 25 - OCt.14							Completed

A.3 Development Phase

Successive Approximation Model (SAM)	Task Name	September 2022		October 2022		November 2022		December 2022		January 2023		...	Status
DEVELOPMENT PHASE	*Interface Design*		September - January 31										Completed
	Incorporating Health Topics			October 15 - January 31									Completed
	Coding (Interaction, Movement, etc.)					November 15 - January 31							Completed
	Client Consultation				Oct.18	Nov.8		Dec.1-9			Jan.20		Completed
	Debugging and Game Finalization									Jan.16 - 27			Completed

A.4 Evaluation Phase

Successive Approximation Model (SAM)	Task Name	...	Feb 2023	March 2023		April 2023		May 2023		Status
EVALUATION PHASE	*Distribution of APK*		Feb.15							Completed
	Distibution and Conduct of Pre-test		Feb.15	March 2 - 22						Completed
	Actual Game Testing			March 2 - 22						Completed
	Distibution and Conduct of Post-test			March 2 - 22						Completed
	Summative Assessment				Mar.23					Completed
	Results Collection		Feb.15 - March 2 - 22							Completed
	Results Evaluation		February 15 - 1st week of April							Completed

A.5 Extras

Task Name	Sep.2022	Oct.2022	Nov.2022	Dec.2022	Jan.2023	Feb.2023	Mar.2023	Apr.2023	May.2023	Status
Defense	Sep.28		Nov.18				Mar.09		May.04	Completed
Revision of Document	August 3 - May 21									Completed

Appendix B

Certifications

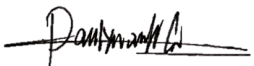
B.1 Proofreading Certificate

C E R T I F I C A T E O F P R O O F R E A D I N G

This is to certify that the undersigned has reviewed and been proofread carefully the Thesis Manuscript entitled "**Dr. Abel's Adventures: A Game Based Application to Enhance Knowledge and Awareness About Communicable Diseases**" developed by **Josef Rex O. Armenta** and **Jefen B. De Villa** is aligned with the set of structural rules that govern the composition of sentences, phrases, and words in the English language. Also, all corrections and recommendations made have been done and/or incorporated in the final manuscript.

Issued this 27th day of June 2023.

Signed:



PAUL CHRISTIAN R. ABAD, MBA
License No. 1227237

Master in Business Administration
University of Nueva Caceres, 2019

Bachelor of Science in Business Administration
Major in Computer Management and Accounting
Ateneo de Naga University, 2006

Senior High School English Subject Teacher
Camarines Sur National High School, 2016-Present

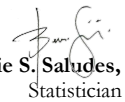
English for Career Development
University of Pennsylvania, August 10, 2020-September 3, 2020

Grammarian / English Editor

B.2 Statistician Certificate

CERTIFICATION

This is to certify that the research paper entitled **"Dr. Abel's Adventures: A Game Based Application to Enhance Knowledge and Awareness About Communicable Diseases"** prepared and submitted by Josef Rex O. Armenta and Jefen B. De Villa in partial fulfilment of the requirements for the degree Bachelor of Science in Information Technology has been assisted and provided with statistical treatment and data analysis by the undersigned.

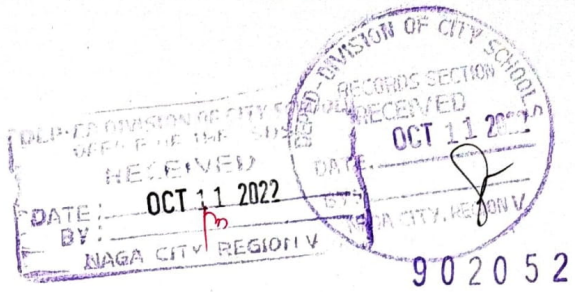


Benjie S. Saludes, MMath
Statistician

Appendix C

Consent Forms

C.1 Letter of Request



October 8, 2022

MARIANO B. DE GUZMAN CESO VI
Schools Division Superintendent
Division of Naga City

Good day Sir,

We are 4th year BS Information Technology students from Ateneo de Naga University taking our Capstone Project 1. We are writing this letter to formally ask for your permission to conduct our research study entitled **"Dr. Kwak Kwak: A Game Based Application to Enhance Knowledge and Awareness About Communicable Diseases"** inside Camarines Sur National High School.

Dr. Kwak Kwak is a Game Based Learning (GBL) system that we are currently working on which aims to help in the efforts in teaching the students about communicable diseases in an engaging and fun way. The main purpose of this study is to help in enhancing the students' Health Literacy especially about communicable diseases to help them be prepared in future situations like the COVID-19 pandemic, and also to evaluate the effectiveness of our system as a teaching tool in teaching Health related topics to Grade 8 students. As part of our Capstone Project we will develop a mobile game that will integrate the topics in Grade 8 Health Subject specifically about the Prevention and Control of Communicable Diseases and Disorders, and test the said system with the students to determine the effectiveness of Game Based Learning as a teaching tool.

Rest assured that the data gathered in this study will be treated with utmost confidentiality and to be used for academic purposes only. If you have any questions or concerns, feel free to contact us at 09163535365 or at jrarmenta@gbox.adnu.edu.ph.

Thank you for taking the time to consider our request. Your approval will be of great help for the success of this project.

Sincerely,

Josef Rex O. Armenta
Proponent

Jefen B. De Villa
Proponent

Gian Xavier Petalio
Proponent

PRINCESS (DEPED - SDS OFFICE)
NAGA

* 0951 318 2702

Endorsed by:

Raphael Henry M. Garay
Capstone Project Adviser
Ateneo de Naga University

C.2 Endorsement Letter

Republika ng Pilipinas
Kagawaran ng Edukasyon
REHIYON 5
SANGAY NG MGA PAARALAN NG LUNGSOD NG NAGA

OCT 14 2022

1st ENDORSEMENT
October 14, 2022

Respectfully forwarded to the **Principal of Camarines Sur National High School** the herein letter request of **Mr. Josef Rex O. Armenta, Mr. Jefen B. De Villa and Mr. Gian Xavier Petalio** of **Ateneo de Naga University** to conduct data gathering with teachers and Grade 8 students in relation to their study, **"Dr. Kwak Kwak: A Game Based Application to Enhance Knowledge and Awareness about Communicable Diseases".** It is hereby enjoined that the researcher must observe the *Republic Act 10173 (Data Privacy Act of 2012)* in gathering data. This office shall be furnished with a copy of the findings and recommendations upon completion of the study.

MARIANO B. DE GUZMAN CESO VI
Schools Division Superintendent

PRS ETCR 2022
Address: Roxas Avenue, Triangulo, Naga City
Telephone No.: 054-871-3455
Email Address: naga.city@deped.gov.ph
Website: www.depednaga.ph

NAGA THRIVES
Page 1 of 1

Appendix D

Evaluation Tools

D.1 Survey Questionnaires

SURVEY QUESTIONNAIRE

Magandang Araw!

Kami ay grupo ng mga mag-aaral mula sa Ateneo de Naga University na kumuha ng kursong BS Information Technology (IT) na kasalukuyang nagsasagawa ng aming *Capstone Project* na may pamagat na *"Dr Kwak Kwak: A Game Based Application to Enhance Knowledge and Awareness About Communicable Diseases".*

Kaugnay nito inihanda namin ang talatanungan upang makapangalap ng mga datos na kailangan sa aming pananaliksik, kung gayon, maaaring sagutan ng may katapatan ang mga sumusunod na aytem. Tinitiyak namin na ang mga impormasyong ibabahagi ay mananatiling kompidensyal.

Maraming Salamat po!

- Mga Mananaliksik

PANUTO: Punan ng angkop na impormasyon ang mga sumusunod na aytem. Lagyan ng ng tsek (/) ang patlang o kahon na nagrerepresenta sa iyong sagot.

Pangalan (opsyonal): _____
Seksyon: _____ Kasarian: Lalaki __ Babae __ Edad: __

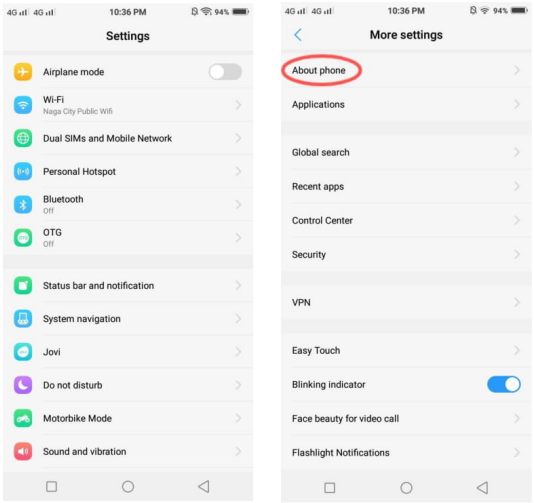
1. Meron ka bang *mobile phone*?
   ☐ Meron
   ☐ Wala *(kung "wala" ang iyong sagot pwede ka nang huminto at ibalik ang questionnaire sa mga mananaliksik)*

2. Ano ang *operating system* ng iyong cellphone?
   ☐ Android
   ☐ Apple / IOS *(kung ito ang sagot mo pwede ka nang huminto at ibalik ang questionnaire sa mga mananaliksik)*

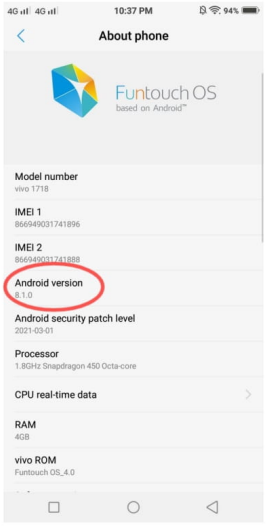
3. Ano ang *android version* ng iyong cellphone? Kung hindi alam kung paano kunin ang android version, pwedeng sumangguni sa likod na bahagi ng questionnaire
   ☐ 4.0 pababa ☐ 5.0 - 6.0
   ☐ 4.1 - 4.4 ☐ 7.0 - 9.0 pataas

INSTRUKSYON SA PAGKUHA NG *ANDROID VERSION*

1. Buksan ang cellphone at pumunta sa “Settings”
2. Mag scroll sa baba at hanapin ang “About Phone”



3. I-click ang “About Phone” hanggang sa makita ang “Android Version”



D.2 Day 1 Test Questionnaires

D.2.1 Pretest Questionnaire

PRETEST QUESTIONNAIRE

Name: _____ Section: _____ Date: _____

Direction: Encircle the letter of the correct answer.

