

# Interactive Learning Adventures: Learn Coding Through Gamification

BY  
ONG ZHAN WEN

A REPORT  
SUBMITTED TO  
Universiti Tunku Abdul Rahman  
in partial fulfillment of the requirements  
for the degree of  
BACHELOR OF COMPUTER SCIENCE (HONOURS)  
Faculty of Information and Communication Technology  
(Kampar Campus)

JUNE 2024

## REPORT STATUS DECLARATION FORM

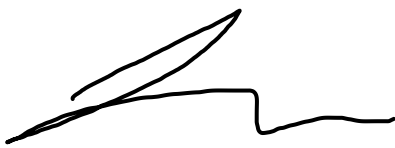
**Title:** INTERACTIVE LEARNING ADVENTURES: LEARN CODING  
THROUGH GAMIFICATION

**Academic Session:** JUNE 2024

I ONG ZHAN WEN  
**(CAPITAL LETTER)**

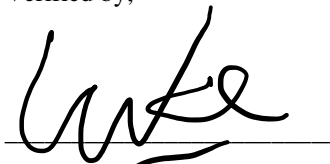
declare that I allow this Final Year Project Report to be kept in  
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.



(Author's signature)

Verified by,



(Supervisor's signature)

**Address:**

1503, Jalan Tanjung Bendahara  
05300, Alor Setar, Kedah

Mr Luke Lee Chee Chien

Supervisor's name

**Date:** 13/09/2024

**Date:** 13/09/2024

<b>Universiti Tunku Abdul Rahman</b>			
Form Title : <b>Sample of Submission Sheet for FYP/Dissertation/Thesis</b>			
Form Number: <b>FM-IAD-004</b>	Rev No.: <b>0</b>	Effective Date: <b>21 JUNE 2011</b>	Page No.: <b>1 of 1</b>

**FACULTY OF INFORMATION TECHNOLOGY**

**UNIVERSITI TUNKU ABDUL RAHMAN**

Date: 28/08/2024

**SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS**

It is hereby certified that ONG ZHAN WEN (ID No: 21ACB04076) has completed this final year project entitled “ Interactive Learning Adventures: Learn Coding through Gamification ” under the supervision of Mr. Luke Lee Chee Chien (Supervisor) from the Department of Computer Science, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,



(ONG ZHAN WEN)

\*Delete whichever not applicable

## DECLARATION OF ORIGINALITY

I declare that this report entitled “**INTERACTIVE LEARNING ADVENTURES: LEARN CODING THROUGH GAMIFICATION**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.



Signature : \_\_\_\_\_

Name : Ong Zhan Wen

Date : 28/08/2024

## **ACKNOWLEDGEMENTS**

I would like to express my sincere thanks and appreciation to my supervisor, Mr Luke Lee Chee Chien, who provided me with this rare opportunity to create a fully functional game for education purposes. He has also provided me with useful advice and tips during development. This project marks my initial foray into establishing a career in the game development field.

Finally, I would like to thank my family members for their love, support, and continuous encouragement throughout the course.

## **ABSTRACT**

Coding through gamification introduces the concept of using game-like elements in the context of teaching coding skills. This approach aims to enhance the learning experience and engagement of learners in coding, which is a traditionally complex and technical subject. By incorporating elements such as challenges, rewards, and interactive scenarios inspired by games, coding education becomes more accessible, enjoyable, and effective. Furthermore, with the target subjects being children, chances of them being able to take an interest into coding through the usage of gamification will be more successful. Teaching coding through gamification for children will have benefits such as increased motivation, improved problem-solving skills, and the development of a growth mindset. It can also highlight how gamification encourages learners to explore coding concepts independently, fostering creativity and a deeper understanding of programming principles.

# TABLE OF CONTENTS

<b>TITLE PAGE</b>	<b>i</b>
<b>REPORT STATUS DECLARATION FORM</b>	<b>ii</b>
<b>FYP THESIS SUBMISSION FORM</b>	<b>iii</b>
<b>DECLARATION OF ORIGINALITY</b>	<b>iv</b>
<b>ACKNOWLEDGEMENTS</b>	<b>v</b>
<b>ABSTRACT</b>	<b>vi</b>
<b>TABLE OF CONTENTS</b>	<b>vii</b>
<b>LIST OF FIGURES</b>	<b>x</b>
<b>LIST OF TABLES</b>	<b>xii</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xiii</b>
<b>CHAPTER 1 INTRODUCTION</b>	<b>1</b>
1.1 Project Background	1
1.2 Problem Statement	1
1.3 Motivation	2
1.4 Project Scope and Objectives	2
1.5 Contribution	3
1.6 Report Organization	3
<b>CHAPTER 2 LITERATURE REVIEW</b>	<b>5</b>
2.1 Swift Playground	5
2.2 Scratch	7
2.3 CodinGame	8
2.4 Limitations of Previous Works/ Studies	9
2.4.1 Swift Playground	9
2.4.2 Scratch	9
2.4.3 CodinGame	10
2.5 Proposed Solution	10

<b>CHAPTER 3 SYSTEM METHODOLOGY</b>	<b>11</b>
3.1 System Design	11
• Concept Design	11
• Story Writing and Planning	12
• Syllabus Integration	12
• Game Play Development	13
3.2 System Architecture Diagram	14
3.3 Use Case Diagram and Description	16
3.4 Activity Diagram	19
<b>CHAPTER 4 SYSTEM DESIGN</b>	<b>22</b>
4.1 System Block Diagram	22
4.2 System Components Specifications	23
4.2.1 User Control	23
4.2.2 Input Manager	23
4.2.3 Player Script	24
4.2.4 Quizzes	26
4.2.5 Dialogue System	27
4.2.6 Boss Script	28
4.2.7 Enemy Script	30
4.2.8 Platform	30
4.2.9 Game Object	31
4.2.10 Game	32
<b>CHAPTER 5 SYSTEM IMPLEMENTATION</b>	<b>33</b>
5.1 Hardware Specifications	33
5.2 Software Specifications	33
5.3 Setting and Configuration	34
5.4 System Operation	35
5.5 Implementation Issues and Challenges	40
5.6 Concluding Remark	41



<b>CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION</b>	<b>42</b>
6.1 System Testing and Performance Metrics	42
6.2 Testing Setup and Result	42
6.3 Project Challenges	44
6.4 Objectives Evaluation	44
6.5 Concluding Remark	45
<b>CHAPTER 7 CONCLUSION AND RECOMMENDATION</b>	<b>46</b>
7.1 Conclusion	46
7.2 Recommendation	46
<b>REFERENCES</b>	<b>48</b>
<b>WEEKLY LOG</b>	<b>49</b>
<b>POSTER</b>	<b>55</b>
<b>PLAGIARISM CHECK RESULT</b>	<b>56</b>
<b>FYP2 CHECKLIST</b>	<b>59</b>

## LIST OF FIGURES

<b>Figure Number</b>	<b>Title</b>	<b>Page</b>
Figure 2.1.1	Code guide function in Swift Playground	5
Figure 2.1.2	User interface and environment for coding in Swift Playground	6
Figure 2.2.1	Scratch programming interface for drag and drop coding	7
Figure 2.2.2	Home page for Scratch website that allows users to publish their projects for other users to view and interact with.	7
Figure 2.3.1	Coding interface in CodinGame that allows users to input codes that controls the movement of characters in game	8
Figure 2.3.2	Different games that focus on different concepts in programming are available in CodinGame main page	8
Figure 3.1.1	An overview of the animation sheet of the main character	11
Figure 3.1.2	An example of the layout for the selected tile map	12
Figure 3.2.1	The System Architecture Diagram	14
Figure 3.3.1	Use Case Diagram for the system	16
Figure 3.4.1	Activity Diagram of Start Game	19
Figure 3.4.2	Activity Diagram for gameplay loop	20
Figure 3.4.3	Activity Diagram for Interaction loop	21
Figure 4.1.1	The overall system block diagram	22
Figure 4.1.2	Overview of the collection of input keys that are registered in the system	23
Figure 4.1.3	Code snippet of MovePressed Function	24
Figure 4.1.4	GetMoveDirection() method	24
Figure 4.1.5	General Flowchart of Player Movement	25
Figure 4.1.6	Code snippet of quiz loop logic in Ink	26
Figure 4.1.7	Some snippets of questions available that are related to C++	27
Figure 4.1.8	Code Snippet of dialogue system	28
Figure 4.1.9	Dialogue System that is available in game	28
Figure 4.1.10	Code snippet of boss script	29
Figure 4.1.11	Code snippet of boss dialogue script that works together with Dialogue System	29

<b>Figure Number</b>	<b>Title</b>	<b>Page</b>
Figure 4.1.12	A code snippet of the general enemy in game	30
Figure 4.1.13	A floating platform with waypoints	30
Figure 4.1.14	A breakable platform that looks like a normal platform	31
Figure 4.1.15	Floating chests that are available in the game world	31
Figure 4.1.16	A wooden plaque that provide hints for the player	31
Figure 5.3.1	Creating a new scoped registry	34
Figure 5.3.2	Installing the necessary plugins	35
Figure 5.4.1	Main Menu screen for the game	35
Figure 5.4.2	First Floor of the game	36
Figure 5.4.3	Second Floor of the game	36
Figure 5.4.4	Wooden plaque that guides players	37
Figure 5.4.5	Floating chests puzzle after the wooden plaque	37
Figure 5.4.6	Encounter with slimes as a challenge	38
Figure 5.4.7	A value that appears above the slime	38
Figure 5.4.8	Breakable platforms with answers	39
Figure 5.4.9	Lock and key system	39
Figure 5.4.10	Final boss of the level	40

## LIST OF TABLES

<b>Table Number</b>	<b>Title</b>	<b>Page</b>
Table 3.3.2	Start Game Description	16
Table 3.3.3	Move Character Description	17
Table 3.3.4	Interactions with Game Object Description	17
Table 3.3.5	View Health Description	17
Table 3.3.6	Answer Question Description	17
Table 3.3.7	Pause Game Description	18
Table 5.1.1	Specifications of Laptop	33
Table 6.1.1	Start Game Test Case	42
Table 6.1.2	Move Character Test Case	42
Table 6.1.3	Interact with Game Objects Test Case	43
Table 6.1.4	In game character health Test Case	44

## **LIST OF ABBREVIATIONS**

*NPC* Non Playable Character

# Chapter 1

## Introduction

In this chapter, the background and motivation of my research and my contributions to the field will be outlined.

### 1.1 Project Background

Early IT education has become more than a steppingstone to work chances in this era of rapid technological growth. It is the foundation of digital literacy, preparing youngsters to navigate our increasingly digital environment with confidence. This education is more than just programming and understanding concepts, it is also a gateway to important life skills. Young learners develop problem-solving skills, logical thinking, and creativity because of IT education, and these traits carry over into a variety of job pathways. Therefore, this software aims to serve as an exciting portal to the world of IT, and acts as a vanguard of this educational shift. It provides a riveting trip by combining enjoyable challenges with colourful images and dynamic events.

### 1.2 Problem Statement

In this current world where technology saturates every facet of our existence, nurturing a generation adept at navigating this digital landscape becomes an imperative. With the rise of digital technology such as Big Data, Artificial Intelligence, and Internet of Things(IoT), having a good grasp of coding knowledge will prepare the future generation to face the digitalized future head-on. With this, parents should properly equip their children with the basic knowledge of programming from an early age[1]. Equipping children with fundamental IT skills is tantamount to gifting them a universal language, one that holds the key to unlocking a realm of endless possibilities. By introducing them to IT education at an early juncture, we extend beyond the role of mere educators – we become enablers of future creators, innovators, and astute critical thinkers. Moreover, this early initiation into IT education bridges the chasm of digital disparity that exists in our society. It guarantees that every child, irrespective of their background, gains access to the skills that will inevitably shape the future job panorama. This democratization of knowledge creates a paradigm shift, elevating technology from being a passive presence in their lives to a tool they wield, thus transforming them from consumers to active contributors.

### **1.3 Motivation**

In the current situation which were defined by technological leaps and digital prowess, the value of early Information Technology (IT) education transcends far beyond the scope of mere job prospects. It is a gateway to digital literacy, an empowerment that equips children to navigate the increasingly digitalized world with confidence, discernment, and a holistic understanding of the technology that envelops us. By introducing IT education to young minds from the outset, we instill in them the vital skills to comprehend, evaluate, and utilize the technological tools that are seamlessly woven into the fabric of modern existence. Building something from scratch prompts children to use their imagination and problem solving skills all together at the same time, which leads to improvement in creativity. Therefore, through coding, children are exposed to the art of creative thinking and superb problem solving skills [2]. Empowered by this software, children embark on an exhilarating journey into the realm of coding, where gripping challenges and immersive gameplay become their guiding stars. This software aims to be a portal to a captivating universe where each challenge becomes an opportunity, and each triumph is a testament to their growing prowess.

### **1.4 Project Scope and Objectives**

As of the current era, technology exists in our everyday life, therefore it is important to nurture a new generation that has a fundamental understanding of its functioning. With this aim in mind, the project scope is to create a software that incorporates gaming elements into coding. With an immersive storyline, exciting challenges and quests, young users will embark into the world of coding.

C++ language is used as a medium to expose the users to the concepts of programming as adequate knowledge of C++ allows a basic understanding of other programming languages as it covers the fundamentals principles of other programming languages too.

For the design of interface, this software features a classic 2D style platform game, where users will explore the world and solve challenges that involves programming knowledge along the way. The interface will be simple, clear, and concise, fitting with the theme of minimalism without forgoing fun features.

The software will undergo a rigorous testing process to make sure it satisfies the benchmark for performance, dependability, and usability. As the overall aim is to create a game that is attractive and pleasant looking in terms of visuals without forgoing its primary objective which

is to spread the knowledge of coding through gamification process, each testing phase will be highly monitored to ensure that users are able to experience the best out of it when it is released into the market.

## **1.5 Contributions**

This software acts as a beacon of hope in the field of coding education for children. It works as a master translator, breaking down complex lines of code into their most basic components and making them understandable and relevant. The software's strength is in its capacity to disentangle the complexities and present them to young minds in a way that sparks understanding and curiosity. Aside from simplification, this programme has a unique capability: it can cultivate a genuine enthusiasm for coding. It is a digital portal that invites students into the enthralling world of coding, providing more than simply instruction but also adventures. Its interface is an interactive canvas where abstract notions become real accomplishments and coding ceases to be a faraway concept to become an important part of their environment. It also creates an environment in which learning feels like fun. Children are more than just learners in this digital world; they are explorers, creators, and problem solvers. The appeal of coding extends beyond lines of text to include interactive scenarios, animated challenges, and a variety of interesting activities. This software uses gamification elements, such as levels, rewards, and points to further enhance the learning experience. The integration of such elements creates a sense of engagement, challenge, and achievement in children, making learning more engaging and enjoyable. Through every obstacle overcome, children will be more motivated to further improve themselves and aiming for a higher level of mastery[3]. Furthermore, this software allows youngsters to confidently navigate the complex environment of coding. It provides the required tools, ranging from step-by-step tutorials to real-time assistance, allowing students to learn the subtleties of coding without difficulty.