1. Communicable diseases are diseases that _____?
    a. Are passed from parents to children through gene mutation
    b. Can be passed from a host to another
    c. Cannot be passed from a host to another

2. Which type of Pathogen is described as a single-celled organism, and most widespread among living things?
    a. Bacteria
    b. Virus
    c. Fungi

3. Which of the following is/are symptom/s of Urinary Tract Infection (UTI)?
    a. Constant urge to urinate
    b. Burning feeling when urinating
    c. Both a and b

4. Which of the following is/are symptom/s of Tuberculosis?
    a. Coughing up blood
    b. Circular Rashes
    c. Both a and b

5. Which of the following is/are symptom/s of Typhoid Fever?
    a. Fever
    b. Diarrhea
    c. Both a and b

6. What bacteria causes Typhoid Fever?
    a. Salmonella Typhi
    b. E. coli
    c. Lactobacillus

7. Which of the following is/are a good practice/s to prevent getting Urinary Tract Infection (UTI)?
    a. Holding your pee
    b. Drinking plenty of fluids
    c. Both a and b

8. Which of the following is/are a good practice/s to control the spread of Tuberculosis?
    a. Visiting friends to get support
    b. Covering mouth when coughing
    c. Both a and b

9. Aside from securing a non contaminated drinking water, what else is a good preventive measure to avoid getting infected by Typhoid Fever?
    a. Frequent washing of hands
    b. Drinking alcoholic beverages
    c. Having a well ventilated room

10. How can Tuberculosis be passed from one person to another?
    a. Eating food near an infected person
    b. Shaking hands with an infected person
    c. Contact with droplets from an infected person

D.2.2 Post-test Questionnaire

POST-TEST QUESTIONNAIRE

Name: _____ Section: _____ Date: _____

Direction: Encircle the letter of the correct answer.

1. What is the term used to describe diseases that can be passed from a host (person, animal, things) to another host?
    a. Hereditary Disease
    b. Expanding Disease
    c. Communicable Disease
2. Which type of Pathogen is described as a single-celled organism, and most widespread of living things?
    a. Bacteria
    b. Virus
    c. Fungi
3. What disease infects the urinary system, causing persons infected with this disease experience a strong urge to urinate that doesn't go away, as well as burning feeling when urinating?
    a. Urinary Tract Infection (UTI)
    b. Tuberculosis
    c. Strep Throat
4. What disease usually affects the lungs, causing chest pain, coughing, and in worse cases coughing up blood?
    a. Tuberculosis
    b. Athlete's Foot
    c. Measles
5. What disease is caused by "Salmonella Typhi", which causes the infected person fever, nausea, abdominal pain, and diarrhea?
    a. Strep Throat
    b. Salmonisis
    c. Typhoid Fever
6. Which of the following is one of the common causes of Typhoid Fever?
    a. Drinking Contaminated Water
    b. Inhaling Contaminated Air
    c. Talking to an infected person
7. What is a suitable approach to prevent getting Urinary Tract Infection (UTI)?
    a. Drinking plenty of fluids, especially water.
    b. Having a complete 8 hours of sleep.
    c. Always keep your feet dry.
8. What is a good prevention measure to control the spread of Tuberculosis?
    a. Covering mouth when sneezing or coughing.
    b. Taking a bath regularly.
    c. Hangout with friends more often.
9. What is a good prevention measure to avoid getting typhoid fever?
    a. Frequent washing of hands
    b. Taking plenty of rest.
    c. Eat a lot of delicious food.
10. How is Tuberculosis transmitted from one person to another?
    a. by shaking hands with a person with Tuberculosis.
    b. by inhaling air contaminated with Tuberculosis germs.
    c. by sharing towels.

D.3 Day 2 Test Questionnaires

D.3.1 Pretest Questionnaire

PRETEST QUESTIONNAIRE

Name: _____ Section: _____ Date: _____

Direction: Encircle the letter of the correct answer.

1. Which type of Pathogen is described as an extremely small organism, and uses its host to survive and reproduce?
    a. Bacteria
    b. Virus
    c. Protozoa

2. What type of Pathogen caused the COVID-19 Pandemic?
    a. Bacteria
    b. Virus
    c. Fungi

3. Fever, cough, body aches, and chills are symptoms of what disease?
    a. Influenza
    b. Warts
    c. Athlete's foot

4. What disease causes rough grainy bumps in the skin of the infected person?
    a. Acne
    b. Ringworm
    c. Warts

5. What disease causes Coughing, and Loss of taste and smell to the infected person?
    a. Warts
    b. Tonsillitis
    c. COVID-19

6. Which of the following is the best prevention measure for COVID-19?
    a. Vaccination
    b. Vasectomy
    c. Vacation

7. _____ is a disease outbreak that spans several countries and affects a large number of people.
    a. Pandemic
    b. Endemic
    c. Paramedic

8. What is a good preventive measure to avoid spreading COVID-19?
    a. Covering mouth when coughing or sneezing
    b. Staying isolated from others
    c. Both a and b

9. What is a good preventive measure to avoid getting infected by Influenza?
    a. Frequent hand washing
    b. Only eating expensive food
    c. Drinking Antibiotics daily

10. What is a good way to cure Warts?
    a. Application of Salicylic acid
    b. Drinking Antibiotics
    c. Sleeping 8 hours a day

D.3.2 Post-test Questionnaire

POST-TEST QUESTIONNAIRE

Name: _____ Section: _____ Date: _____

Direction: Encircle the letter of the correct answer.

1. Which type of Pathogen is described as an extremely small organism, and uses its host to survive and reproduce?
    a. Bacteria
    b. Virus
    c. Protozoa
2. What type of Pathogen causes COVID-19 Pandemic?
    a. Bacteria
    b. Virus
    c. Fungi
3. Which of the following is/are a symptom/s of Influenza?
    a. Fever, and cough
    b. Body aches and chills
    c. Both a and b
4. Which of the following is/are symptom/s of Warts?
    a. Rough grainy bumps
    b. Smooth silky wounds
    c. Both a and b
5. Which of the following is/are symptom/s of COVID-19?
    a. Cough
    b. Loss of taste and smell
    c. Both a and b
6. Which of the following is the best prevention measure for COVID-19?
    a. Vaccination
    b. Vasectomy
    c. Vacation
7. _____ is a disease outbreak that spans several countries and affects a large number of people.
    a. Pandemic
    b. Endemic
    c. Paramedic
8. What is a good preventive measure to avoid spreading COVID-19?
    a. Covering mouth when coughing or sneezing
    b. Staying isolated from others
    c. Both a and b
9. What is a good preventive measure to avoid getting infected by Influenza?
    a. Frequent hand washing
    b. Only eating expensive food
    c. Drinking Antibiotics daily
10. What is a good way to cure Warts?
    a. Application of Salicylic acid
    b. Drinking Antibiotics
    c. Sleeping 8 hours a day

D.4 Day 3 Test Questionnaires

D.4.1 Pretest Questionnaire

PRETEST QUESTIONNAIRE

Name: _____ Section: _____ Date: _____

Direction: Encircle the letter of the correct answer.

1. The medical term for ringworm is _____?
    a. Rash
    b. Tinea
    c. Flea
2. In what type of environment does fungi thrive the most?
    a. Moist and sunny places
    b. Dark and damp places
    c. Dark and dry places
3. Where are the common places where fungal infection takes place?
    a. Hair, nails and skin
    b. Gall bladder and Intestines
    c. Heart and lungs
4. A skin infection that usually begins between the toes. It commonly occurs in people whose feet have become very sweaty while confined within tight-fitting shoes.
    a. Athlete's foot
    b. Jock itch
    c. COVID-19
5. An infection that causes red and itchy rashes in warm moist areas of the body, commonly near the groin and inner thighs.
    a. Athlete's foot
    b. Measles
    c. Jock itch
6. What fungal infection causes itchy, circular rash with clearer skin in the middle resembling a figure of a ring?
    a. Ring rash
    b. Roundworm
    c. Ringworm
7. What Pathogen Causes Athlete's Foot?
    a. Bacteria
    b. Fungi
    c. Parasites
8. You can only get Athlete's foot if you are an athlete.
    a. True
    b. False
9. What is a good practice to prevent getting Athlete's foot?
    a. Keep feet dry at all times
    b. Wear shoes with wet feet
    c. Use a friend's used socks
10. Which of the following situations can spread an athlete's foot from one person to another?
    a. Sharing of used socks
    b. Sneezing in a closed room
    c. Sharing food with friend

D.4.2 Post-test Questionnaire

POST-TEST QUESTIONNAIRE

Name: _____ Section: _____ Date: _____

Direction: Encircle the letter of the correct answer.

1. Ringworms are caused by _____.
    a. Worms
    b. Tinea
    c. Rings
2. In what type of environment does fungi thrive the most?
    a. Moist and sunny places
    b. Dark and damp places
    c. Dark and dry places
3. Where are the common places where fungal infection takes place?
    a. Hair, nails and skin
    b. Gall bladder and Intestines
    c. Heart and lungs
4. Which of the following is/are a symptom/s of Athlete's Foot?
    a. Itchy and scaly rashes on toes
    b. Sweaty feet
    c. Both a and b
5. Which of the following is/are a symptom/s of Jock Itch?
    a. Itchy rashes on groin and hip area
    b. Scaly rashes between toes
    c. Both a and b
6. Which of the following is/are a symptom/s of Ringworm of the scalp?
    a. Scaly bald patches on the head
    b. Dandruff
    c. Both a and b
7. What Pathogen Causes Athlete's Foot?
    a. Bacteria
    b. Fungi
    c. Parasites
8. Who can be infected with Athlete's foot?
    a. Athletes
    b. Runners
    c. Anyone
9. What is a good practice to prevent getting Athlete's foot?
    a. Keep feet dry at all times
    b. Wear shoes with wet feet
    c. Use a friend's used socks
10. Which of the following situations can spread an athlete's foot from one person to another?
    a. Sharing of used socks
    b. Sneezing in a closed room
    c. Sharing food with friends

D.5 Day 4 Test Questionnaires

D.5.1 Pretest Questionnaire

PRETEST QUESTIONNAIRE

Name: _____ Section: _____ Date: _____

Direction: Encircle the letter of the correct answer.

1. Helminths are pathogens referred to as _____.
    a. Virus
    b. Parasitic worms
    c. Protozoa
2. What do you call a disease-causing organism that lives in a humans or other animals and derive its nourishment from its host
    a. Virus
    b. Parasite
    c. Bacteria
3. Which of the following is classified as a parasite?
    a. Worm
    b. Chicken
    c. Rat
4. Which of the following is a good practice to prevent helminth diseases?
    a. Getting vaccinated
    b. Drinking water 3x a day
    c. Frequent hand washing
5. Which type of helminth does “Trichinella” belong to?
    a. Roundworm
    b. Tapeworm
    c. Ringworm
6. Which parasitic infection happens when you eat undercooked pork produce?
    a. Porchiosis
    b. Trichinosis
    c. Schistosomiasis
7. What type of transmission is it called when a disease is passed by mosquito bites?
    a. Air-borne transmission
    b. Anopheles transmission
    c. Vector-borne transmission
8. Which parasite is transmitted by mosquito bites, and causes Malaria to its host?
    a. Trichinella
    b. Plasmodium
    c. Malarium
9. Which is a good practice to prevent the risk of getting infected by Malaria?
    a. Cleaning your surroundings
    b. Securing 8 hours of sleep
    c. Drinking antibiotics daily
10. Which is a good practice to prevent the risk of getting infected by Trichinosis?
    a. Never eat meat
    b. Cooking meat properly
    c. Cleaning your surroundings

D.5.2 Post-test Questionnaire

POST-TEST QUESTIONNAIRE

Name: _____ Section: _____ Date: _____

Direction: Encircle the letter of the correct answer.

1. Helminths are pathogens referred to as _____.
    a. Virus
    b. Parasitic worms
    c. Protozoa
2. What do you call a disease-causing organism that lives in a humans or other animals and derive its nourishment from its host
    a. Virus
    b. Parasite
    c. Bacteria
3. Which of the following is classified as a parasite?
    a. Worm
    b. Chicken
    c. Rat
4. Which of the following is/are a good practice/s to prevent helminth diseases?
    a. Getting vaccinated
    b. Frequent handwashing
    c. Both a and b
5. Which type of helminth does “Trichinella” belong to?
    a. Roundworm
    b. Tapeworm
    c. Ringworm
6. What parasitic disease are you at risk of when you eat undercooked pork?
    a. Porchiosis
    b. Trichinosis
    c. Schistosomiasis
7. What type of transmission is it called when a disease is passed by mosquito bites?
    a. Air-borne transmission
    b. Anopheles transmission
    c. Vector-borne transmission
8. What disease does the parasite Plasmodium cause?
    a. Schistosomiasis
    b. Malaria
    c. Pediculosis
9. Which is a good practice to prevent the risk of getting infected by Malaria?
    a. Cleaning your surroundings
    b. Securing 8 hours of sleep
    c. Drinking antibiotics daily
10. Which is a good practice to prevent the risk of getting infected by Trichinosis?
    a. Never eat meat
    b. Cooking meat properly
    c. Cleaning your surroundings

Appendix E

Code Listing

E.1 Audio Manager

```
using System.Linq;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class AudioManager : MonoBehaviour
{
    [SerializeField] List<AudioData> sfxList;

    [SerializeField] AudioSource musicPlayer;
    [SerializeField] AudioSource sfxPlayer;

    Dictionary<AudioId, AudioData> sfxLookup;

    public static AudioManager i {get; private set;}

    private void Awake()
    {
        i = this;
    }

    private void Start()
    {
        sfxLookup = sfxList.ToDictionary(x => x.id);
    }

    public void PlayMusic(AudioClip clip, bool loop=true)
    {
        if (clip == null) return;

        musicPlayer.clip = clip;
        musicPlayer.loop = loop;
        musicPlayer.Play();
    }

    public void StopMusic()
    {
        musicPlayer.Stop();
    }

    public void PlaySfx(AudioClip clip)
    {
        if (clip == null) return;

        sfxPlayer.PlayOneShot(clip);
    }

    public void PlaySfx(AudioId audioId)
    {
        if(!sfxLookup.ContainsKey(audioId)) return;

        var audioData = sfxLookup[audioId];
        PlaySfx(audioData.clip);
    }
}

```

```
57 // Trigger = Start of Dialogue
58 // ButtonPress = Press of buttons / options
59 public enum AudioId {Trigger, CorrectAnswer, WrongAnswer, GameOver, UIButton}
60
61 [System.Serializable]
62 public class AudioData
63 {
64     public AudioId id;
65     public AudioClip clip;
66 }
```

E.2 Dialogue

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  [System.Serializable]
6  public class Dialogue
7  {
8      [SerializeField] string name;
9      [TextArea][SerializeField] string line;
10     [SerializeField] string[] options;
11
12     public string Name {
13         get {return name;}
14     }
15
16     public string Line {
17         get {return line;}
18     }
19
20     public string[] Options {
21         get {return options;}
22     }
23 }
```

E.3 Quiz

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  [System.Serializable]
6  public class Quiz
7  {
8      [SerializeField] string name;
9      [TextArea][SerializeField] string line;
10     [SerializeField] string[] options;
11     [SerializeField] int answer;
12
13     public string Name {
14         get {return name;}
15     }
```

```

    public string Line {
        get {return line;}
    }

    public string[] Options {
        get {return options;}
    }

    public int Answer {
        get {return answer;}
    }
}
```

E.4 Interactable

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public interface Interactable
{
    void Interact(PlayerController player);
}
```

E.5 IPlayerTriggerable

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public interface IPlayerTriggerable
{
    void OnPlayerTrigger(PlayerController player);
}
```

E.6 Game Controller

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public enum GameState {FreeRoam, Dialogue, ChestQuiz, PatientQuiz, Paused, Cutscene, Game

public class GameController : MonoBehaviour
{
    [SerializeField] PlayerController player;
    [SerializeField] GameObject gameOverScreen;
    [SerializeField] GameObject gameClearScreen;
    [SerializeField] GameObject gamePauseScreen;
    [SerializeField] AudioClip explorationMusic;
```

```
16      [SerializeField] AudioClip dialogueMusic;
17      [SerializeField] GameObject playerControls;
18      [SerializeField] AudioClip winMusic;
19      [SerializeField] GameObject missionBox;
20      [SerializeField] GameObject heart;
21      [SerializeField] GameObject Controls;
22
23      GameState state;
24      GameState prevState;
25      GameState stateBeforePause;
26
27      public static GameController Instance {get; private set;}
28
29      string testfile = "test";
30
31      private void Awake()
32      {
33          Instance = this;
34      }
35
36      private void Start()
37      {
38          Application.targetFrameRate = 300;
39          SavingSystem.i.Load(testfile);
40          player.CurrentChapter = PlayerPrefs.GetInt("CurrentChapter", 1);
41          player.ShowHP();
42
43          if(player.Progress > 0)
44          {
45              Missions.Instance.NextMission(player.currentMission);
46          }
47
48          AudioManager.i.PlayMusic(explorationMusic);
49
50          DialogueManager.Instance.OnStartDialogue += () =>
51          {
52              AudioManager.i.PlayMusic(dialogueMusic);
53              prevState = state;
54              state = GameState.Dialogue;
55          };
56          DialogueManager.Instance.OnFinishDialogue += () =>
57          {
58              AudioManager.i.PlayMusic(explorationMusic);
59              SavingSystem.i.Save(testfile);
60              state = prevState;
61          };
62          ChestQuizManager.Instance.OnChestInteraction += () =>
63          {
64              AudioManager.i.PlayMusic(dialogueMusic);
65              state = GameState.ChestQuiz;
66          };
67          ChestQuizManager.Instance.OnFinishQuiz += () =>
68          {
69              AudioManager.i.PlayMusic(explorationMusic);
70              SavingSystem.i.Save(testfile);
71              state = GameState.FreeRoam;
72          };
73          PatientQuizManager.Instance.OnPatientInteraction += () =>
```

```
74          {
75              AudioManager.i.PlayMusic(dialogueMusic);
76              state = GameState.PatientQuiz;
77          };
78          PatientQuizManager.Instance.OnFinishQuiz += () =>
79          {
80              AudioManager.i.PlayMusic(explorationMusic);
81              SavingSystem.i.Save(testfile);
82              state = GameState.FreeRoam;
83          };
84          PatientQuizManager.Instance.OnGameOver += () =>
85          {
86              playerControls.SetActive(false);
87              gameOverScreen.SetActive(true);
88              missionBox.SetActive(false);
89              state = GameState.GameOver;
90          };
91          PatientQuizManager.Instance.OnGameClear += () =>
92          {
93              SavingSystem.i.Save(testfile);
94              AudioManager.i.PlayMusic(winMusic);
95              playerControls.SetActive(false);
96              gameClearScreen.SetActive(true);
97              missionBox.SetActive(false);
98              state = GameState.GameClear;
99          };
100     }
101
102     public void PauseGame(bool pause)
103     {
104         if(pause)
105         {
106             stateBeforePause = state;
107             state = GameState.Paused;
108         }
109         else
110         {
111             state = stateBeforePause;
112         }
113     }
114
115     public void StartCutsceneState()
116     {
117         state = GameState.Cutscene;
118     }
119
120     public void StartFreeRoamState()
121     {
122         state = GameState.FreeRoam;
123     }
124
125     private void Update()
126     {
127         if(state == GameState.FreeRoam)
128         {
129             player.HandleUpdate();
130         }
131         else if(state == GameState.PatientQuiz)
```

```
132         {
133             PatientQuizManager.Instance.HandleUpdate();
134         }
135         else if(state == GameState.ChestQuiz)
136         {
137             ChestQuizManager.Instance.HandleUpdate();
138         }
139         else if(state == GameState.Dialogue)
140         {
141             DialogueManager.Instance.HandleUpdate();
142         }
143         else if(state == GameState.GameOver)
144         {
145             if(SimpleInput.GetButton("Retry"))
146             {
147                 AudioManager.i.PlaySfx(AudioId.UIButton);
148                 SceneManager.LoadScene(player.CurrentChapter);
149                 missionBox.SetActive(true);
150                 state = GameState.FreeRoam;
151             }
152             else if(SimpleInput.GetButton("MainMenu"))
153             {
154                 AudioManager.i.PlaySfx(AudioId.UIButton);
155                 SceneManager.LoadScene(0);
156             }
157         }
158         else if(state == GameState.GameClear)
159         {
160             if(SimpleInput.GetButton("MainMenu"))
161             {
162                 AudioManager.i.PlaySfx(AudioId.UIButton);
163                 SceneManager.LoadScene(0);
164             }
165             else if(SimpleInput.GetButton("Continue"))
166             {
167                 AudioManager.i.PlaySfx(AudioId.UIButton);
168                 gameClearScreen.SetActive(false);
169                 playerControls.SetActive(true);
170                 missionBox.SetActive(true);
171                 state = GameState.FreeRoam;
172             }
173         }
174 
175         // Pause Controls
176         if(SimpleInput.GetButton("Pause"))
177         {
178             AudioManager.i.PlaySfx(AudioId.UIButton);
179             AudioManager.i.StopMusic();
180             gamePauseScreen.SetActive(true);
181             missionBox.SetActive(false);
182             playerControls.SetActive(false);
183             heart.SetActive(false);
184             PauseGame(true);
185         }
186         if(SimpleInput.GetButton("Resume"))
187         {
```