## **1.6 Report Organisation**

The details of the research are shown in the following chapters. In chapter 2, similar applications, websites, and software are reviewed and discussed based on its functionalities and interaction with users. Later then, the reviewed products are being analyzed for its



limitations and a proposed method for the project will be outlined. Chapter 3 will describe the overall system methodology which includes the System Architecture Diagram, Use Case Diagram and descriptions, and Activity Diagram. Chapter 4 will describe the preliminary work done to realize this project. Lastly, Chapter 5 will summarize the overall outcome of the project and how well can it be used by users in real life.

# Chapter 2

## Literature Review

### 2.1 Swift Playground

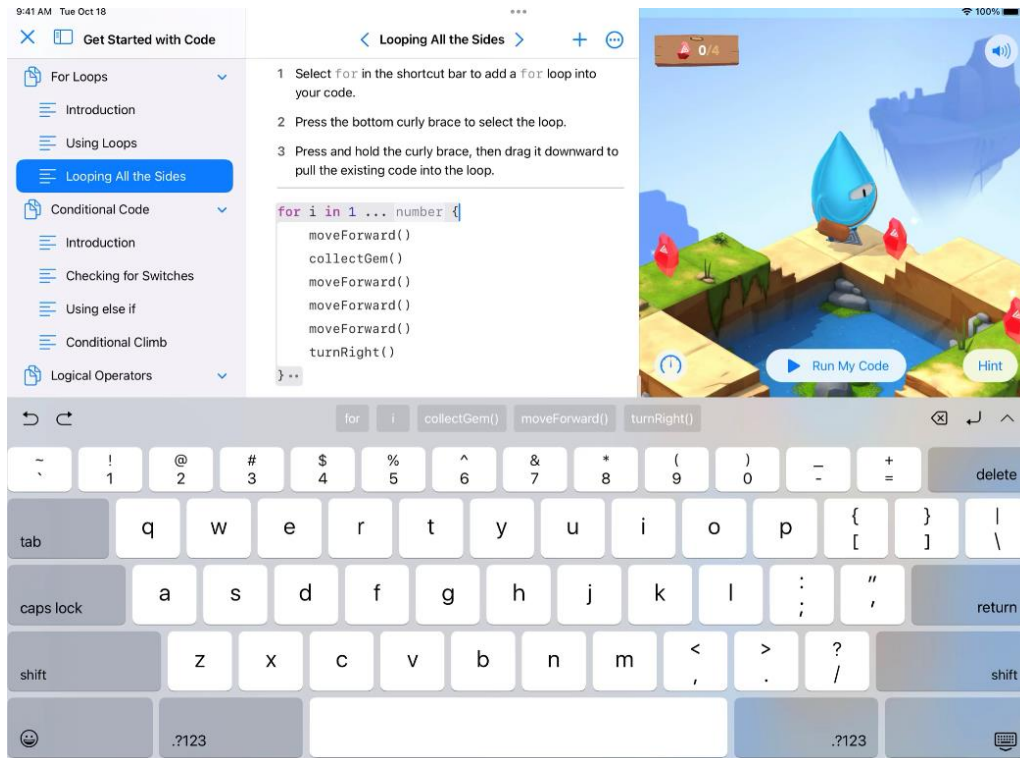


Figure 2.1.1 Code guide function in Swift Playground

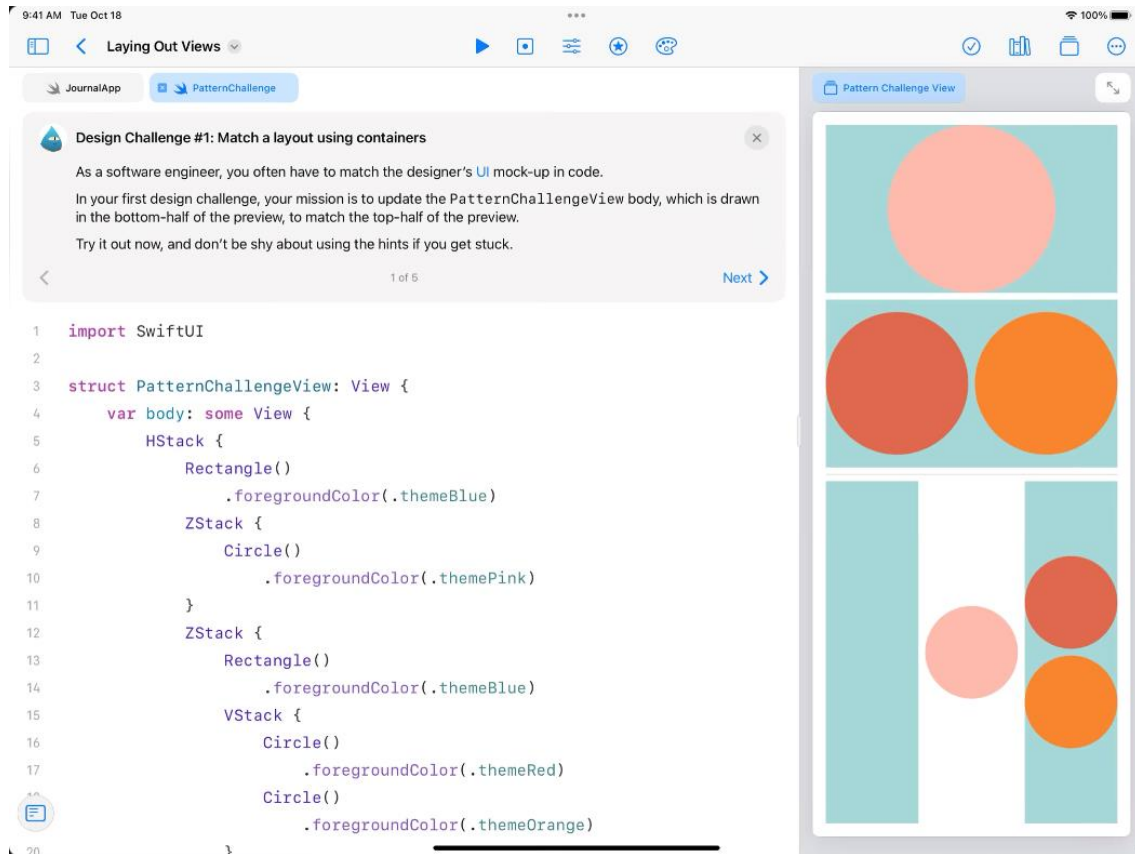


Figure 2.1.1 User interface and environment for coding in Swift Playground

Swift Playground makes it fun to learn code and build real apps. Children can solve interactive puzzles following the guide “Getting started with Code” and learn the basics of building apps in “Get Started with Apps”. Children can experiment a wide range of challenges and samples that lets them explore unique coding experiences. Children without any prior coding experience can try the application as it is suitable for beginners. The programming language involved in this application is Swift and SwiftUI, which was created by Apple and used widely by professionals.

## 2.2 Scratch

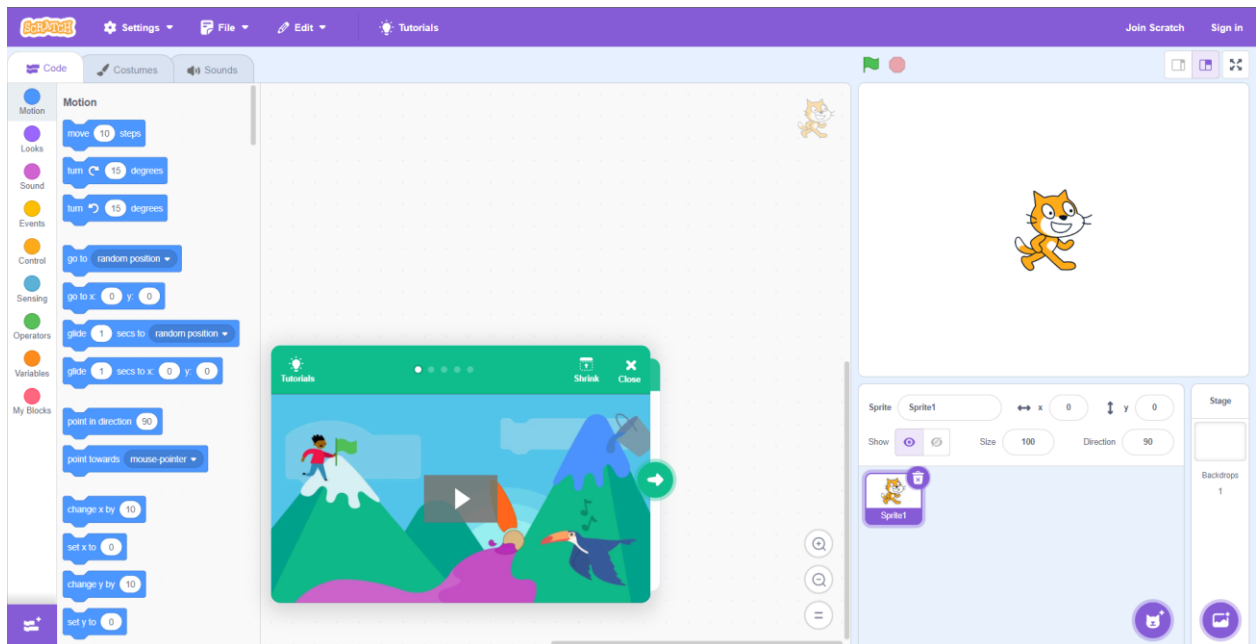


Figure 2.2.1 Scratch programming interface for drag and drop coding

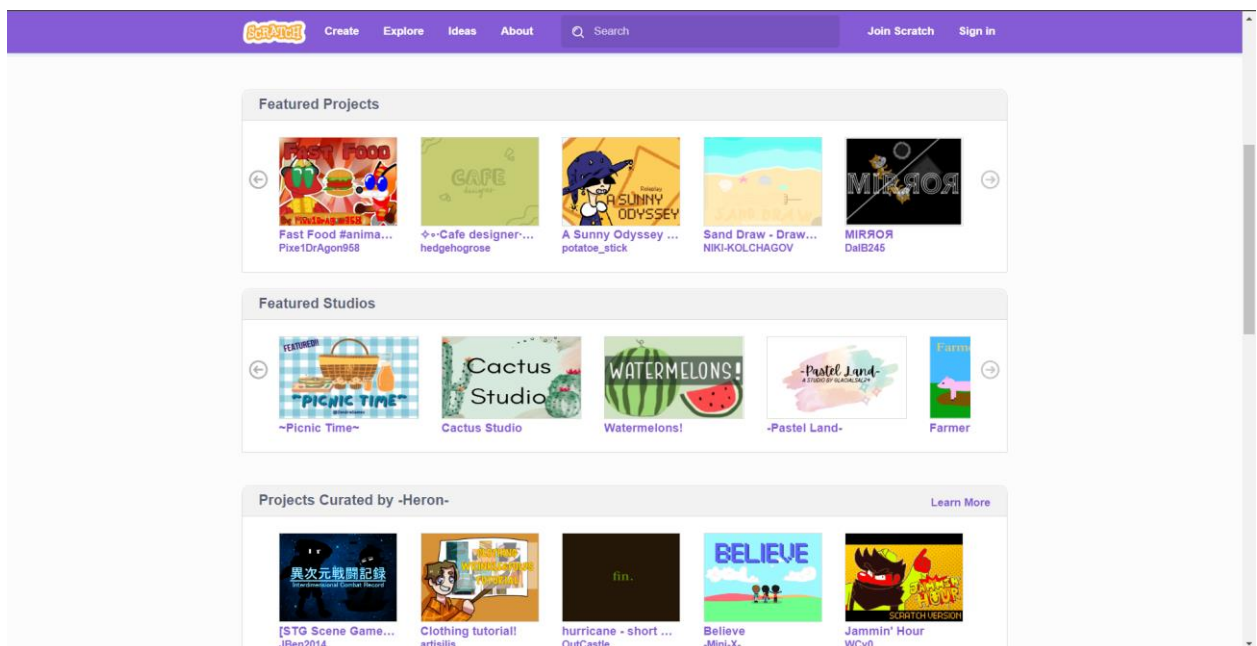


Figure 2.2.2 Home page for Scratch website that allows users to publish their projects for other users to view and interact with.

Scratch is a block-based programming language available online suitable for children and those who are new to the world of programming. Scratch has a simple visual interface that allows users to create digital stories, games, and animations. Users can learn about the basics of

Scratch programming language such as events, control variables and more by building their own game and projects. The projects created are then shared to allow users to interact with it. Scratch helps children to learn important strategies for solving problems, designing projects, and communicating ideas.