```
188                 AudioManager.i.PlaySfx(AudioId.UIButton);
189                 AudioManager.i.PlayMusic(explorationMusic);
190                 gamePauseScreen.SetActive(false);
191                 missionBox.SetActive(true);
192                 playerControls.SetActive(true);
193                 heart.SetActive(true);
194                 Controls.SetActive(false);
195                 PauseGame(false);
196             }
197             if(SimpleInput.GetButton("MainMenu"))
198             {
199                 AudioManager.i.PlaySfx(AudioId.UIButton);
200                 SceneManager.LoadScene(0);
201             }
202             if(SimpleInput.GetButton("Controls"))
203             {
204                 AudioManager.i.PlaySfx(AudioId.UIButton);
205                 Controls.SetActive(true);
206                 gamePauseScreen.SetActive(false);
207             }
208             if(SimpleInput.GetButton("closeInfo"))
209             {
210                 AudioManager.i.PlaySfx(AudioId.UIButton);
211                 gamePauseScreen.SetActive(false);
212                 Controls.SetActive(false);
213                 missionBox.SetActive(true);
214                 playerControls.SetActive(true);
215                 heart.SetActive(true);
216                 PauseGame(false);
217             }
218         }
219     }
```

E.7 Cutscene Manager

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4   using UnityEngine.UI;
5   using UnityEngine.SceneManagement;
6
7   public class CutsceneManager : MonoBehaviour
8   {
9       #region UI References
10
11      [SerializeField] GameObject dialogueBox;
12      [SerializeField] Text nameText;
13      [SerializeField] Text dialogueText;
14      [SerializeField] GameObject[] optionsBox;
15      [SerializeField] Text[] optionsText;
16      [SerializeField] AudioClip backgroundMusic;
17
18      #endregion
19
20      public static CutsceneManager Instance {get; private set;}
21
```

```
22      private Dialogue[] dialogue;
23      private const int lettersPerSecond = 100;
24      private int currentLine = 0;
25      private bool isTyping;
26
27      private void Awake()
28      {
29          Instance = this;
30      }
31
32      private void Start()
33      {
34          AudioManager.i.PlayMusic(backgroundMusic);
35      }
36
37      public void Update()
38      {
39          if(!isTyping)
40          {
41              if(SimpleInput.GetButton("Option1") || SimpleInput.GetButton("Option2") || Si
42              {
43                  AudioManager.i.PlaySfx(AudioId.Trigger);
44                  NextLine();
45              }
46          }
47      }
48
49      public IEnumerator StartDialogue(Dialogue[] dialogue)
50      {
51          yield return new WaitForEndOfFrame();
52
53          this.dialogue = dialogue;
54
55          dialogueBox.SetActive(true);
56
57          AudioManager.i.PlaySfx(AudioId.Trigger);
58          StartCoroutine(Type(dialogue[currentLine].Line, dialogue[currentLine].Name));
59      }
60
61      public IEnumerator Type(string line, string name)
62      {
63          HideOptions();
64          isTyping = true;
65
66          nameText.text = name;
67          dialogueText.text = "";
68
69          foreach(var letter in line.ToCharArray())
70          {
71              dialogueText.text += letter;
72              yield return new WaitForSeconds(1f / lettersPerSecond);
73          }
74
75          isTyping = false;
76          ShowOptions(dialogue[currentLine].Options);
77      }
78
79      private void NextLine()
80      {
```

```
81          ++currentLine;
82          if(currentLine < dialogue.Length)
83          {
84              AudioManager.i.PlaySfx(AudioId.Trigger);
85              StartCoroutine(Type(dialogue[currentLine].Line, dialogue[currentLine].Name));
86          }
87          else
88          {
89              dialogueText.text = "";
90              currentLine = 0;
91              dialogueBox.SetActive(false);
92              SceneManager.LoadScene(PlayerPrefs.GetInt("CurrentChapter"));
93          }
94      }
95
96      private void ShowOptions(string[] options)
97      {
98          for(int i = 0; i < options.Length; i++)
99          {
100             optionsBox[i].SetActive(true);
101             optionsText[i].text = options[i];
102         }
103     }
104
105     private void HideOptions()
106     {
107         for(int i = 0; i < optionsBox.Length; i++)
108         {
109             optionsBox[i].SetActive(false);
110         }
111     }
112 }
```

E.8 Chest Quiz Manager

```
1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4  using UnityEngine;
5  using UnityEngine.UI;
6
7  public class ChestQuizManager : MonoBehaviour
8  {
9      [SerializeField] GameObject dialogueBox;
10     [SerializeField] Text nameText;
11     [SerializeField] Text dialogueText;
12     [SerializeField] GameObject[] optionsBox;
13     [SerializeField] Text[] optionsText;
14     [SerializeField] GameObject playerControls;
15
16     private Dialogue[] answer;
17     private Quiz[] quiz;
18     private const int lettersPerSecond = 100;
19     private int currentLine = 0;
20     private bool isTyping;
21
```

```
22      PlayerController player;
23
24      public event Action OnChestInteraction;
25      public event Action OnFinishQuiz;
26
27      public static ChestQuizManager Instance {get; private set;}
28
29      private void Awake()
30      {
31          Instance = this;
32      }
33
34      public void HandleUpdate()
35      {
36          if(!isTyping)
37          {
38              if(SimpleInput.GetButton("Option1"))
39              {
40                  ChestQuizManager.Instance.CheckAnswer(0);
41              }
42              if(SimpleInput.GetButton("Option2"))
43              {
44                  ChestQuizManager.Instance.CheckAnswer(1);
45              }
46              if(SimpleInput.GetButton("Option3"))
47              {
48                  ChestQuizManager.Instance.CheckAnswer(2);
49              }
50          }
51
52      }
53
54      public IEnumerator StartQuiz(PlayerController player, Quiz[] quiz, Dialogue[] answer)
55      {
56          yield return new WaitForEndOfFrame();
57
58          Missions.Instance.HideMission();
59
60          this.player = player;
61          this.quiz = quiz;
62          this.answer = answer;
63          OnChestInteraction?.Invoke();
64
65          Logo.Instance.SetLogo(quiz[currentLine].Name);
66
67          dialogueBox.SetActive(true);
68          playerControls.SetActive(false);
69
70          AudioManager.i.PlaySfx(AudioId.Trigger);
71          StartCoroutine(TypeQuiz(quiz[currentLine].Line));
72      }
73
74      public IEnumerator TypeQuiz(string question)
75      {
76          HideOptions();
77          isTyping = true;
78
79          nameText.text = quiz[currentLine].Name;
```

```
80          dialogueText.text = "";
81
82          foreach(var letter in question.ToCharArray())
83          {
84              dialogueText.text += letter;
85              yield return new WaitForSeconds(1f / lettersPerSecond);
86          }
87
88          isTyping = false;
89          ShowOptions(quiz[currentLine].Options);
90      }
91
92      public void NextLine()
93      {
94          ++currentLine;
95          if(currentLine < quiz.Length)
96          {
97              AudioManager.i.PlaySfx(AudioId.Trigger);
98              StartCoroutine(TypeQuiz(quiz[currentLine].Line));
99          }
100         else
101         {
102             dialogueText.text = "";
103             currentLine = 0;
104             dialogueBox.SetActive(false);
105             playerControls.SetActive(true);
106             Missions.Instance.ShowMission();
107
108             OnFinishQuiz?.Invoke();
109             StartCoroutine(DialogueManager.Instance.StartDialogue(answer));
110         }
111     }
112
113     private void ShowOptions(string[] options)
114     {
115         for(int i = 0; i < options.Length; i++)
116         {
117             optionsBox[i].SetActive(true);
118             optionsText[i].text = options[i];
119         }
120     }
121
122     private void HideOptions()
123     {
124         for(int i = 0; i < optionsBox.Length; i++)
125         {
126             optionsBox[i].SetActive(false);
127         }
128     }
129
130     public void CheckAnswer(int answer)
131     {
132         optionsBox[answer].SetActive(false);
133         // check if answer is correct
134         if(answer == quiz[currentLine].Answer)
135         {
136             AudioManager.i.PlaySfx(AudioId.CorrectAnswer);
137             player.AddHeart();
```

```
138             NextLine();
139         }
140         else
141         {
142             AudioManager.i.PlaySfx(AudioId.WrongAnswer);
143             NextLine();
144         }
145     }
146 }
```

E.9 Patient Quiz Manager

```
1  using System;
2  using System.Collections;
3  using System.Collections.Generic;
4  using UnityEngine;
5  using UnityEngine.UI;
6
7  public class PatientQuizManager : MonoBehaviour
8  {
9      /*"(I don't think " + quiz[currentLine].Answer + "is the correct answer. Let me think
10
11     [SerializeField] GameObject dialogueBox;
12     [SerializeField] GameObject logo;
13     [SerializeField] Text nameText;
14     [SerializeField] Text dialogueText;
15     [SerializeField] GameObject[] optionsBox;
16     [SerializeField] Text[] optionsText;
17     [SerializeField] GameObject playerControls;
18     [SerializeField] Sprite spriteAbel;
19
20     private Quiz[] quiz;
21     private const int lettersPerSecond = 100;
22     private int currentLine = 0;
23     private bool isTyping;
24     private string mission;
25
26     PlayerController player;
27
28     public event Action OnPatientInteraction;
29     public event Action OnFinishQuiz;
30     public event Action OnGameOver;
31     public event Action OnGameClear;
32
33     public static PatientQuizManager Instance {get; private set;}
34
35     private void Awake()
36     {
37         Instance = this;
38     }
39
40     public void HandleUpdate()
41     {
42         if(!isTyping)
43         {
44             if(SimpleInput.GetButton("Option1"))
45             {
```

```
46                 PatientQuizManager.Instance.CheckAnswer(0);
47             }
48             if(SimpleInput.GetButton("Option2"))
49             {
50                 PatientQuizManager.Instance.CheckAnswer(1);
51             }
52             if(SimpleInput.GetButton("Option3"))
53             {
54                 PatientQuizManager.Instance.CheckAnswer(2);
55             }
56         }
57     }
58 
59     public IEnumerator StartQuiz(PlayerController player, Quiz[] quiz, string mission)
60     {
61         yield return new WaitForEndOfFrame();
62 
63         Missions.Instance.HideMission();
64 
65         this.player = player;
66         this.quiz = quiz;
67         this.mission = mission;
68         player.CurrentMission = mission;
69         OnPatientInteraction?.Invoke();
70 
71         Debug.Log(currentLine);
72         Logo.Instance.SetLogo(quiz[currentLine].Name);
73         dialogueBox.SetActive(true);
74         playerControls.SetActive(false);
75 
76         AudioManager.i.PlaySfx(AudioId.Trigger);
77         StartCoroutine(TypeQuiz(quiz[currentLine].Line));
78     }
79 
80     public IEnumerator TypeQuiz(string question)
81     {
82         HideOptions();
83         isTyping = true;
84 
85         nameText.text = quiz[currentLine].Name;
86         dialogueText.text = "";
87 
88         foreach(var letter in question.ToCharArray())
89         {
90             dialogueText.text += letter;
91             yield return new WaitForSeconds(1f / lettersPerSecond);
92         }
93 
94         isTyping = false;
95         ShowOptions(quiz[currentLine].Options);
96     }
97 
98     public void NextLine()
99     {
100        ++currentLine;
101        if(currentLine < quiz.Length)
102        {
103            AudioManager.i.PlaySfx(AudioId.Trigger);
```

```
104             Logo.Instance.SetLogo(quiz[currentLine].Name);
105             StartCoroutine(TypeQuiz(quiz[currentLine].Line));
106         }
107         else
108         {
109             dialogueText.text = "";
110             currentLine = 0;
111             dialogueBox.SetActive(false);
112             playerControls.SetActive(true);
113             player.UpdateProgress();
114             Missions.Instance.NextMission(mission);
115 
116             if(player.Progress == 5)
117             {
118                 player.UpdateChapter();
119                 OnGameClear?.Invoke();
120             }
121             else
122                 OnFinishQuiz?.Invoke();
123         }
124     }
125 
126     private void ShowOptions(string[] options)
127     {
128         for(int i = 0; i < options.Length; i++)
129         {
130             optionsBox[i].SetActive(true);
131             optionsText[i].text = options[i];
132         }
133     }
134 
135     private void HideOptions()
136     {
137         for(int i = 0; i < optionsBox.Length; i++)
138         {
139             optionsBox[i].SetActive(false);
140         }
141     }
142 
143     public void CheckAnswer(int answer)
144     {
145         optionsBox[answer].SetActive(false);
146         // check if answer is correct
147         if(answer == quiz[currentLine].Answer)
148             NextLine();
149         else
150         {
151             player.LoseHeart();
152             if(nameText.text == "Mr. Mayor")
153             {
154                 NextLine();
155             }
156             if(player.CurrentHP == 0)
157             {
158                 AudioManager.i.StopMusic();
159                 AudioManager.i.PlaySfx(AudioId.GameOver);
160                 dialogueBox.SetActive(false);
```

```
                playerControls.SetActive(true);
                OnGameOver?.Invoke();
            }
        }
    }
}
```

E.10 Dialogue Manager

```
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class DialogueManager : MonoBehaviour
{
    #region UI References

    [SerializeField] GameObject dialogueBox;
    [SerializeField] GameObject nameBox;
    [SerializeField] Text nameText;
    [SerializeField] Text dialogueText;
    [SerializeField] GameObject[] optionsBox;
    [SerializeField] Text[] optionsText;
    [SerializeField] GameObject playerControls;

    #endregion

    public static DialogueManager Instance {get; private set;}

    private Dialogue[] dialogue;
    private const int lettersPerSecond = 100;
    private int currentLine = 0;
    private bool isTyping;

    public event Action OnStartDialogue;
    public event Action OnFinishDialogue;

    private void Awake()
    {
        Instance = this;
    }

    public void HandleUpdate()
    {
        if(!isTyping)
        {
            if(SimpleInput.GetButton("Option1") || SimpleInput.GetButton("Option2") || Si
            {
                AudioManager.i.PlaySfx(AudioId.Trigger);
                NextLine();
            }
        }
    }

    public IEnumerator StartDialogue(Dialogue[] dialogue)
```

```
49      {
50          yield return new WaitForEndOfFrame();
51          Missions.Instance.HideMission();
52
53          OnStartDialogue?.Invoke();
54
55          this.dialogue = dialogue;
56
57          Logo.Instance.SetLogo(dialogue[currentLine].Name);
58          dialogueBox.SetActive(true);
59          playerControls.SetActive(false);
60
61          AudioManager.i.PlaySfx(AudioId.Trigger);
62          StartCoroutine(Type(dialogue[currentLine].Line, dialogue[currentLine].Name));
63      }
64
65      public IEnumerator Type(string line, string name)
66      {
67          HideOptions();
68          isTyping = true;
69
70          nameText.text = name;
71          dialogueText.text = "";
72
73          foreach(var letter in line.ToCharArray())
74          {
75              dialogueText.text += letter;
76              yield return new WaitForSeconds(1f / lettersPerSecond);
77          }
78
79          isTyping = false;
80          ShowOptions(dialogue[currentLine].Options);
81      }
82
83      private void NextLine()
84      {
85          ++currentLine;
86          if(currentLine < dialogue.Length)
87          {
88              AudioManager.i.PlaySfx(AudioId.Trigger);
89              Logo.Instance.SetLogo(dialogue[currentLine].Name);
90              StartCoroutine(Type(dialogue[currentLine].Line, dialogue[currentLine].Name));
91          }
92          else
93          {
94              dialogueText.text = "";
95              currentLine = 0;
96              dialogueBox.SetActive(false);
97              playerControls.SetActive(true);
98              Missions.Instance.ShowMission();
99
100             OnFinishDialogue?.Invoke();
101         }
102     }
103
104     private void ShowOptions(string[] options)
105     {
106         for(int i = 0; i < options.Length; i++)
```

```
107         {
108             optionsBox[i].SetActive(true);
109             optionsText[i].text = options[i];
110         }
111     }
112 
113     private void HideOptions()
114     {
115         for(int i = 0; i < optionsBox.Length; i++)
116         {
117             optionsBox[i].SetActive(false);
118         }
119     }
120 }
```

E.11 Main Menu

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5  
6  public class MainMenu : MonoBehaviour
7  {
8      [SerializeField] GameObject title;
9      [SerializeField] GameObject information;
10     [SerializeField] GameObject beginButton;
11     [SerializeField] GameObject continueButton;
12     [SerializeField] GameObject chapterCodeButton;
13     [SerializeField] GameObject chapterCode;
14     [SerializeField] AudioClip MenuMusic;
15     [SerializeField] GameObject InfoButton;
16     [SerializeField] GameObject Controls;
17     [SerializeField] GameObject Credits;
18     /*
19     [SerializeField] GameObject continueButton;
20     [SerializeField] GameObject chaptersButton;
21     [SerializeField] GameObject chapterSelect;
22     [SerializeField] GameObject credits;
23     [SerializeField] GameObject creditScreen;
24     */
25     public Animator transition;
26     public float transitionTime = 1f;
27 
28     private void Start()
29     {
30         AudioManager.i.PlayMusic(MenuMusic);
31     }
32 
33     public void Update()
34     {
35         if(SimpleInput.GetButton("Begin"))
36         {
37             AudioManager.i.PlaySfx(AudioId.UIButton);
38             // If save file detected present a continue button
39             if(SavingSystem.i.CheckFile("test"))
```

```
40              {
41                  continueButton.SetActive(true);
42              }
43
44              beginButton.SetActive(false);
45              chapterCodeButton.SetActive(true);
46          }
47          else if(SimpleInput.GetButton("Code"))
48          {
49              AudioManager.i.PlaySfx(AudioId.UIButton);
50              chapterCodeButton.SetActive(false);
51              //continueButton.SetActive(false);
52              chapterCode.SetActive(true);
53          }
54          else if(SimpleInput.GetButton("Continue"))
55          {
56              AudioManager.i.PlaySfx(AudioId.UIButton);
57              SceneManager.LoadScene(PlayerPrefs.GetInt("CurrentChapter"));
58          }
59          else if(SimpleInput.GetButton("InformationButton"))
60          {
61              AudioManager.i.PlaySfx(AudioId.UIButton);
62              information.SetActive(true);
63              title.SetActive(false);
64              InfoButton.SetActive(false);
65          }
66          else if(SimpleInput.GetButton("closeInfo"))
67          {
68              AudioManager.i.PlaySfx(AudioId.UIButton);
69              information.SetActive(false);
70              title.SetActive(true);
71              InfoButton.SetActive(true);
72              Controls.SetActive(false);
73              Credits.SetActive(false);
74              //chapterCode.SetActive(false);
75          }
76          else if(SimpleInput.GetButton("closecode"))
77          {
78              AudioManager.i.PlaySfx(AudioId.UIButton);
79              chapterCode.SetActive(false);
80              chapterCodeButton.SetActive(true);
81          }
82          else if(SimpleInput.GetButton("Controls"))
83          {
84              AudioManager.i.PlaySfx(AudioId.UIButton);
85              information.SetActive(false);
86              Controls.SetActive(true);
87          }
88          else if(SimpleInput.GetButton("Credits"))
89          {
90              AudioManager.i.PlaySfx(AudioId.UIButton);
91              information.SetActive(false);
92              Credits.SetActive(true);
93          }
94      }
95
```

```
96      public void LoadNextLevel()
97      {
98          StartCoroutine(LoadLevel(SceneManager.GetActiveScene().buildIndex + 1));
99      }
100
101     IEnumerator LoadLevel(int levelIndex)
102     {
103         transition.SetTrigger("Start");
104
105         yield return new WaitForSeconds(transitionTime);
106
107         SceneManager.LoadScene(levelIndex);
108     }
109 }
```

E.12 Chapter Code Checker

```
1  using UnityEngine;
2  using UnityEngine.UI;
3  using UnityEngine.SceneManagement;
4
5  public class CheckPassword : MonoBehaviour
6  {
7      private InputField inputTextField;
8
9      void Start()
10     {
11         inputTextField = GetComponent<InputField>();
12     }
13
14     public void CheckPasswordAndLoadScene()
15     {
16         SavingSystem.i.Delete("test");
17         if(inputTextField.text == "bacteria111")
18         {
19             PlayerPrefs.SetInt("CurrentChapter", 1);
20             SceneManager.LoadScene("Cutscene");
21         }
22         else if(inputTextField.text == "virus212")
23         {
24             PlayerPrefs.SetInt("CurrentChapter", 2);
25             LoadChapter();
26         }
27         else if(inputTextField.text == "fungi313")
28         {
29             PlayerPrefs.SetInt("CurrentChapter", 3);
30             LoadChapter();
31         }
32         else if(inputTextField.text == "parasite414")
33         {
34             PlayerPrefs.SetInt("CurrentChapter", 4);
35             LoadChapter();
36         }
37     }
38
39     public void LoadChapter()
```

```
40      {
41          SceneManager.LoadScene(PlayerPrefs.GetInt("CurrentChapter", 1));
42      }
43  }
```

E.13 Player Controller

```
1   using System;
2   using System.Collections;
3   using System.Collections.Generic;
4   using UnityEngine;
5
6   public class PlayerController : MonoBehaviour, ISavable
7   {
8       [SerializeField] LayerMask objectLayer;
9       [SerializeField] LayerMask interactableLayer;
10      [SerializeField] LayerMask portalLayer;
11      [SerializeField] LayerMask fovLayer;
12      [SerializeField] LayerMask instructionLayer;
13      [SerializeField] LayerMask monologueLayer;
14      [SerializeField] GameObject[] hearts;
15
16      const int maxHP = 5;
17      Vector3 startingPosition;
18      public int currentHP;
19      public int currentChapter = 1;
20      public int progress;
21      public string currentMission;
22
23      public string CurrentMission {
24          get => currentMission;
25          set => currentMission = value;
26      }
27
28      public Vector3 StartingPosition {
29          get => startingPosition;
30      }
31
32      public int CurrentHP {
33          get => currentHP;
34          set => currentHP = value;
35      }
36
37      public int CurrentChapter {
38          get => currentChapter;
39          set => currentChapter = value;
40      }
41
42      public int Progress {
43          get {return progress;}
44      }
45
46      private float moveSpeed;
47      private bool isMoving;
48      private Vector2 input;
49
50      public bool IsMoving{
51          set => isMoving = value;
```