## 2.3 CodinGame

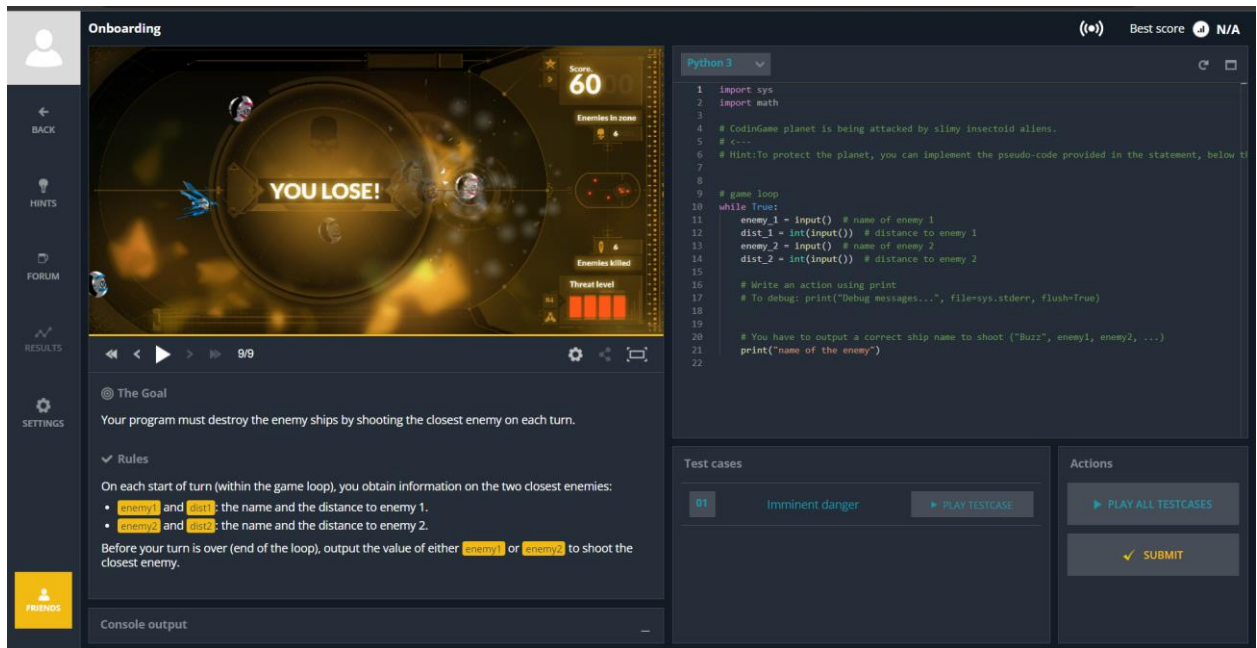


Figure 2.3.1 Coding interface in CodinGame that allows users to input codes that controls the movement of characters in game

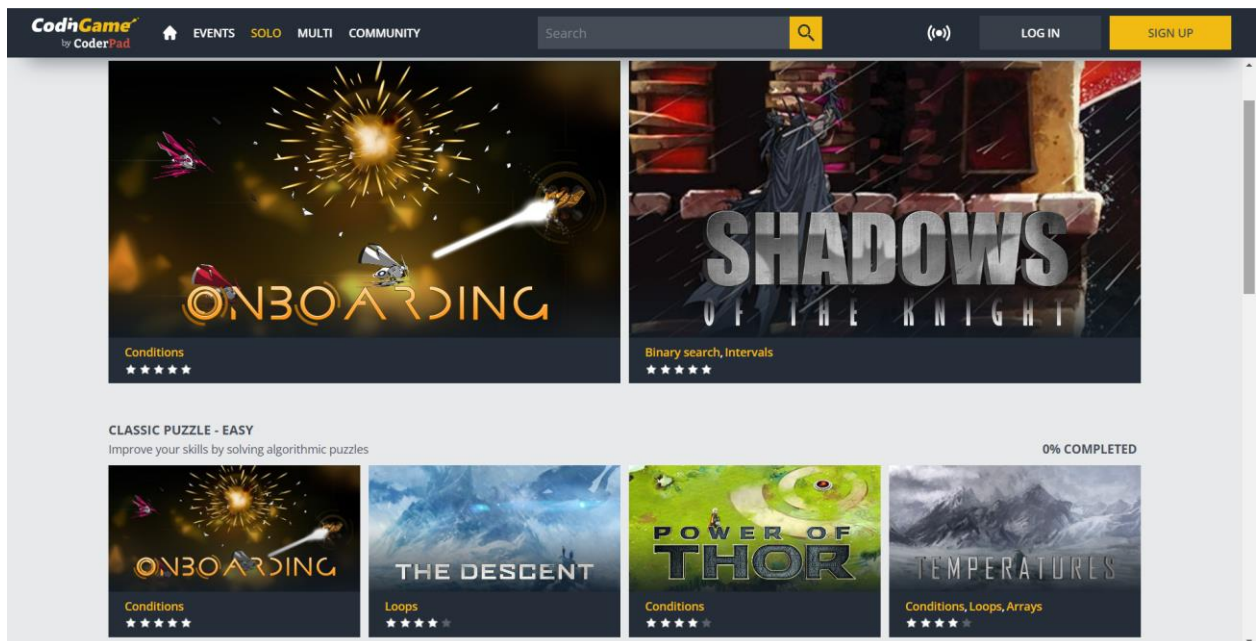


Figure 2.3.2 Different games that focus on different concepts in programming are available in CodinGame main page

CodinGame is an online platform that allows users to train and improve their skills in different concepts of programming using games and puzzles. CodinGame has games with over 25 languages, providing users with different types of languages and concepts to practice with. This platform also supports multiple users, which allows multiplayer events and competition to happen.

## **2.4 Limitation of previous works/ studies**

### **2.4.1 Swift Playground**

While this application certainly proves beneficial for children embarking on their programming journey, a few limitations dampen its potential. Notably, its exclusivity to iPad hinders accessibility, catering solely to those with access to this device and operating system. The implications are substantial, as many families lack iPads, thus limiting the application's reach and impact. Moreover, the concept of coding through a tablet interface remains unconventional within the programming sphere. This introduces a learning curve, potentially deterring some users from fully engaging with the coding world. Finally, there is a potential concern in the application's approach to introducing young users to coding. An absence of a proper orientation or introductory phase might render the learning experience monotonous and unappealing. Imposing the complexities of coding without establishing its context could lead to disinterest among young users, effectively diminishing the intended excitement and fun that coding endeavours should evoke.

### **2.4.2 Scratch**

Scratch, while a remarkable platform for introducing coding to beginners, does come with certain limitations. Notably, it lacks a clear progression path toward text-based programming languages, making it less conducive for learners aspiring to dive deeper into coding. Moreover, for advanced coders, Scratch might feel somewhat restrictive due to the absence of certain advanced features. Lastly, it does not provide an option for creating 3D projects, limiting the scope of what users can express and explore within the platform. Additionally, given that Scratch operates as a web-based platform, it is important to note that it relies on an internet connection. This dependency can pose challenges for those in areas with limited or no internet

access, making the process of learning programming considerably more difficult under such circumstances.

### **2.4.3 CodinGame**

CodinGame primarily caters to individuals who are already proficient in programming languages. Most of its challenges and games assume a foundational understanding of programming concepts, rendering it less accessible to beginners seeking to learn programming from scratch. Additionally, CodinGame's web-based nature poses a limitation, as it necessitates a continuous internet connection for use, which may not be feasible in all learning environments.

### **2.5 Proposed Solution**

Making the software available on computer is a must as all programming should be done through a computer. This will allow the young users experience and engage fully the joys of coding. Furthermore, to counter the monotonous and unappealing weakness, this project has an immersive in game story that will captivate the young audience, which is also accompanied by colourful visuals. With game elements incorporated into this software, it taps into the intrinsic human desire for achievement, competition, and rewards. The gamification of programming courses motivates young users to actively participate and invest in their learning journey. Furthermore, this software will also be available offline, which allows users to learn coding without relying on Internet connection.

# Chapter 3

## System Methodology

The process of project development is separated into different phases for an easier approach, which were the concept design, story planning and writing, syllabus integration and gameplay development.

### 3.1 System Design

#### Concept Design

The development of this project first starts with the concept design, which includes a character sprite sheet obtained from [4]. This sprite sheet has been selected due to having the necessary animation sets that suits all types of desired scenarios.

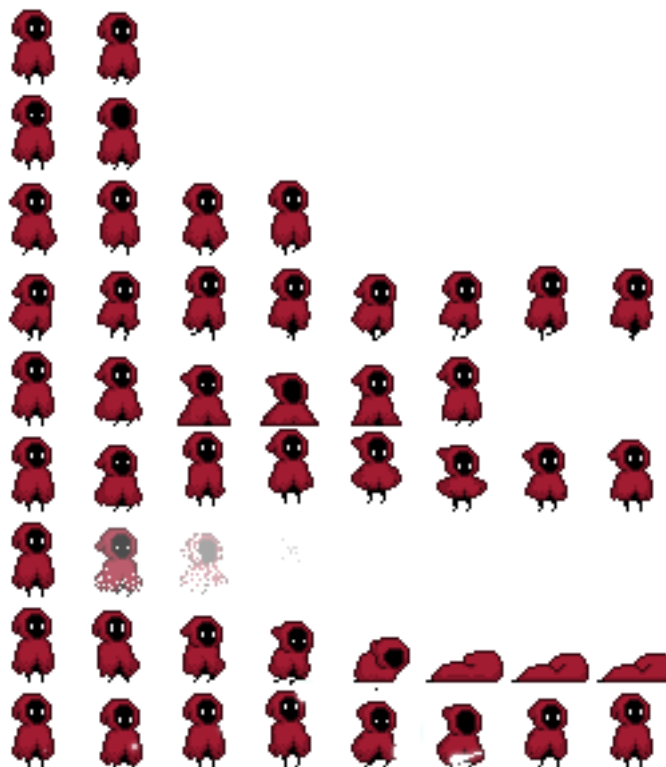


Figure 3.1.1 An overview of the animation sheet of the main character



Besides that, a tile set for the map design has also been selected after a careful consideration of the aesthetics and suitability between the character design and tile map design. Such considerations are necessary for a visually pleasing and engaging game environment.

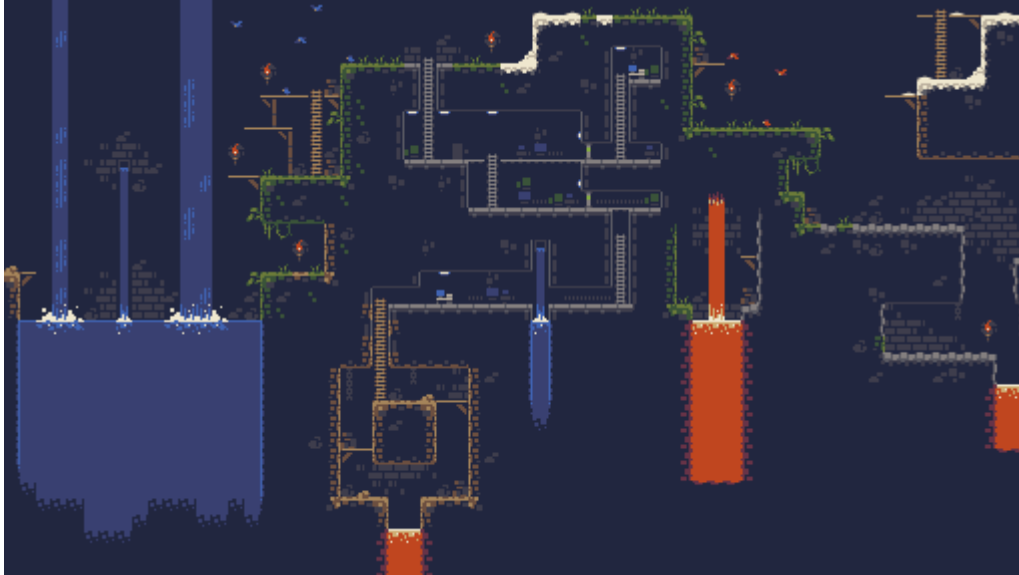


Figure 3.1.2 An example of the layout for the selected tile map

### **Story Writing and Planning**

After deciding on the visuals, the next step of development is the story planning. To create a both fun and engaging game, an interesting storyline is needed to be created that appeals to the boundless imagination of a children's mind without forgoing the educational content. Both the storyline and narration need to provide enough contexts for the interactive puzzle challenges too. The final storyline draft description involves a protagonist (the player/user) who was accidentally summoned to the virtual plane of Viltres and is required to navigate through different levels inspired by the seven deadly sins to save the virtual plane and return to the original world.

### **Syllabus Integration**

To cater the young users, a detailed plan of the syllabus has been decided to allow easy understanding of the programming concepts. Each level has been created to represent a different coding concept. Some of the programming concepts and principles to be taught are variable and data types, arithmetic operations, conditional statements and loops, arrays and strings and so on. With the layout of syllabus, users will be able to correctly grasp the

programming knowledge. For this current prototype with one level, we focus on the basics of programming knowledge which are the data types.

### **Gameplay Development**

The overall system that is developed will only consist of the software. This software is created using Unity application which also acts as the game engine that handles and creates the game world while implementing the game mechanics and handling player interactions. Visual Studio 2022 is also used to create the C# scripts necessary for game mechanics and player interactions.

### 3.2 System Architecture Diagram

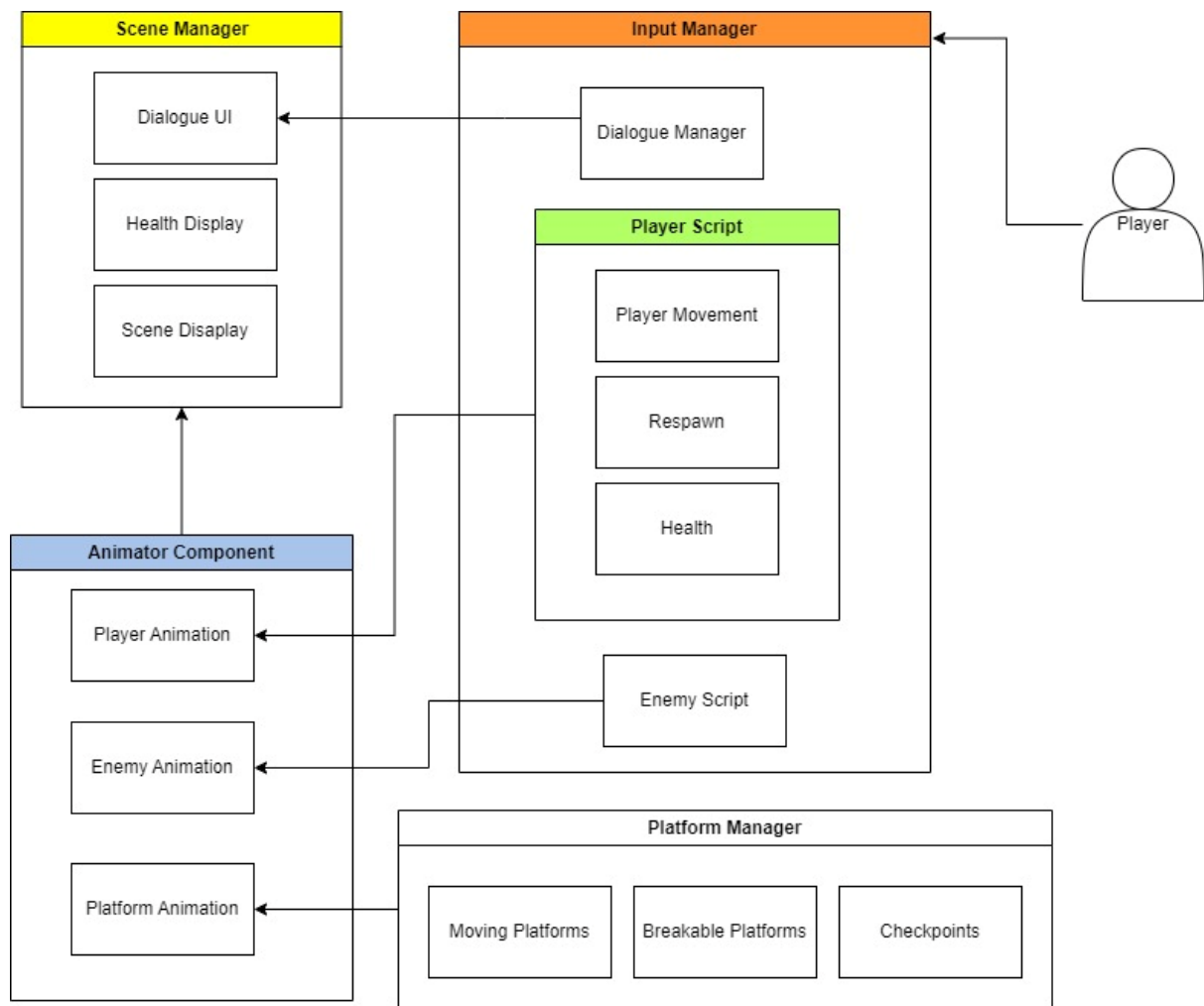


Figure 3.2.1 The system architecture diagram

Figure 3.2.1 illustrates the system architecture of the project, highlighting the key components and their interactions. The core components are designed to manage specific functionalities essential to the gameplay.