```
52      }
53
54      public Animator animator;
55      private SpriteRenderer spriteRenderer;
56
57      public void Awake()
58      {
59          animator = GetComponent<Animator>();
60          spriteRenderer = GetComponent<SpriteRenderer>();
61          startingPosition = transform.position;
62          currentHP = maxHP;
63      }
64
65      public void HandleUpdate()
66      {
67          if(!isMoving)
68          {
69              CheckForMonologue();
70              if(SimpleInput.GetButton("Run"))
71              {
72                  moveSpeed = 10;
73              }
74              else
75              {
76                  moveSpeed = 5;
77              }
78
79              input.x = Mathf.RoundToInt(SimpleInput.GetAxisRaw("Horizontal"));
80              input.y = Mathf.RoundToInt(SimpleInput.GetAxisRaw("Vertical"));
81
82              // to remove player diagonal movements
83              if(input.x != 0) input.y = 0;
84
85              if(input != Vector2.zero)
86              {
87                  StartCoroutine(Move(input));
88              }
89
90              animator.SetBool("isMoving", isMoving);
91
92              if(SimpleInput.GetButton("Action"))
93              {
94                  Interact();
95              }
96          }
97      }
98
99      // Handles Player Movement
100     private IEnumerator Move(Vector2 moveVec)
101     {
102         animator.SetFloat("moveX", moveVec.x);
103         animator.SetFloat("moveY", moveVec.y);
104
105         var targetPos = transform.position;
106         targetPos.x += moveVec.x;
107         targetPos.y += moveVec.y;
108
109         // flip sprite when moving left or right side
```

```
110         if(input.x < 0)
111         {
112             spriteRenderer.flipX = true;
113         }
114         else if(input.x > 0)
115         {
116             spriteRenderer.flipX = false;
117         }
118 
119         if(!IsWalkable(targetPos))
120         {
121             yield break;
122         }
123 
124         isMoving = true;
125         animator.SetBool("isMoving", isMoving);
126 
127         // check if player current position reached target postition
128         while((targetPos - transform.position).sqrMagnitude > Mathf.Epsilon)
129         {
130             transform.position = Vector3.MoveTowards(transform.position, targetPos, moveS
131             yield return null;
132         }
133 
134         transform.position = targetPos;
135 
136         isMoving = false;
137 
138         CheckForPortal();
139         CheckForFov();
140         CheckForInstruction();
141     }
142 
143     private bool IsWalkable(Vector3 targetPos)
144     {
145         if(Physics2D.OverlapCircle(targetPos, 0.1f, objectLayer | interactableLayer) != n
146         {
147             return false;
148         }
149 
150         return true;
151     }
152 
153     private void CheckForPortal()
154     {
155         var collider = Physics2D.OverlapCircle(transform.position, 0, portalLayer);
156 
157         if(collider != null)
158         {
159             collider.GetComponent<IPlayerTriggerable>()?.OnPlayerTrigger(this);
160         }
161     }
162 
163     private void CheckForFov()
164     {
165         var collider = Physics2D.OverlapCircle(transform.position, 0, fovLayer);
166 
167         if(collider != null)
```

```
168         {
169             isMoving = false;
170             animator.SetBool("isMoving", isMoving);
171             collider.GetComponentInParent<IPlayerTriggerable>()?.OnPlayerTrigger(this);
172         }
173     }
174 
175     public void MoveToNPC(Vector3 collider)
176     {
177         var targetPos = collider - transform.position;
178         var moveVec = targetPos - targetPos.normalized;
179 
180         StartCoroutine(Move(moveVec));
181     }
182 
183     private void CheckForInstruction()
184     {
185         var collider = Physics2D.OverlapCircle(transform.position, 0, instructionLayer);
186 
187         if(collider != null)
188         {
189             isMoving = false;
190             animator.SetBool("isMoving", isMoving);
191             collider.GetComponent<IPlayerTriggerable>()?.OnPlayerTrigger(this);
192         }
193     }
194 
195     private void CheckForMonologue()
196     {
197         var collider = Physics2D.OverlapCircle(transform.position, 0, monologueLayer);
198 
199         if(collider != null)
200         {
201             isMoving = false;
202             animator.SetBool("isMoving", isMoving);
203             collider.GetComponent<IPlayerTriggerable>()?.OnPlayerTrigger(this);
204         }
205     }
206 
207     public void Interact()
208     {
209         var faceDir = new Vector3(animator.GetFloat("moveX"), animator.GetFloat("moveY"))
210         var interactPosition = transform.position + faceDir;
211 
212         var collider = Physics2D.OverlapCircle(interactPosition, 0.1f, interactableLayer)
213 
214         if(collider != null)
215         {
216             collider.GetComponent<Interactable>()?.Interact(this);
217         }
218     }
219 
220     public void ShowHP()
221     {
222         for(int i = 0; i < currentHP; i++)
223         {
224             hearts[i].SetActive(true);
225         }
```

```
226     }
227
228     public void LoseHeart()
229     {
230         currentHP--;
231         hearts[currentHP].SetActive(false);
232     }
233
234     public void AddHeart()
235     {
236         if(currentHP < maxHP)
237         {
238             hearts[currentHP].SetActive(true);
239             currentHP++;
240         }
241     }
242
243     public void UpdateProgress()
244     {
245         if(progress < 5)
246         {
247             progress++;
248         }
249     }
250
251     public void UpdateChapter()
252     {
253         if(currentChapter < 4)
254         {
255             currentChapter++;
256         }
257         if(currentChapter > PlayerPrefs.GetInt("Chapter", 1))
258         {
259             PlayerPrefs.SetInt("Chapter", currentChapter);
260         }
261     }
262
263     public void SetCurrentChapter(int x)
264     {
265         currentChapter = x;
266     }
267
268     #region Saving PlayerData
269     public object CaptureState()
270     {
271         var saveData = new PlayerData()
272         {
273             currentMission = this.currentMission,
274             currentHP = this.currentHP,
275             currentChapter = this.currentChapter,
276             progress = this.progress,
277             position = new float[] {transform.position.x, transform.position.y}
278         };
279
280         return saveData;
281     }
282
283     public void RestoreState(object state)
```

```
284     {
285         var saveData = (PlayerData)state;
286         var pos = saveData.position;
287 
288         currentMission = saveData.currentMission;
289         currentHP = saveData.currentHP;
290         currentChapter = saveData.currentChapter;
291         progress = saveData.progress;
292         transform.position = new Vector3(pos[0], pos[1]);
293 
294     }
295 
296     #endregion
297 }
298 
299 [System.Serializable]
300 public class PlayerData
301 {
302     public string currentMission;
303     public int currentHP;
304     public int currentChapter;
305     public int progress;
306     public float[] position;
307 }
```

E.14 Monologue Controller

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5 
6  public class MonologueController : MonoBehaviour, IPlayerTriggerable, ISavable
7  {
8      [SerializeField] Dialogue[] dialogue;              // Delete after transferring conte
9      [SerializeField] Quiz[] quiz;
10     [TextArea][SerializeField] string mission;
11     bool isTriggered;
12 
13     PlayerController player;
14 
15     public IEnumerator TriggerDialogue()
16     {
17         GameController.Instance.PauseGame(true);
18         yield return new WaitForSeconds(0.1f);
19         GameController.Instance.PauseGame(false);
20 
21         StartCoroutine(PatientQuizManager.Instance.StartQuiz(player, quiz, mission));
22     }
23     public void OnPlayerTrigger(PlayerController player)
24     {
25         this.player = player;
26 
27         if(!isTriggered)
28         {
29             StartCoroutine(TriggerDialogue());
30             isTriggered = true;
31         }
```

```
32      }
33
34      public object CaptureState()
35      {
36          return isTriggered;
37      }
38
39      public void RestoreState(object state)
40      {
41          isTriggered = (bool)state;
42      }
43  }
```

E.15 Patient Controller

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class PatientController : MonoBehaviour, Interactable, ISavable
6   {
7       [SerializeField] Dialogue[] dialogue; // dialogue to be executed when patient prior is
8       [SerializeField] Dialogue[] afterDialogue;  // dialogue to be executed when this pati
9       [SerializeField] Quiz[] quiz; // set of questions npc asks the player, to be executed
10      [SerializeField] int order; // order of npc in storyline of the chapter
11      [TextArea][SerializeField] string mission;
12      bool isTriggered;
13
14      public void Interact(PlayerController player)
15      {
16          if(player.Progress < order) // means player should find npc[order-1] to interact
17          {
18              StartCoroutine(DialogueManager.Instance.StartDialogue(dialogue));
19          }
20          else
21          {
22              if(!isTriggered) // means first interaction
23              {
24                  StartCoroutine(PatientQuizManager.Instance.StartQuiz(player, quiz, missio
25                  isTriggered = true; // marks that player already interacted with this pat
26              }
27              else
28              {
29                  StartCoroutine(DialogueManager.Instance.StartDialogue(afterDialogue));
30              }
31          }
32      }
33
34      public object CaptureState()
35      {
36          return isTriggered;
37      }
38
39      public void RestoreState(object state)
40      {
41          isTriggered = (bool)state;
```

```
    }
}
```

E.16 Information Controller

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class InformationController : MonoBehaviour, Interactable, ISavable
{
    [SerializeField] Dialogue[] dialogue;
    [SerializeField] Dialogue[] afterDialogue;
    bool isTriggered;

    public void Interact(PlayerController player)
    {
        if(!isTriggered) // means first time palyer interacts with current chest
        {
            StartCoroutine(DialogueManager.Instance.StartDialogue(dialogue));
            isTriggered = true; // marks that the player already interacted with this npc
        }
        else
        {
            StartCoroutine(DialogueManager.Instance.StartDialogue(afterDialogue));
        }
    }

    public object CaptureState()
    {
        return isTriggered;
    }

    public void RestoreState(object state)
    {
        isTriggered = (bool)state;
    }
}
```