For the Input Manager, this component is responsible for capturing and processing player inputs using Unity's latest input system. It also handles the dialogue interactions within the game, ensuring seamless communication between the player and the game world.

As for the player script, it will manage all player-related functionalities, including movement, respawning, and health management. This script ensures smooth player control and behavior in various scenarios.

For platform manager, the component oversees overseeing all platform-related mechanics, such as moving platforms, floating platforms, and checkpoint systems. It dynamically handles interactions between the player and different types of platforms in the game.

The animator component is connected to the Input Manager, Platform Manager, and Player Script, the Animator component controls all animations related to player actions, platform interactions, and overall gameplay. It ensures fluid transitions and visual feedback for each action taken by the player.

Lastly, the scene manager handles the overall management of game scenes and UI related elements. It oversees the transitions between different levels, loading screens and also ensures that the user interface updates appropriately as the player progresses through the game.

Overall, these components form a system that delivers a smooth interactive gameplay experience by efficiently managing inputs, animations, platform behavior and scene transitions.

### 3.3 Use Case Diagram and Description

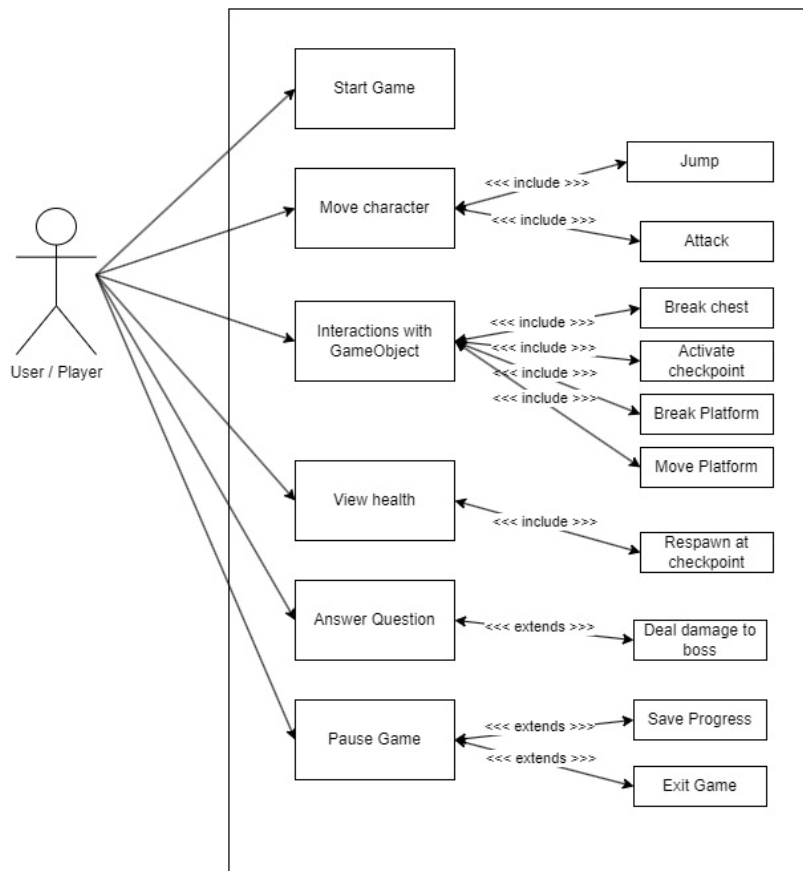


Figure 3.3.1 Use Case Diagram for the system

The Use Case diagram above illustrates the player interactions that are available in the game. The main use cases and their descriptions are shown in the tables below with suitable description:

Table 3.3.2 Start Game Description

Use Case	Start Game
Aim	Start the game
Actor	User / Player
Trigger	When interact with play button

Flow	User/player enters the game
------	-----------------------------

Table 3.3.3 Move Character Description

Use Case	Move character
Aim	To move the character
Actor	User / Player
Trigger	When the input buttons are pressed
Flow	User/player controls the avatar to move and carry out other actions such as Jump or Attack.

Table 3.3.4 Interactions with Game Object Description

Use Case	Interactions with Game Object
Aim	Interact with in game objects
Actor	User / Player
Trigger	When the input buttons are pressed
Flow	User/ player interact with game objects that are available throughout the game world.

Table 3.3.5 View Health Description

Use Case	View Health
Aim	Health points able to be deduct from player
Actor	User / Player
Trigger	When player selects a wrong choice
Flow	User/player will be deducted a health point if they select a wrong choice, and if the user/ player overall health points fall below 0 they will respawn at the nearest check point

Table 3.3.6 Answer Question Description

Use Case	Answer Question
Aim	Answers questions in the form of interaction
Actor	User / Player

Trigger	When in contact with the boss Game Object
Flow	User/player will answer certain amounts of questions when they come into contact with the boss. A point of damage will be dealt to the boss if they answered correctly.

Table 3.3.7 Pause Game Description

Use Case	Pause Game
Aim	Pause the Game
Actor	User / Player
Trigger	When interact with user interface
Flow	User/player is able to pause the game and is presented with either to save their progress or to exit the game.

### 3.4 Activity Diagram

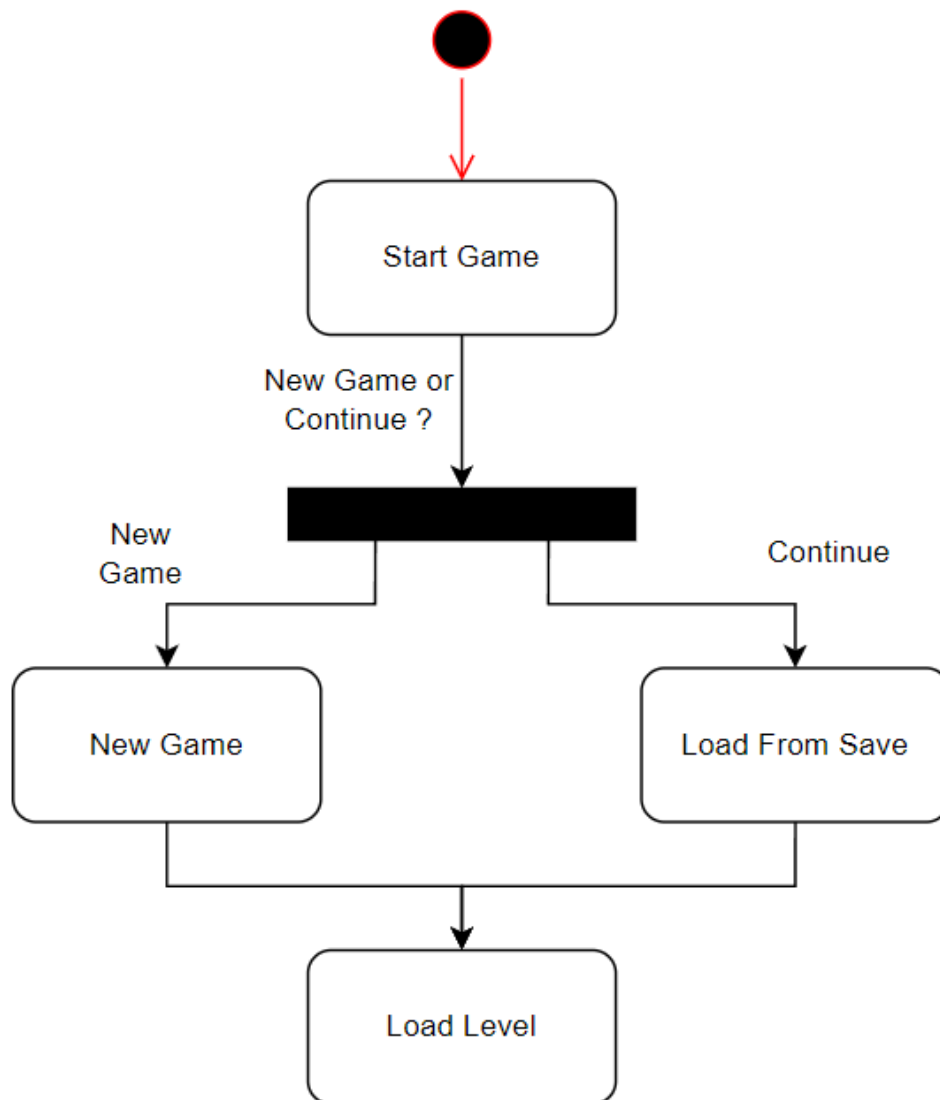


Figure 3.4.1 Activity Diagram of Start Game

The Activity Diagram of the game starting are as follows. Firstly, upon loading the game, users are able to select a new game option or continue from previous save option. If the users select the new game option, then a new game will start. If the users choose the continue option, then the game will load from a save file and users are able to continue their progress.



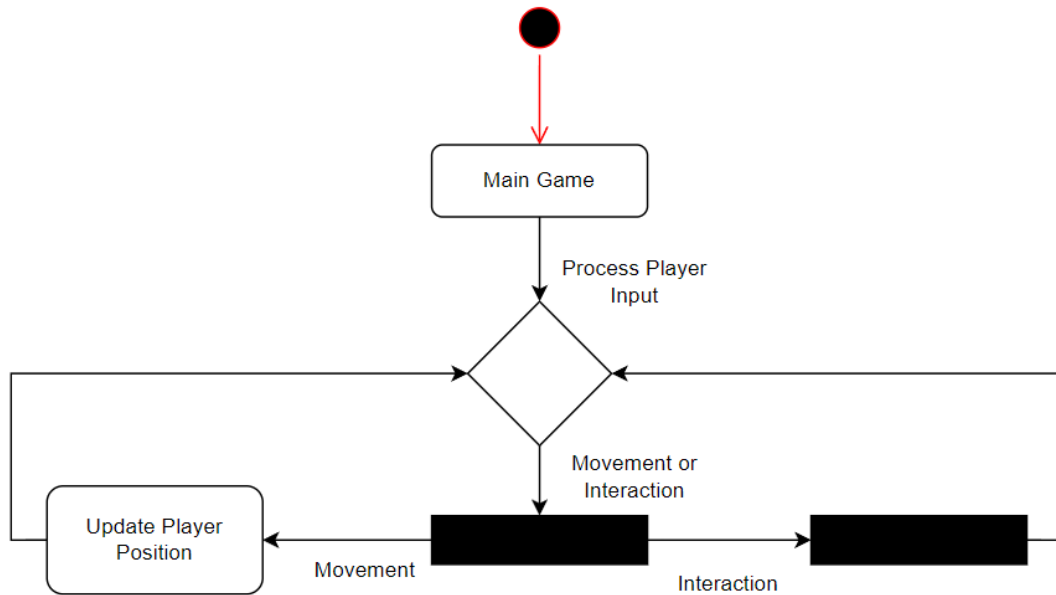


Figure 3.4.2 Activity Diagram for gameplay loop

The Activity Diagram above shows the core gameplay loop that focuses on the player input processing. Upon entering the main game, the game system waits for a player input. When an input key is pressed, the game will start process the player input, which plays a key role for all player driven actions. At the decision point of Movement or Interaction, the player is presented with 2 options to undergo, which is either moving the player in game or interact with game objects that are present throughout the game world. If movement choice is made, the game will update the players in game position. The movement options include walking, jumping, or attacking. After carrying out the requested input, the game system will then go back to awaiting inputs from player.

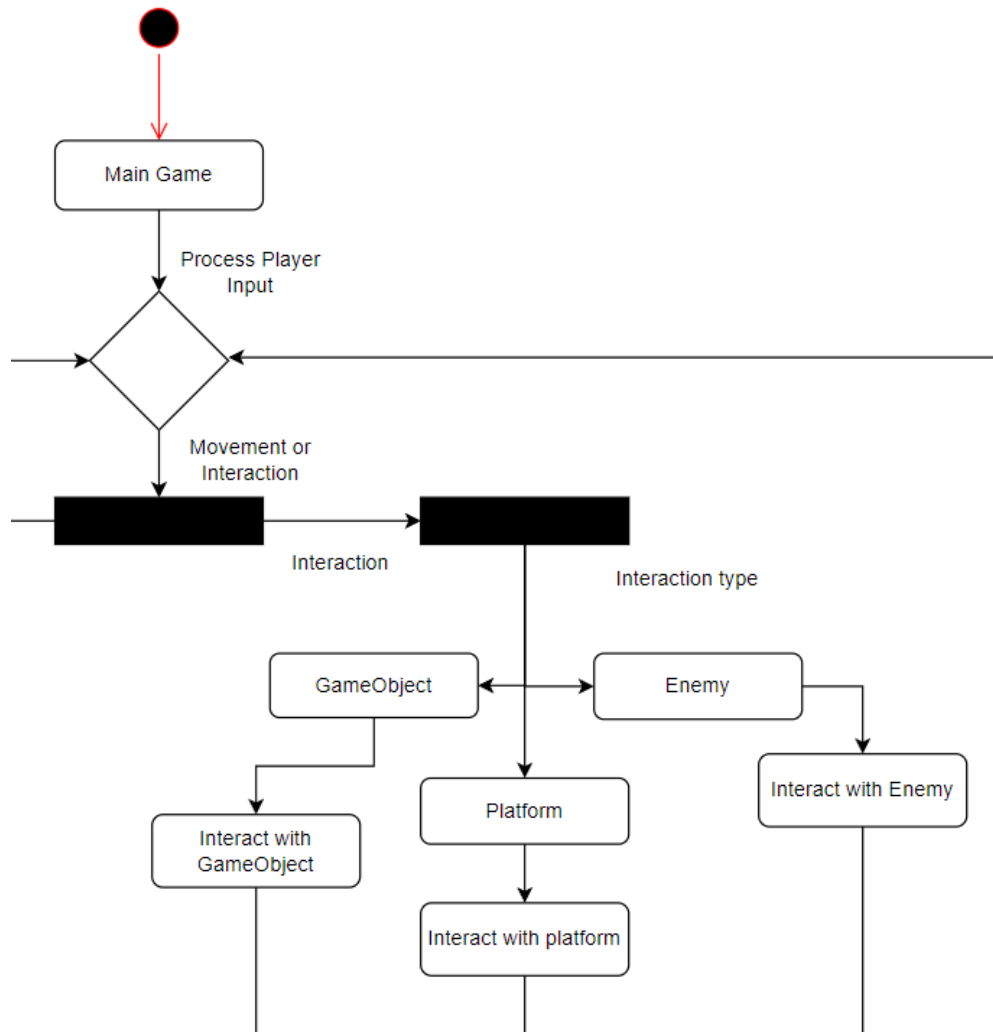


Figure 3.4.3 Activity Diagram for Interaction loop

This activity diagram represents the decision-making process for the interactions part within a game. After entering the Main Game state, which transitions into processing player input. Based on the input, a decision is made whether the player performs a movement or an interaction. If the Interaction choice is made, the process continues by determining the type of interaction. The player can interact with a Game Object or an Enemy, or they can interact with a Platform. Depending on the specific interaction, the system defines how the player interacts with game objects, enemies, or platforms, ultimately guiding the game's response to player actions. After allowing the player to have interaction, the game system will then goes back to wait for new inputs from player.