E.17 Field Of View

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Fov : MonoBehaviour, IPlayerTriggerable, ISavable
{
    [SerializeField] GameObject alertImage;
    [SerializeField] Quiz[] dialogue;
    PlayerController player;
    bool isTriggered;
    [TextArea][SerializeField] string mission;

    public IEnumerator TriggerDialogue()
```

```
15      {
16          GameController.Instance.PauseGame(true);
17
18          alertImage.SetActive(true);
19          yield return new WaitForSeconds(0.5f);
20
21          player.IsMoving = true;
22          player.MoveToNPC(transform.position);
23          alertImage.SetActive(false);
24
25          GameController.Instance.PauseGame(false);
26
27          yield return new WaitForSeconds(0.5f);
28          StartCoroutine(PatientQuizManager.Instance.StartQuiz(player, dialogue, mission));
29      }
30
31      public void OnPlayerTrigger(PlayerController player)
32      {
33          this.player = player;
34          if(!isTriggered)
35          {
36              StartCoroutine(TriggerDialogue());
37              isTriggered = true;
38          }
39      }
40
41      public object CaptureState()
42      {
43          return isTriggered;
44      }
45
46      public void RestoreState(object state)
47      {
48          isTriggered = (bool)state;
49      }
50  }
```

E.18 InstructionController

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4   using UnityEngine.UI;
5
6   public class InstructionController : MonoBehaviour, IPlayerTriggerable, ISavable
7   {
8       [SerializeField] GameObject instructionImage;
9       [SerializeField] Dialogue[] dialogue;               // Delete After transferring to Q
10      [SerializeField] Quiz[] quiz;
11      [SerializeField] GameObject shadow;
12      PlayerController player;
13      bool isTriggered;
14      [TextArea][SerializeField] string mission;
15
16      public void OnPlayerTrigger(PlayerController player)
17      {
18          this.player = player;
```

```
19          if(!isTriggered)
20          {
21              StartCoroutine(TriggerDialogue());
22          }
23      }
24
25      public IEnumerator TriggerDialogue()
26      {
27          GameController.Instance.PauseGame(true);
28          yield return new WaitForSeconds(0.1f);
29          GameController.Instance.PauseGame(false);
30
31          StartCoroutine(PatientQuizManager.Instance.StartQuiz(player, quiz, mission));
32          // StartCoroutine(DialogueManager.Instance.StartDialogue(dialogue));
33          // player.UpdateProgress();
34          isTriggered = true;
35      }
36
37      private void Update()
38      {
39          if(isTriggered)
40          {
41              instructionImage.SetActive(false);
42              shadow.SetActive(false);
43          }
44      }
45      public object CaptureState()
46      {
47          return isTriggered;
48      }
49
50      public void RestoreState(object state)
51      {
52          isTriggered = (bool)state;
53      }
54  }
```

E.19 Chest Controller

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class ChestController : MonoBehaviour, Interactable, ISavable
6   {
7       [SerializeField] Quiz[] quiz;
8       [SerializeField] Dialogue[] answer;
9       bool isTriggered;
10
11      Animator animator;
12
13      private void Awake()
14      {
15          animator = GetComponent<Animator>();
16      }
17
```

```
18      void Update()
19      {
20          animator.SetBool("start", isTriggered);
21      }
22
23      public void Interact(PlayerController player)
24      {
25          if(!isTriggered) // means first time palyer interacts with current chest
26          {
27              isTriggered = true; // marks that the player already interacted with this ches
28              animator.SetBool("isTriggered", isTriggered);
29              StartCoroutine(ChestQuizManager.Instance.StartQuiz(player, quiz, answer)); //
30          }
31          else
32          {
33              StartCoroutine(DialogueManager.Instance.StartDialogue(answer)); // on second
34          }
35      }
36
37      public object CaptureState()
38      {
39          return isTriggered;
40      }
41
42      public void RestoreState(object state)
43      {
44          isTriggered = (bool)state;
45      }
46  }
```

E.20 Door

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class Door : MonoBehaviour
6   {
7       [SerializeField] GameObject openDoor;
8       [SerializeField] GameObject closeDoor;
9
10      void OnTriggerEnter2D(Collider2D col)
11      {
12          if(col.gameObject.tag == "Player")
13          {
14              openDoor.SetActive(false);
15              closeDoor.SetActive(true);
16          }
17      }
18
19      void OnTriggerExit2D(Collider2D col)
20      {
21          if(col.gameObject.tag == "Player")
22          {
23              openDoor.SetActive(true);
24              closeDoor.SetActive(false);
```

```
25          }
26      }
27  }
```

E.21 Player

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class Player : MonoBehaviour
6   {
7       private void Awake()
8       {
9           DontDestroyOnLoad(gameObject);
10      }
11  }
```

E.22 PlayerLoader

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4
5   public class PlayerLoader : MonoBehaviour
6   {
7       [SerializeField] GameObject playerPrefab;
8
9       private void Awake()
10      {
11          var existingObjects = FindObjectsOfType<Player>();
12          if(existingObjects.Length == 0)
13          {
14              Instantiate(playerPrefab, new Vector3(0,0,0), Quaternion.identity);
15          }
16      }
17  }
```

E.23 Saving System

```
1   using System;
2   using System.Collections;
3   using System.Collections.Generic;
4   using System.IO;
5   using System.Runtime.Serialization.Formatters.Binary;
6   using UnityEngine;
7   using UnityEngine.SceneManagement;
8
9   public class SavingSystem : MonoBehaviour
10  {
11      public static SavingSystem i { get; private set; }
12      private void Awake()
13      {
```

```
14          i = this;
15      }
16
17      Dictionary<string, object> gameState = new Dictionary<string, object>();
18
19      public void CaptureEntityStates(List<SavableEntity> savableEntities)
20      {
21          foreach (SavableEntity savable in savableEntities)
22          {
23              gameState[savable.UniqueId] = savable.CaptureState();
24          }
25      }
26
27      public void RestoreEntityStates(List<SavableEntity> savableEntities)
28      {
29          foreach (SavableEntity savable in savableEntities)
30          {
31              string id = savable.UniqueId;
32              if (gameState.ContainsKey(id))
33                  savable.RestoreState(gameState[id]);
34          }
35      }
36
37      public void Save(string saveFile)
38      {
39          CaptureState(gameState);
40          SaveFile(saveFile, gameState);
41      }
42
43      public void Load(string saveFile)
44      {
45          gameState = LoadFile(saveFile);
46          RestoreState(gameState);
47      }
48
49      public void Delete(string saveFile)
50      {
51          File.Delete(GetPath(saveFile));
52      }
53
54      public bool CheckFile(string saveFile)
55      {
56          string path = GetPath(saveFile);
57          if(File.Exists(path))
58              return true;
59          else
60              return false;
61      }
62
63      // Used to capture states of all savable objects in the game
64      private void CaptureState(Dictionary<string, object> state)
65      {
66          foreach (SavableEntity savable in FindObjectsOfType<SavableEntity>())
67          {
68              state[savable.UniqueId] = savable.CaptureState();
69          }
70      }
71
```

```
72      // Used to restore states of all savable objects in the game
73      private void RestoreState(Dictionary<string, object> state)
74      {
75          foreach (SavableEntity savable in FindObjectsOfType<SavableEntity>())
76          {
77              string id = savable.UniqueId;
78              if (state.ContainsKey(id))
79                  savable.RestoreState(state[id]);
80          }
81      }
82
83      void SaveFile(string saveFile, Dictionary<string, object> state)
84      {
85          string path = GetPath(saveFile);
86          print($"saving to {path}");
87
88          using (FileStream fs = File.Open(path, FileMode.Create))
89          {
90              // Serialize our object
91              BinaryFormatter binaryFormatter = new BinaryFormatter();
92              binaryFormatter.Serialize(fs, state);
93          }
94      }
95
96      Dictionary<string, object> LoadFile(string saveFile)
97      {
98          string path = GetPath(saveFile);
99          if (!File.Exists(path))
100             return new Dictionary<string, object>();
101
102         using (FileStream fs = File.Open(path, FileMode.Open))
103         {
104             // Deserialize our object
105             BinaryFormatter binaryFormatter = new BinaryFormatter();
106             return (Dictionary<string, object>)binaryFormatter.Deserialize(fs);
107         }
108     }
109
110     private string GetPath(string saveFile)
111     {
112         return Path.Combine(Application.persistentDataPath, saveFile);
113     }
114 }
```

E.24 ISavable

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  public interface ISavable
8  {
9      object CaptureState();
10     void RestoreState(object state);
```

```
}
```

E.25 Savable Entity

```
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEditor;
using UnityEngine;

[ExecuteAlways]
public class SavableEntity : MonoBehaviour
{
    [SerializeField] string uniqueId = "";
    static Dictionary<string, SavableEntity> globalLookup = new Dictionary<string, Savable

    public string UniqueId => uniqueId;

    // Used to capture state of the gameobject on which the savableEntity is attached
    public object CaptureState()
    {
        Dictionary<string, object> state = new Dictionary<string, object>();
        foreach (ISavable savable in GetComponents<ISavable>())
        {
            state[savable.GetType().ToString()] = savable.CaptureState();
        }
        return state;
    }

    // Used to restore state of the gameobject on which the savableEntity is attached
    public void RestoreState(object state)
    {
        Dictionary<string, object> stateDict = (Dictionary<string, object>)state;
        foreach (ISavable savable in GetComponents<ISavable>())
        {
            string id = savable.GetType().ToString();

            if (stateDict.ContainsKey(id))
                savable.RestoreState(stateDict[id]);
        }
    }

#if UNITY_EDITOR
    // Update method used for generating UUID of the SavableEntity
    private void Update()
    {
        // don't execute in playmode
        if (Application.IsPlaying(gameObject)) return;

        // don't generate Id for prefabs (prefab scene will have path as null)
        if (String.IsNullOrEmpty(gameObject.scene.path)) return;

        SerializedObject serializedObject = new SerializedObject(this);
        SerializedProperty property = serializedObject.FindProperty("uniqueId");

        if (String.IsNullOrEmpty(property.stringValue) || !IsUnique(property.stringValue))
```

```
53          {
54              property.stringValue = Guid.NewGuid().ToString();
55              serializedObject.ApplyModifiedProperties();
56          }
57
58          globalLookup[property.stringValue] = this;
59      }
60  #endif
61
62      private bool IsUnique(string candidate)
63      {
64          if (!globalLookup.ContainsKey(candidate)) return true;
65
66          if (globalLookup[candidate] == this) return true;
67
68          // Handle scene unloading cases
69          if (globalLookup[candidate] == null)
70          {
71              globalLookup.Remove(candidate);
72              return true;
73          }
74
75          // Handle edge cases like designer manually changing the UUID
76          if (globalLookup[candidate].UniqueId != candidate)
77          {
78              globalLookup.Remove(candidate);
79              return true;
80          }
81
82          return false;
83      }
84  }
```

E.26 Logo

```
1   using System.Collections;
2   using System.Collections.Generic;
3   using UnityEngine;
4   using UnityEngine.UI;
5
6   public class Logo : MonoBehaviour
7   {
8       [SerializeField] GameObject logoBox;
9       [SerializeField] Sprite spriteAbel;
10      [SerializeField] Sprite spriteMayor;
11      [SerializeField] Sprite spriteArchitect;
12      [SerializeField] Sprite spriteEngineer;
13      [SerializeField] Sprite spriteForeman;
14      [SerializeField] Sprite spriteElectrician;
15      [SerializeField] Sprite spritePlumber;
16      [SerializeField] Sprite spriteCleaner;
17      [SerializeField] Sprite spriteGardener;
18      [SerializeField] Sprite spritePainter;
19      [SerializeField] Sprite spriteDesigner;
20      [SerializeField] Sprite spriteGuard;
21      [SerializeField] Sprite spriteClerk;
```