# Chapter 4

## System Design

### 4.1 System Block Diagram

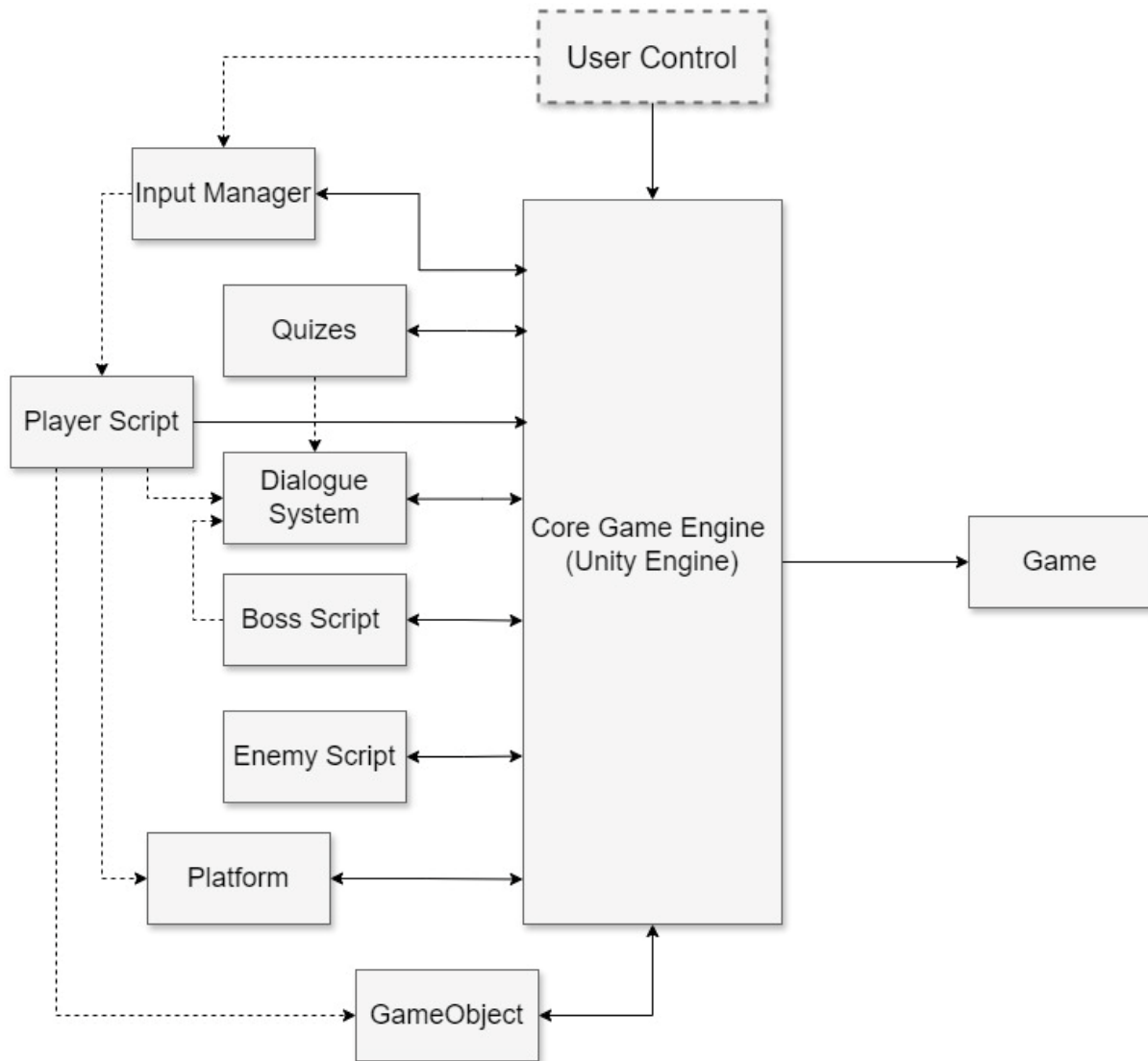


Figure 4.1.1 The overall system block diagram

Figure 4.1.1 shows the overall system block diagram on how the game is built. The entire game is heavily centered around the Core Game Engine, which is the Unity Engine. The central component interacts with other components to create a final product which is the game itself.

## 4.2 System Block Components Specification

### 4.2.1 User Control

The user control refers to the hardware part which is the keyboard and mouse devices. Movement keys that are W, A, S and D are used to control the player in game. Buttons E allows the player to interact with game objects available. The left key of the mouse can also be used to interact with game objects while also serves as an attack button. An overall input keys that are registered with the Unity Input System are shown below.

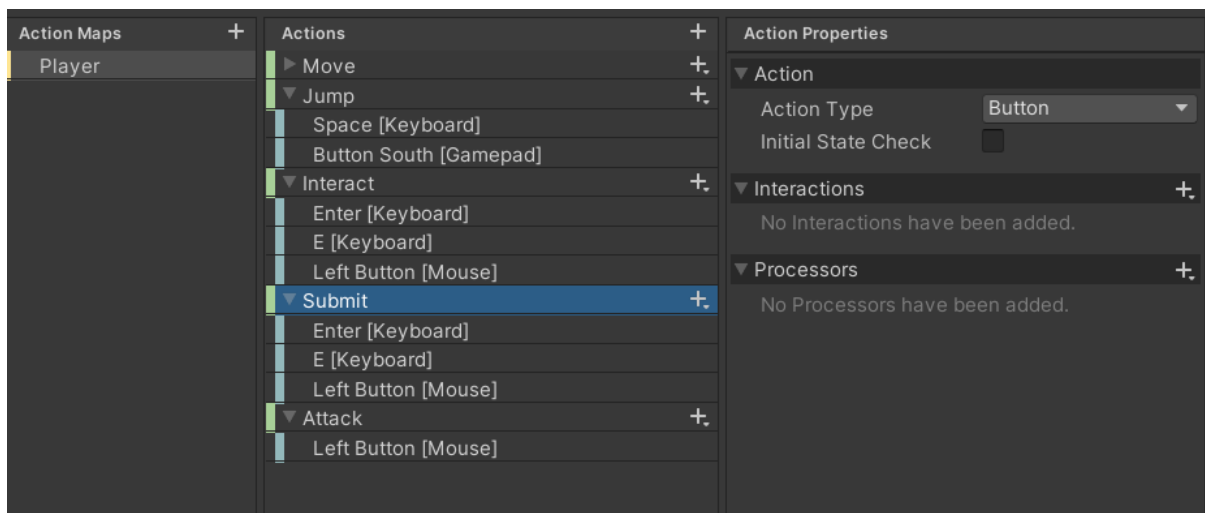


Figure 4.1.2 Overview of the collection of input keys that are registered in the system.

### 4.2.2 Input Manager

The Input Manager refers to the Input Manager script that is created to carry out the inputs that are registered in the Unity Input System. Using the code “`InputAction.CallbackContext`”, it is able to handle the registered buttons to carry out the desired actions. Some of the functions that are available in it are the `MovePressed` function.

```

public void MovePressed(InputAction.CallbackContext context)
{
    if (context.performed)
    {
        moveDirection = context.ReadValue<Vector2>();
    }else if (context.canceled)
    {
        moveDirection = context.ReadValue<Vector2>();
    }
}

```

Figure 4.1.3: Code snippet of MovePressed Function

This function handles the movement input. If the input actions are performed, eg. a key is pressed, it reads a vector 2 value and stores it in the move direction. After the values are stored, there is a GetMoveDirection() method the will returns the current value of moveDirection.

```

public Vector2 GetMoveDirection()
{
    return moveDirection;
}

```

Figure 4.1.4: GetMoveDirection() method.

### 4.2.3 Player Script

The player script is one of the vital parts of a component that is needed in the game. It gives life to the in game avatar that the user is controlling with input keys. A general flowchart for the overall player script has been visualized below.

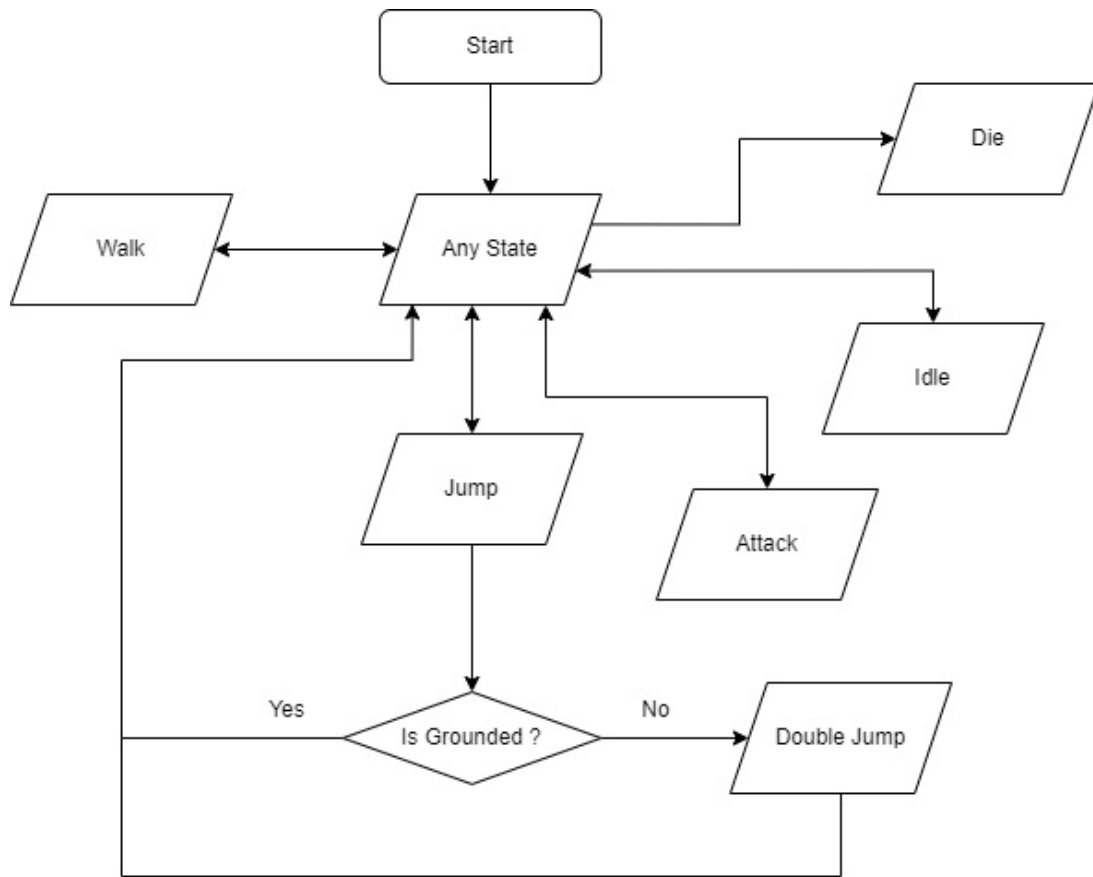


Figure 4.1.5 General Flowchart of Player Movement

A general rundown of the flowchart is as follows. Firstly, when the game is started, the character can enter any state, which includes walk state, idle state or jump state. However, by default, the character will usually enter the idle state. This state will be maintained until there is an input from the user. Depending on the choice made, there are two states that the character can enter. One of which is the walking state. In this state, the character will move along the direction according to user input. Once there is no input from user, the character will exit walk state and goes back to idle state. Another state that the user can input is the Jump state. Upon pressing the spacebar button, the character will perform a jump. And if the user presses the spacebar button again, the player movement script will first check if the character is grounded or not. If the player is grounded, then the double jump action will not be performed. Else, if the character is not grounded, the double jump action can be performed. After the action has been executed, the player will return to any state depending on the user input, else it will return to the idle state. Lastly, once the player's health falls below 0, it will enter the die state.