```
22      [SerializeField] Sprite spriteNurse;
23      [SerializeField] Sprite spriteGeneral;
24      [SerializeField] Sprite spriteCombo;
25      [SerializeField] Sprite spriteChestClosed;
26      [SerializeField] Sprite spriteChestOpen;
27      [SerializeField] Sprite spriteSign;
28      public static Logo Instance {get; private set;}
29
30      private void Awake()
31      {
32          Instance = this;
33      }
34
35      public void SetLogo(string name)
36      {
37          if(name == "Doctor Abel" || name == "Doctor Abel ")
38          {
39              logoBox.GetComponent<Image>().sprite = spriteAbel;
40          }
41          else if(name == "Mr. Mayor" || name == "Mr. Mayor ")
42          {
43              logoBox.GetComponent<Image>().sprite = spriteMayor;
44          }
45          else if(name == "Architect JC" || name == "Architect JC ")
46          {
47              logoBox.GetComponent<Image>().sprite = spriteArchitect;
48          }
49          else if(name == "Engineer John" || name == "Engineer John ")
50          {
51              logoBox.GetComponent<Image>().sprite = spriteEngineer;
52          }
53          else if(name == "Foreman Carlo" || name == "Foreman Carlo ")
54          {
55              logoBox.GetComponent<Image>().sprite = spriteForeman;
56          }
57          else if(name == "Electrician Kaloy" || name == "Electrician Kaloy ")
58          {
59              logoBox.GetComponent<Image>().sprite = spriteElectrician;
60          }
61          else if(name == "Plumber Juan" || name == "Plumber Juan ")
62          {
63              logoBox.GetComponent<Image>().sprite = spritePlumber;
64          }
65          else if(name == "Cleaner Sef" || name == "Cleaner Sef ")
66          {
67              logoBox.GetComponent<Image>().sprite = spriteCleaner;
68          }
69          else if(name == "Gardener Fred" || name == "Gardener Fred ")
70          {
71              logoBox.GetComponent<Image>().sprite = spriteGardener;
72          }
73          else if(name == "Painter Daisy" || name == "Painter Daisy ")
74          {
75              logoBox.GetComponent<Image>().sprite = spritePainter;
76          }
77          else if(name == "Designer Jessie" || name == "Designer Jessie")
```

```
78          {
79              logoBox.GetComponent<Image>().sprite = spriteDesigner;
80          }
81          else if(name == "Guard Magnus" || name == "Guard Magnus ")
82          {
83              logoBox.GetComponent<Image>().sprite = spriteGuard;
84          }
85          else if(name == "Clerk Mariane" || name == "Clerk Mariane ")
86          {
87              logoBox.GetComponent<Image>().sprite = spriteClerk;
88          }
89          else if(name == "Nurse Joy" || name == "Nurse Joy ")
90          {
91              logoBox.GetComponent<Image>().sprite = spriteNurse;
92          }
93          else if(name == "JC, John and Carlo")
94          {
95              logoBox.GetComponent<Image>().sprite = spriteCombo;
96          }
97          else if(name == "Chest" || name == "Quiz")
98          {
99              logoBox.GetComponent<Image>().sprite = spriteChestClosed;
100         }
101         else if(name == "Answer" || name == "Answer ")
102         {
103             logoBox.GetComponent<Image>().sprite = spriteChestOpen;
104         }
105         else if(name == "Aguilar's Residence" || name == "Monument" || name == "Little Ro
106         {
107             logoBox.GetComponent<Image>().sprite = spriteSign;
108         }
109         else
110             logoBox.GetComponent<Image>().sprite = spriteGeneral;
111     }
112 }
```

E.27 Missions

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class Missions : MonoBehaviour
7  {
8      [SerializeField] GameObject missionBox;
9      [SerializeField] Text missionText;
10     private string mission;
11
12     private bool isTriggered;
13
14     public static Missions Instance {get; private set;}
15
16     private void Awake()
17     {
18         Instance = this;
```

```
19      }
20
21      public void HideMission()
22      {
23          missionBox.SetActive(false);
24      }
25
26      public void ShowMission()
27      {
28          if(isTriggered)
29          {
30              missionBox.SetActive(true);
31          }
32      }
33
34      public void NextMission(string mission)
35      {
36          isTriggered = true;
37          this.mission = mission;
38          missionText.text = this.mission;
39          ShowMission();
40      }
41  }
```

REFERENCES

[1] T. Abel and D. McQueen, *Critical health literacy and the covid-19 crisis*, (2020).

[2] B. Alexander, D. Pearce, M. Lee, M.-A. Detharet, and M. Hyakutake, *Game-based learning an emerging market analysis*, (2022).

[3] J. Alghamdi and C. Holand, *Game-play: Effects of online gamified and game-based learning on dispositions, abilities and behaviours of primary learners*, (2017).

[4] M. Arztmann, L. Hornstra, J. Jeuring, and L. Kester, *Effects of games in stem education: A meta-analysis on the moderating role of student background characteristics*, (2022).

[5] J. Brooke, *Sus: A quick and dirty usability scale*, Usability Eval. Ind., 189 (1995).

[6] J. Brooke, *Sus: A quick and dirty usability scale*, ResearchGate, (1995).

[7] A. Brooks, *What is community health and why is it important?*, (2019).

[8] FilipiKnow, *The filipino doctor who helped discover erythromycin (but never got paid for it)*, 2022.

[9] C. for Disease Control and Prevention, *What is health literacy?*, 2023. Last reviewed March 31, 2023.

[10] C. for Teaching Excellence, *Gamification and game-based learning*, (2022).

[11] J. Hamari, J. Kovisto, and H. Sarsa, *Does gamification work? — a literature review of empirical studies on gamification*, (2014).

[12] A. Hellerstedt and P. Mozelius, *Game-based learning - a long history*, 06 2019.

[13] S. P. Hwa, *Pedagogical change in mathematics learning: Harnessing the power of digital game-based learning*, (2018).

[14] E. E. Kabanova and E. A. Vetrova, *The use of modern electronic gadgets in the educational process of the university*, (2019).

[15] A. Katsaounidou, L. Vrysis, R. Kotsakis, C. Dimoulas, and A. Veglis, *Mathe the game: A serious game for education and training in news verification*, (2019).

[16] J. Kirriemuir and C. A. McFarlane, *Literature review in games and learning*, (2004).

[17] C. R. S. Madarang, *Metro manila residents are panic buying amid covid-19. why they do it and why they shouldn't*, (2020).

[18] M. P. Magdalena, *Digital advertising – the management of online forwarding news and products*, (2019).

[19] L. Marshall, D. Finch, and J. Bibby, *The nation's health as an asset building the evidence on the social and economic value of health*, (2018).

[20] R. E. Mayer, *Computer games in education*, 70 (2019).

[21] S. McLeod, *Kolb's learning styles and experiential learning cycle*, (2017).

[22] J. P. S. J. MD, *Health literacy for all filipinos*, Philippine Medical Association, (2019), p. 1.

[23] R. F. Music, *Quality music and sounds for videos, trailers, animation, games*, (2020).

[24] C. J. C. Nicomedes and R. M. A. Avila, *An analysis on the panic during covid-19 pandemic through an online form*, (2020).

[25] D. Nutbeam, *Health literacy as a population strategy for health promotion*, (2017).

[26] C. of Disease Control and Prevention, *Understanding health literacy*, 2022. Last reviewed September 13, 2022.

[27] D. of Education, *Standards and competencies for five-year-old filipino children*, (2016).

[28] U. of Waterloo, *Gamification and game-based learning*, 2022. Last modified September 22, 2022.

[29] W. H. Organization, *Advice for the public*, (2019).

[30] ——, *10 global health issues to track in 2021*, (2020).

[31] L. Paakkari and O. Okan, *Covid-19: Health literacy is an underestimated problem*, (2020).

[32] V. Peters, G. Vissers, and G. Heijne, *The validity of games*, (1998).

[33] A. Pho and A. Dinscore, *Game-based learning*, (2015).

[34] N. Phutela, A. N. Chowdary, S. Anchlia, D. Jaisinghani, and G. Gabrani, *Unlock me: A real-world driven smartphone game to stimulate covid-19 awareness*, (2022).

[35] T. Pinas, *Deped melcs: English*, 2022.

[36] ——, *Deped melcs: Mapeh*, 2022.

[37] A. Protopsaltis, *Learning theories and serious games*, (2011).

[38] S. Rapini, *Beyond textbooks and lectures: Digital game-based learning in stem subjects*, (2012).

[39] B. Richards and P. Frankland, *The persistence and transience of memory, abilities and behaviours of primary learners*, (2017).

[40] R. Rivas, *Hoarding, overpricing would lead to criminal charges, warns dti*, (2020).

[41] L. Saunders and M. A. Wong, *Instruction in libraries and information centers: An introduction*, (2020).

[42] S. Seereeram, *Dr. aguilar's antibiotic discovery in the philippines*, 2020.

[43] H. Spring, *Health literacy and covid-19*, Health Information & Libraries Journal, (2020), pp. 171–72.

[44] W. T, *Measuring and interpreting system usability scale (sus)*, 2021.

[45] Y. Terada, *Why students forget—and what you can do about it*, (2017).

[46] L. Thomas, *Quasi-experimental design — definition, types & examples*, 2020. Revised on July 21, 2022.

[47] N. Thomas, *How to use the system usability scale (sus) to evaluate the usability of your website*, 2019.

[48] A. Toda, P. Valle, and S. Isotani, *The dark side of gamification: An overview of negative effects of gamification in education*, (2018).

[49] M. C. C. Tolabing, K. C. D. Co, O. M. Mendoza, N. R. C. Mira, R. R. Quizon, M. S. B. Tempongko, M. A. M. Mamangon, I. T. O. Salido, and P. W. Chang, *Prevalence of limited health literacy in the philippines: First national survey*, (2022).

[50] T. Town, *Game-based learning (gbl): The oldest tool of education*, (2022).

[51] J. Trybus, *Game-based learning: What it is, why it works, and where it's going*, (2012).

[52] E. University, *What is community health?*, (2020).

[53] K. S. University, *Successive approximation model (sam)*, (2021).

[54] W. G. University, *Experiential learning theory*, 2020.

VITA

Josef Rex O. Armenta is BS Information Technology student of the Department of Computer Science at the Ateneo de Naga University.

Jefen B. De Villa is BS Information Technology student of the Department of Computer Science at the Ateneo de Naga University.