#### 4.2.4 Quizzes

Quizzes component plays the role of an educational component of the game. Through a story engine known as Inky Editor, it allows the creation of questions in the form of quizzes. Ink scripting language is widely used and accepted by many games as a narrative scripting language. It was largely built around the idea of marking up pure texts with flow in order to produce narrative scripts as stated in [5]. This component works hand in hand with the dialogue manager and boss script to present players questions that are related to C++ knowledge.

```
▼ === question_loop ===  
~ question_count += 1  
{  
- question_count == 1: -> question1  
- question_count == 2: -> question2  
- question_count == 3: -> question3  
- question_count == 4: -> question4  
- question_count == 5: -> question5  
- question_count == 6: -> question6  
- question_count == 7: -> question7  
- question_count == 8: -> question8  
- question_count == 9: -> question9  
- question_count == 10: -> question10  
- else: -> end  
}
```

Figure 4.1.6 Code snippet of quiz loop logic in Ink

```
▼ === question1 ===
What can be considered as an integer ?
* [A whole number]
  ~ correct = true
* [A decimal]
  ~ correct = false
* [A true false question]
  ~ correct = false
* [A group of words]
  ~ correct = false
- -> question\_result

▼ === question2 ===
What data type would you use to store a single letter?
* [An int]
  ~ correct = false
* [A bool]
  ~ correct = false
* [A char]
  ~ correct = true
* [A float]
  ~ correct = false
- -> question\_result
```

Figure 4.1.7 Some snippets of questions available that are related to C++

### 4.2.5 Dialogue System

The dialogue system acts as a key component in managing the dialogue interactions of this game. It interacts with Ink Story Engine to handle the flow of conversations, player choices and UI updates. Overall, the system facilitates the process of entering and exiting dialogues, displaying lines of dialogue, presenting player choices, and responding to player inputs during conversations. This system helps ensure the dialogue flow is controlled and the user can interact with dialogue choices in a smooth way. A code snippet of the dialogue system is as shown below.



```

private void DisplayChoices()
{
    List<Choice> currentChoices = currentStory.currentChoices;

    //Debug check
    if (currentChoices.Count > choices.Length)
    {
        Debug.LogError("More Choices were given than the UI can support. Num of choices given: " + currentChoices.Count);
    }

    int index = 0;
    // enable and initialize the choices up to the amount of choices for this line of dialogue
    foreach (Choice choice in currentChoices)
    {
        choices[index].gameObject.SetActive(true);
        choicesText[index].text = choice.text;
        index++;
    }
    // go thru the remaining choices the UI supports and make sure they are hidden
    for (int i = index; i < choices.Length; i++)
    {
        choices[i].gameObject.SetActive(false);
    }
}

```

Figure 4.1.8 Code Snippet of dialogue system

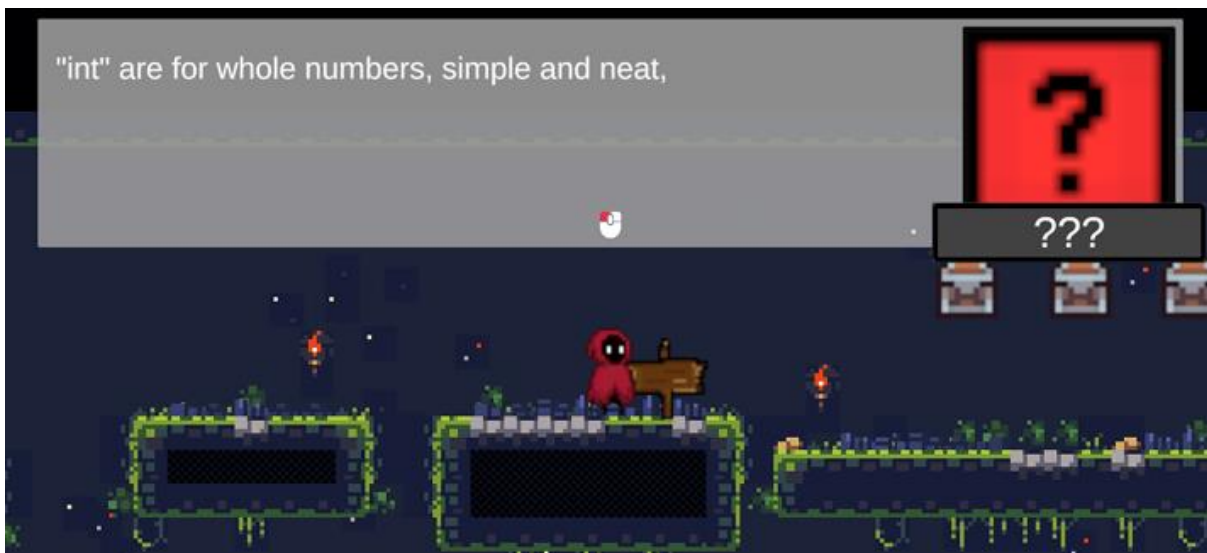


Figure 4.1.9 Dialogue System that is available in game

#### 4.2.6 Boss Script

The **Boss Script** manages the behaviour of the final boss in the game. As the last challenge, the boss tests the player's accumulated knowledge from gameplay through a series of interaction with the dialogue system. It serves as the level's gatekeeper, preventing access to new stages until the player successfully demonstrates sufficient mastery over the game's content. The script not only controls the boss's health and combat mechanics but also integrates seamlessly with the dialogue system, creating a more engaging and fun boss encounter. This combination of combat and dialogue heightens the intensity of the final confrontation, making it a memorable milestone in the game. A code snippet of the boss script are as follows.

Bachelor of Computer Science (Honours)  
 Faculty of Information and Communication Technology (Kampar Campus), UTAR

```

public class Boss : MonoBehaviour
{
    public GameObject projectilePrefab;
    public Transform projectilePos;

    [SerializeField] private int maxHealth = 10;
    private int currentHealth;

    private void Start()
    {
        currentHealth = maxHealth;
        Debug.Log("Boss initialized with " + currentHealth + " health");
    }

    public void TakeDamage(int damage)
    {
        currentHealth -= damage;
        currentHealth = Mathf.Max(currentHealth, 0);
        Debug.Log("Boss took " + damage + " damage. Current health: " + currentHealth);

        if (IsDead())
        {
            Die();
        }
    }
}

```

Figure 4.1.10 Code snippet of boss script

```

private void StartBossFight()
{
    isFightOngoing = true;
    DialogueManager.GetInstance().EnterDialogueMode(inkJSON);
    DialogueManager.GetInstance().OnChoiceMade += HandleChoiceMade;
    Debug.Log("BossDialogue: Boss fight started. Dialogue mode entered.");
}

private void HandleChoiceMade(int choiceIndex)
{
    Debug.Log($"BossDialogue: Choice made. Index: {choiceIndex}");
    StartCoroutine(ProcessAnswerAfterDelay());
}

private IEnumerator ProcessAnswerAfterDelay()
{
    Debug.Log($"BossDialogue: Waiting for {delayAfterAnswer} seconds before processing answer.");
    yield return new WaitForSeconds(delayAfterAnswer);

    bool correct = (bool)DialogueManager.GetInstance().GetVariableState("correct");
    Debug.Log($"BossDialogue: Answer was {(correct ? "correct" : "incorrect")}");
}

```

Figure 4.1.11 Code snippet of boss dialogue script that works together with Dialogue System.

### 4.2.7 Enemy Script

Unlike the boss script that handles complex behaviours, the enemy script handles a standard opponent interaction. In this game, the normal enemy is a slime. Its script has a basic patrol between two selected points. Besides that, it will also attack the player if it is in range of a player. A code snippet is as shown below.

```
private void MoveInDirection(int direction)
{
    idleTimer = 0;

    anim.SetBool("isWalking", true);

    //Makes enemy face a specific direction
    enemy.localScale = new Vector3(Mathf.Abs(initScale.x) * direction, initScale.y, initScale.z);

    //Makes enemy move in the specific direction
    enemy.position = new Vector3(enemy.position.x + Time.deltaTime * direction * speed,
        enemy.position.y, enemy.position.z);
}
```

Figure 4.1.12 A code snippet of the general enemy in game

### 4.2.8 Platform

The platform component is crucial in representing the game level design elements. It includes various types of platforms that the player can interact with, potentially incorporating moving or breakable platforms that add complexity to the gameplay. There is also a temporary door that acts as a platform till the player finds a key to unlock it.



Figure 4.1.13 A floating platform with waypoints.

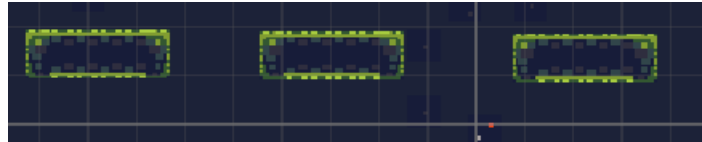


Figure 4.1.14 A breakable platform that looks like a normal platform

#### 4.2.9 Game Object

The game object is a type of component that represents the interactive elements within the game world. It includes floating chests, a key and a plaque in the game. These game objects serve an important role in terms of gameplay and educational aspects.

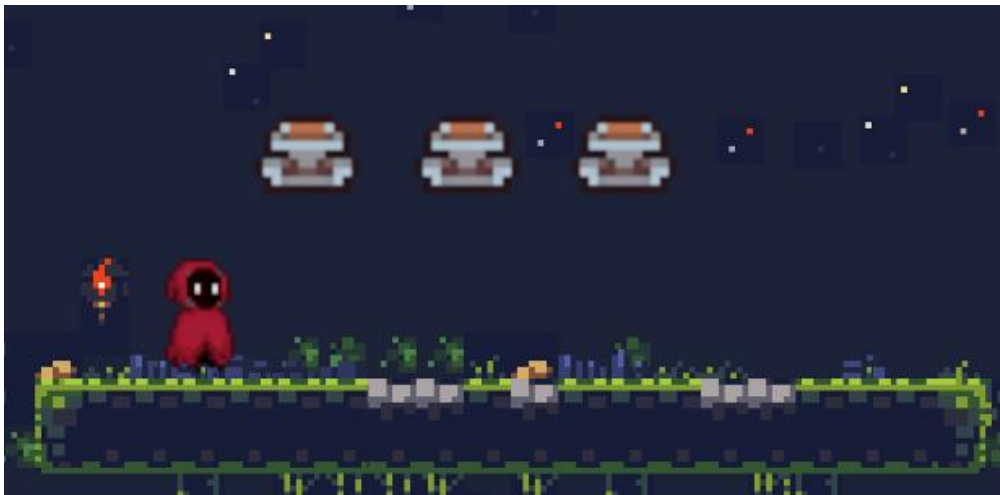


Figure 4.1.15 Floating chests that are available in the game world.

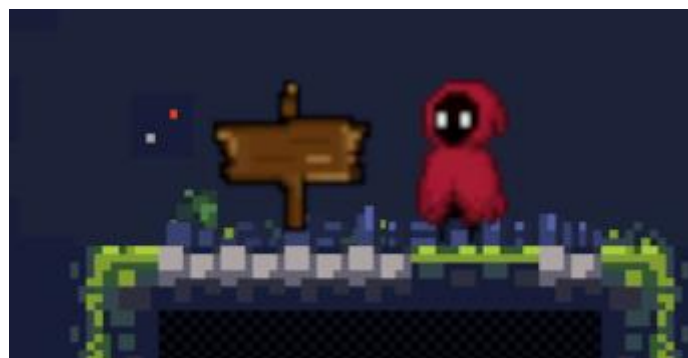


Figure 4.1.16 A wooden plaque that provide hints for the player

#### **4.2.10 Game**

The Game component represents the final output of all these interconnected systems. It's the playable game that the user experiences, combining all the elements managed by the Core Game Engine into a cohesive, educational, and entertaining C++ learning experience.

# Chapter 5

## System Implementation

### 5.1 Hardware Specifications

This project was developed using a laptop with the specifications shown in Table 5.1.1. The laptop was used to store the scripts and the files of the game. All of the software used were installed on the laptop.

Description	Specification
Model	ROG Strix G513QM
Processor	AMD Ryzen 9 5900HX with Radeon Graphics, 3.30 GHz
Operating System	Windows 11 Home Single Language
Graphics	NVIDIA GeForce RTX 3060 AMD Radeon Graphics
Memory	32GB RAM
Storage	ADATA SX6000LNP 512GB HFM 512 GD3JX013N 512GB

Table 5.1.1 Specifications of Laptop

### 5.2 Software Specifications

The software involved in the process of developing this project are listed below as follows:

1. Visual Studio 2022 Enterprise Edition (v17.9)
2. Unity Hub 3.7.0
3. Unity Editor Version 2022.3.5f1
4. LDtk 1.5.3
5. Inky Editor

The project is managed using Unity Hub 3.7.0, with Unity Editor Version 2022.3.5f1 serving as the core platform for game development. Visual Studio 2022 Enterprise Edition (v17.9) was used for writing and debugging scripts, ensuring smooth integration and functionality across the game's mechanics. LDtk 1.5.3 was utilized for creating intricate level designs, allowing for a well-structured and visually compelling game world. Additionally, Inky Editor played a pivotal role in crafting the narrative, where story scripts were written and seamlessly integrated into Unity, driving the interactive storytelling elements of the game.

### 5.3 Setting and Configuration

Several configurations must be carried out in order to smoothly develop the system. Firstly, the LDtk software must be connected to the Unity software. To do that, a plugin must be installed in the Unity software to properly connect them together. In unity, head to Edit > Project Settings > Package Manager and create a new scoped registry as follows:

- Name : OpenUPM
- URL : <https://package.openupm.com>
- Scope(s) : com.cammin.ldtkunity

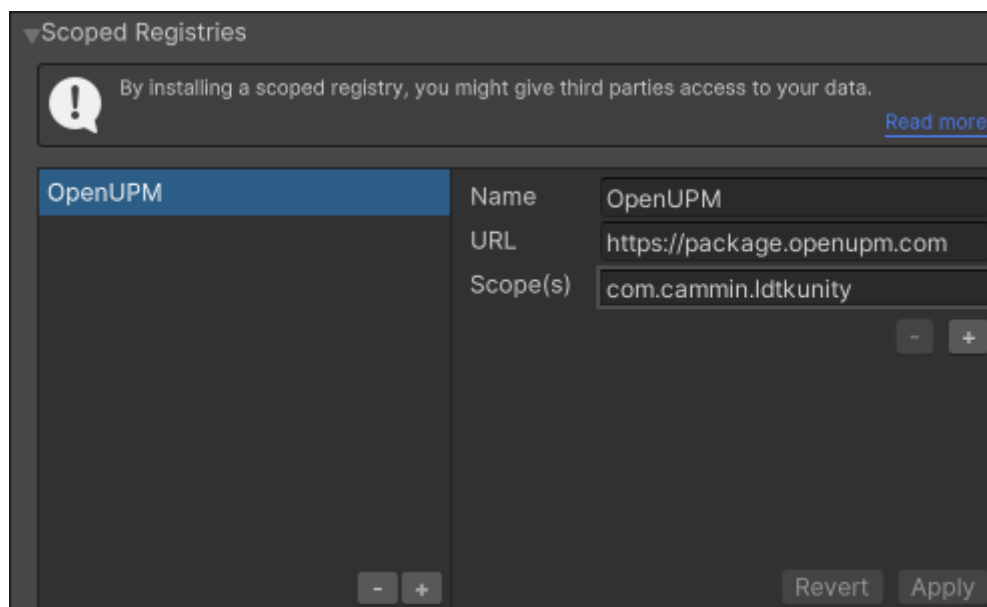


Figure 5.3.1 Creating a new scoped registry

Next, head to Unity Package Manager window to select My Registries and select install.

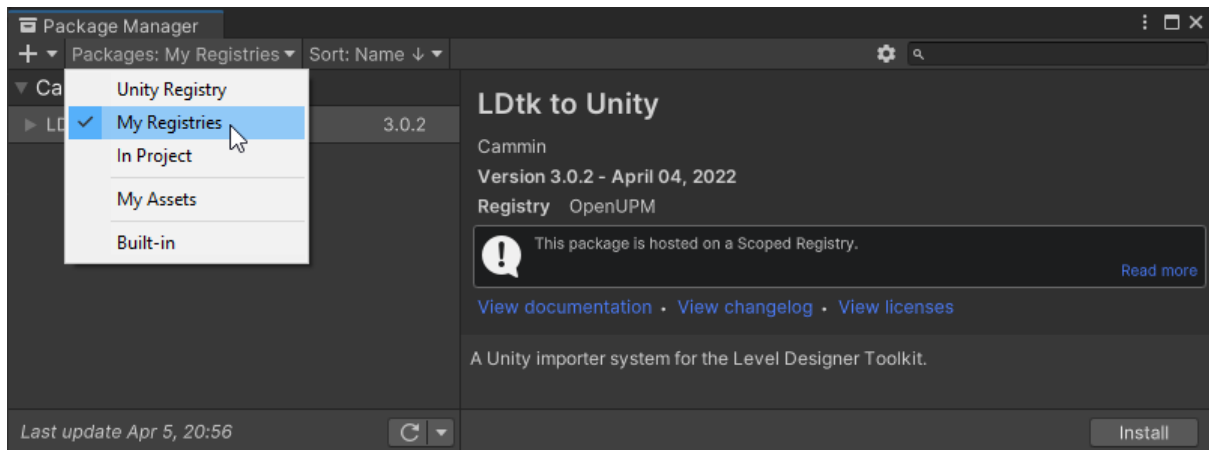


Figure 5.3.2 Installing the necessary plugins

With this, the connection between LDtk software and Unity is created. Any floor maps created in LDtk can be easily exported to Unity software easily and any changes made in LDtk software can be reflected quickly in Unity.

## 5.4 System Operation

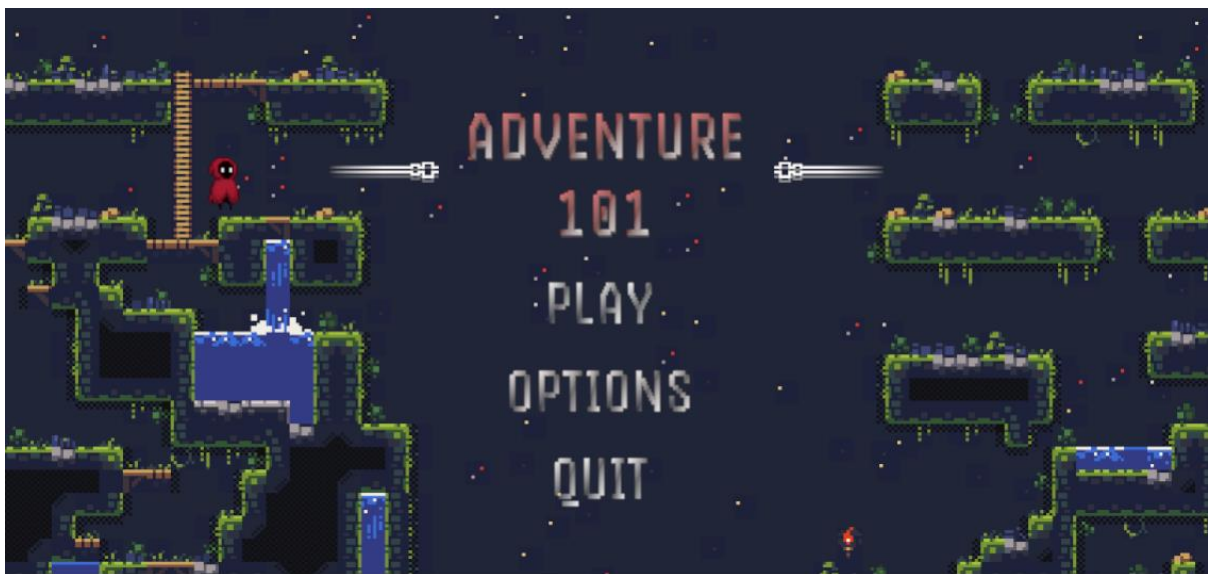


Figure 5.4.1 Main Menu screen for the game

Figure 5.4.1 shows the Main Menu page for the game. The main menu page is the default page that will show up when the game is first started. There are 3 options available for the user to choose, which is Play, Options, and Quit. Users can select which one that they wanted to proceed with it.



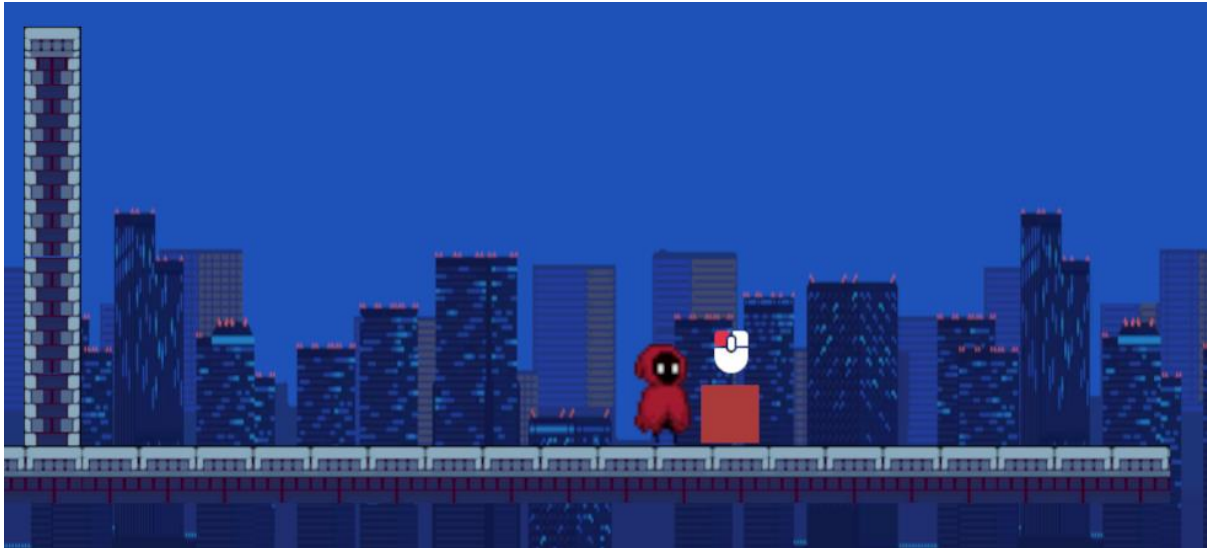


Figure 5.4.2 First Floor of the game

Upon entering the game, users will come into contact with a Non Playable Character (NPC). From the NPC, users are able to learn the backstory of the game before proceeding. The first floor serves as an introduction to the game with various NPCs available. After talking to them, the users can then enter the new level.



Figure 5.4.3 Second Floor of the game

After entering the second floor, users can first approach the NPC to know more about the backstory and floor type. After that, users can head over to explore the entire floor.

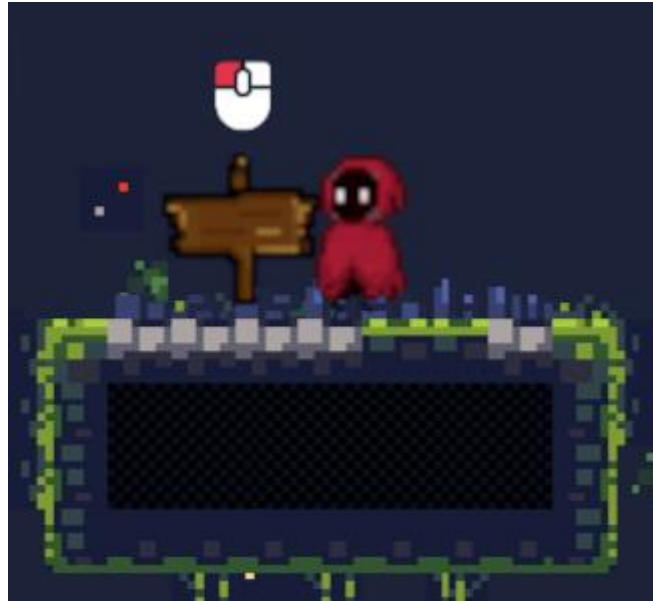


Figure 5.4.4 Wooden plaque that guides players

Players can interact with the wooden plaque available to learn more about the challenges ahead. The wooden plaque guides the player with the basics of data types. This information is useful for players as they will know need the knowledge to face the first puzzle.

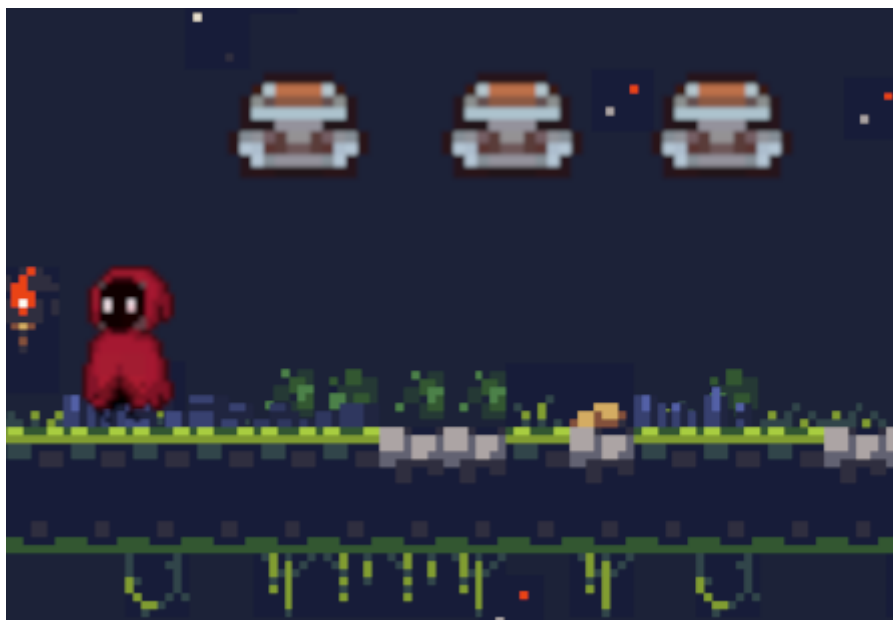


Figure 5.4.5 Floating chests puzzle after the wooden plaque

The floating puzzles is the first puzzle that the users will face. Users will need to have a basic understanding of the various data types to get the correct answer. There will be values that

appear above the chests for the player to choose. The player that makes a wrong decision will lose one point of health if they select the wrong answer.

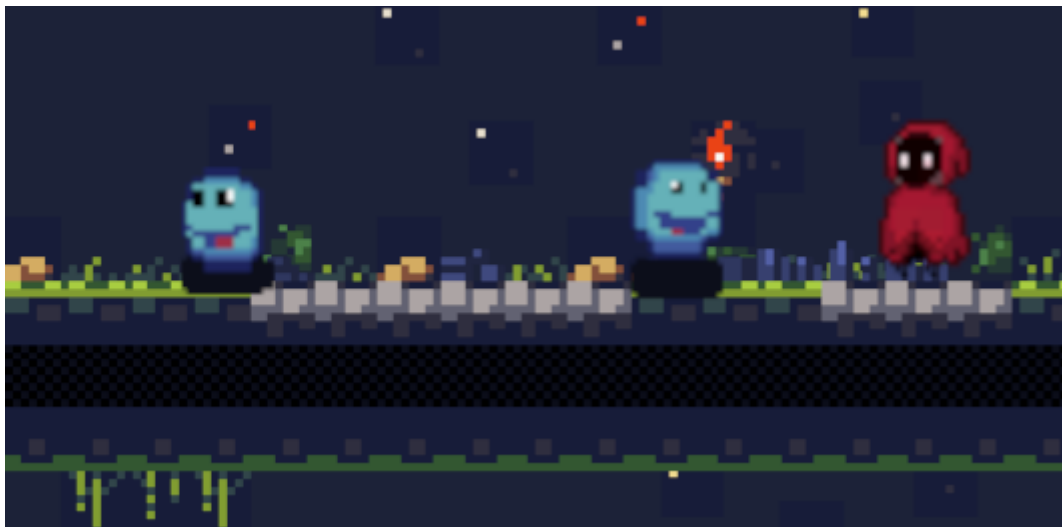


Figure 5.4.6 Encounter with slimes as a challenge

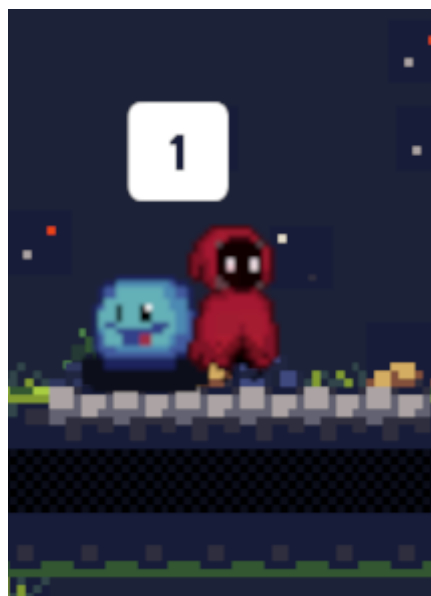


Figure 5.4.7 A value that appears above the slime

The slimes will have a value above them just like the floating chests. Users need to hit the correct slime for the right answer. Conversely, if they hit the wrong slime, they will lose one point of health.



Figure 5.4.8: Breakable platforms with answers

The breakable platform has answers that allow users to choose. From the wooden plaque, users are given a question. They will have to choose the correct platform for their answer. If they select the wrong platform, they will lose one point of health.

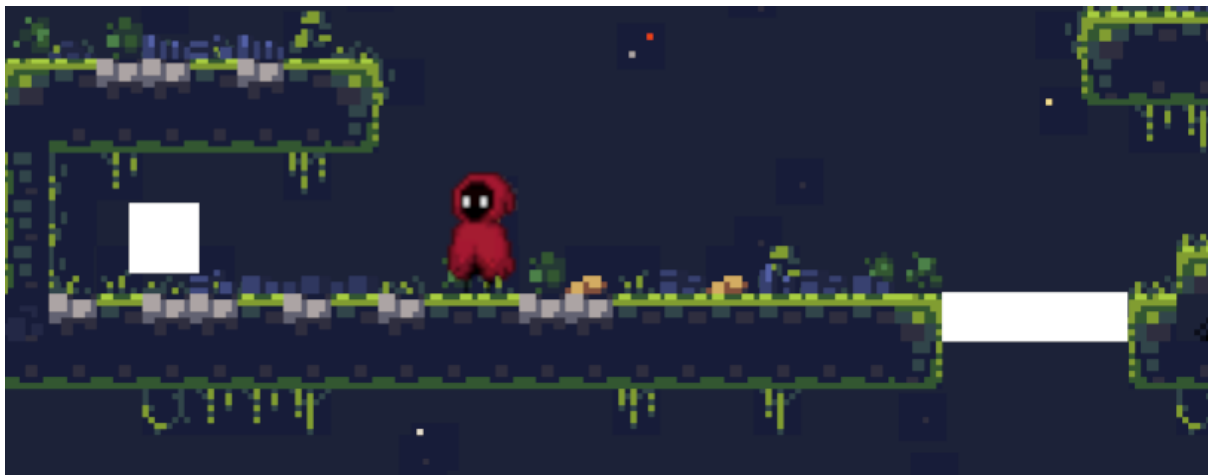


Figure 5.4.9 Lock and key system

The lock and key system will unlock the players to the final boss of the level. By obtaining the key, the lock will automatically open to allow players to enter the boss arena.



Figure 5.4.10 Final boss of the level

The final boss of the level can be found in the deepest part of the level. The players need to answer all the boss questions correctly to win. The player will deal one point of damage to the boss if they answer correctly. The player will only have 3 chances or less depending on their health points. After beating the boss, the player is free to explore the new level.

### 5.5 Implementation Issues and Challenges

When implementing the system, various issues were faced. One major issue that remains till now is the current unity version. Unity version 2022.3.5fl has a major bug that involves in the inability to modify or customize physics shapes. This proved to be a huge problem when faced during development. Several attempts were done to solve it, such as upgrading to the latest version of the Unity but it all ended in failure due to incompatibility. As a result, some of the game elements had to be removed to proceed.

Furthermore, there are problems with integrating the quiz function to the boss script. The boss health would not drop even if the player has managed to answer all questions. The same goes for the player health too when they fail to answer the questions correctly. Fortunately, the problem has been solved.

Lastly, there are not many good guides on various types of puzzles in Unity. A lot of creativity is needed to produce new puzzles. Furthermore, a certain amount of knowledge is also required in order to create new puzzles. The way to solve it is to populate the level with same puzzles but with different question types attached to it.

## **5.6 Concluding Remarks**

Overall, the creation of this game is made easy due to the Internet having sufficient and abundant tutorials that provide step by step guides. Most of the items needed such as animation sprites are readily available in the Internet to be used. The settings and configurations made in this game is necessary as it directly affects the overall gameplay. Most importantly, most of the problems faced along the way are resolved. In short, the implementation is a success.

# Chapter 6

## System Evaluation and Discussion

### 6.1 System Testing and Performance Metrics

After the game is implemented, testing is needed to be carried out to make sure that the game produced is of quality. There are no errors with the basic movements such as moving, jumping and attacking. Furthermore, there are no noticeable lagging or unintended delays. All animation transitions are smooth without any errors and lastly the collision between game objects and platforms has no problems.

As for the main content which is C++ learning, all educational content integration into the game has no noticeable error. The quizzes, puzzles and challenges are all at appropriate locations and sufficient in numbers. The knowledge that is dispersed are also sufficient for target users aged from 12 years old and above. The questions created are accurate and no errors are recorded.

### 6.2 System Testing and result

Table 6.1.1 Start Game Test Case

<b>Start Game Test Case</b>				
<b>Role: User</b>				
<b>No.</b>	<b>Test Case</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Pass/Fail</b>
1	User presses "Start"	The game loads the first level	The game loads the first level	Pass
2	User presses "Options"	The game loads the settings	The game loads the settings	Pass
3	User presses "Exit"	The game exits	The game exits	Pass

Table 6.1.2 Move Character Test Case

<b>Start Game Test Case</b>				
<b>Role: User</b>				
<b>No.</b>	<b>Test Case</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Pass/Fail</b>

1	User presses “A”	The in game character moves to left	The in game character moves to left	Pass
2	User presses “D”	The in game character moves to left	The in game character moves to right	Pass
3	User presses “E” when they are near to a game object	The in game character interacts with a Game Object	The in game character interacts with a Game Object	Pass
4	User presses Left button of mouse	The in game character attacks.	The in game character attacks.	Pass
5	User presses “Spacebar”	The in game character jumps.	The in game character jumps.	Pass
6	User presses “Spacebar” when the in game character is already jumping	The in game character does a double jump	The in game character does a double jump	Pass
7	User presses Left button of mouse when they are near to a game object	The in game character interacts with a Game Object	The in game character interacts with a Game Object	Pass

Table 6.1.3 Interact with Game Objects Test Case

<b>Interact with Game Objects Test Case</b>				
<b>Role: User</b>				
<b>No.</b>	<b>Test Case</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Pass/Fail</b>
1	User presses “E” when they are near to a game object	The in game character interacts with a Game Object	The in game character interacts with a Game Object	Pass



2	User presses Left button of mouse when they are near to a game object	The in game character interacts with a Game Object	The in game character interacts with a Game Object	Pass
---	---	--	--	------

Table 6.1.4 In game character health Test Case

<b>In game character health Test Case</b>				
<b>Role: User</b>				
<b>No.</b>	<b>Test Case</b>	<b>Expected Result</b>	<b>Actual Result</b>	<b>Pass/Fail</b>
1	In game character selects a wrong answer	In game character loses one point of health	In game character loses one point of health	Pass
2	In game character selects a correct answer	In game character does not lose one point of health	In game character does not lose one point of health	Pass

### 6.3 Project Challenges

One significant challenge encountered during the development of this project was the limited availability of open-source games for reference. Without extensive resources to study from a developer's perspective, much of the game's creation relied on observations and insights drawn from a player's point of view. This approach required reverse-engineering gameplay mechanics and design elements seen in other titles, adapting them to fit the unique goals of the project.

### 6.4 Objective Evaluation

The goals of this project have been achieved successfully, guaranteeing a smooth integration of technology learning and fun. The software effectively teaches users the basics of programming with engaging gameplay and C++ coding tasks. Utilizing C++ as the core medium, the game offers a strong base for grasping programming principles that can be used in different languages, thus fulfilling the educational goal.

The simple and straightforward interface of the 2D platform design effectively provides a minimalist yet captivating experience. Users have the opportunity to immerse themselves in a visually attractive setting while still being able to tackle and overcome programming

challenges effectively. The game has effectively balanced an engaging storyline with challenging coding tasks, guaranteeing that it is both fun and educational.

The software has been fine-tuned after undergoing thorough testing to ensure top performance, reliability, and user-friendliness, resulting in a seamless and enjoyable user experience. The game surpasses expectations by providing an enjoyable and educational tool that promotes coding knowledge through gamification, all set for market launch.

### **6.5 Concluding Remarks**

This project has successfully developed an educational game that allows young users to learn the basics of C++ while having fun in game. All objectives stated has been met with great success in the end. From the testing of the game, it is working as intended. Overall, this project is able to achieve its end goal which is to provide fun to users while teaching them.

## 7.0

# Conclusion and Recommendation

### 7.1 Conclusion

Concluding the development journey of an educational coding game within Unity has been a journey marked by both challenges and achievements. The integration of cutting-edge game design principles with fundamental coding concepts has paved the way for a truly transformative learning experience. Through the strategic placement and application of gamification elements such as the inclusion of challenges, rewards, and interactive scenarios, the project aims to revolutionize the accessibility and effectiveness of coding education, particularly for young learners.

While considerable strides have been made in shaping the game's narrative and implementing interactive features, the path has not been without its obstacles. However, these challenges have presented invaluable opportunities for growth and refinement. By addressing these considerations, the project stands to fully realize its potential, which is to empower a new generation of learners to embrace technology and shape a limitless future.

In conclusion, this project aims to hold the promise of igniting curiosity, nurturing creativity, and instilling essential coding skills across the younger generation.

### 7.2 Recommendation

To further improve the project, one key recommendation is to expand the game by adding more levels that introduce additional coding concepts. These new levels could progressively cover a wider range of topics, offering young users a more comprehensive learning experience and deepening their understanding of programming.

Another important suggestion is to incorporate multiple programming languages into the game. By introducing languages beyond C++, such as Python, Java, or JavaScript, users would gain exposure to a broader spectrum of coding skills. As players advance and master the fundamentals, the game's difficulty could be adjusted accordingly, presenting more challenging puzzles and tasks tailored to each new language. This would not only sustain

engagement but also ensure that users develop proficiency in a variety of programming environments, better preparing them for real-world coding challenges.

A final recommendation is to implement a multiplayer or collaborative mode where users can team up to solve coding challenges together. This would foster collaboration, problem-solving, and teamwork, allowing players to learn not only from the game but also from their peers. It could also simulate real-world coding environments where collaboration and communication are essential skills.

# References

- [1] Nandanwar(Khadgi), J. (2022) *Roblox Programming: A tool to learn coding*, *eLearning Industry*. Available at: <https://elearningindustry.com/roblox-programming-a-tool-to-learn-coding>
- [2] Ahmadov, O. (2022) *Why coding is important for kids*, *eLearning Industry*. Available at: <https://elearningindustry.com/why-coding-is-important-for-kids>
- [3] Johnson (Ed.D), R. (2023) *Gamification in Elearning: Engaging learners through game elements*, *eLearning Industry*. Available at: <https://elearningindustry.com/gamification-in-elearning-engaging-learners-through-game-elements>
- [4] Penzilla (2022) *Hooded protagonist animated character by penzilla*, *itch.io*. Available at: <https://penzilla.itch.io/hooded-protagonist>
- [5] Inkle (no date a) *Ink/documentation/writingwithink.md at master · Inkle/Ink, GitHub*. Available at: <https://github.com/inkle/ink/blob/master/Documentation/WritingWithInk.md>

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year: Y3S3</b>	<b>Study week no.:2</b>
<b>Student Name &amp; ID: Ong Zhan Wen, 21ACB04076</b>	
<b>Supervisor: Mr. Luke Lee Chee Chien</b>	
<b>Project Title: Interactive Learning Adventures: Learn Coding Through Gamification</b>	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Fine tuned the box puzzle

## 2. WORK TO BE DONE

Integrate new puzzles into the level

## 3. PROBLEMS ENCOUNTERED

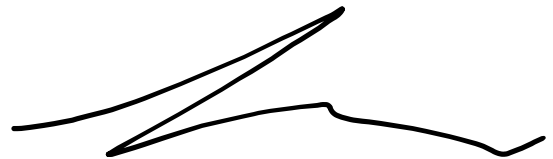
Difficulty in choosing suitable puzzles and challenges that allows combination of game elements and syllabus

## 4. SELF EVALUATION OF THE PROGRESS

Need more thorough research on different puzzles and challenges available



Supervisor's signature



Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year: Y3S3</b>	<b>Study week no.:4</b>
<b>Student Name &amp; ID: Ong Zhan Wen, 21ACB04076</b>	
<b>Supervisor: Mr. Luke Lee Chee Chien</b>	
<b>Project Title: Interactive Learning Adventures: Learn Coding Through Gamification</b>	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Created enemy sprites to allow interaction between player and enemy

## 2. WORK TO BE DONE

Integrate syllabus into the enemy gameplay

## 3. PROBLEMS ENCOUNTERED

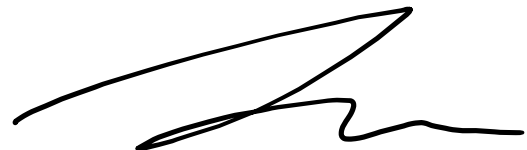
Trouble having enemy game object to patrol from one point to another

## 4. SELF EVALUATION OF THE PROGRESS

Check the enemy code functions more thoroughly



Supervisor's signature



Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year: Y3S3</b>	<b>Study week no.:6</b>
<b>Student Name &amp; ID: Ong Zhan Wen, 21ACB04076</b>	
<b>Supervisor: Mr. Luke Lee Chee Chien</b>	
<b>Project Title: Interactive Learning Adventures: Learn Coding Through Gamification</b>	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Fine tuned the enemy game object, created breakable platforms for puzzle.

## 2. WORK TO BE DONE

Need to integrate syllabus into breakable platforms gameplay

## 3. PROBLEMS ENCOUNTERED

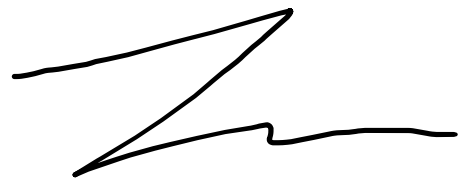
No problems were encountered during current phase

## 4. SELF EVALUATION OF THE PROGRESS

Keep up with the schedule



Supervisor's signature



Student's signature



# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year: Y3S3</b>	<b>Study week no.:8</b>
<b>Student Name &amp; ID: Ong Zhan Wen, 21ACB04076</b>	
<b>Supervisor: Mr. Luke Lee Chee Chien</b>	
<b>Project Title: Interactive Learning Adventures: Learn Coding Through Gamification</b>	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Created lock and key system, integrated final boss game object.

## 2. WORK TO BE DONE

Finish up the boss gameplay.

## 3. PROBLEMS ENCOUNTERED

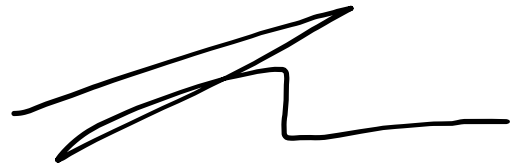
Boss does not perform certain functions when triggered.

## 4. SELF EVALUATION OF THE PROGRESS

Need to seek advice on the problem encountered.



Supervisor's signature



Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year: Y3S3</b>	<b>Study week no.:10</b>
<b>Student Name &amp; ID: Ong Zhan Wen, 21ACB04076</b>	
<b>Supervisor: Mr. Luke Lee Chee Chien</b>	
<b>Project Title: Interactive Learning Adventures: Learn Coding Through Gamification</b>	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Created the user interface for the game.

## 2. WORK TO BE DONE

Fine tune the user interface and write report

## 3. PROBLEMS ENCOUNTERED

No problems were encountered during current phase

## 4. SELF EVALUATION OF THE PROGRESS

No evaluation needed during current phase



Supervisor's signature



Student's signature

# FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

<b>Trimester, Year: Y3S3</b>	<b>Study week no.:12</b>
<b>Student Name &amp; ID: Ong Zhan Wen, 21ACB04076</b>	
<b>Supervisor: Mr. Luke Lee Chee Chien</b>	
<b>Project Title: Interactive Learning Adventures: Learn Coding Through Gamification</b>	

## 1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Completed user interface and boss game object

## 2. WORK TO BE DONE

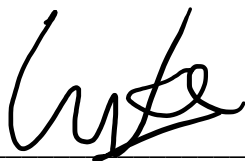
Write report

## 3. PROBLEMS ENCOUNTERED

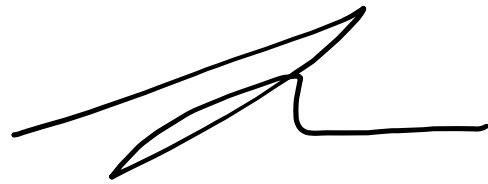
No problems were encountered during current phase

## 4. SELF EVALUATION OF THE PROGRESS

No evaluation needed during current phase



Supervisor's signature



Student's signature

## POSTER

# ADVENTURE 101

## LEARN CODING THROUGH GAMING

### INTRODUCTION



Learn the basics of C++ while exploring the open world! Explore the world of Viltres, the virtual plane along with Heyght, the main character as while learning the basics of C++ through challenges and puzzle solving!

### MAIN FEATURES

- ★ Boost learning effectiveness
- ★ Improves problem solving skills
- ★ In game scenarios enhances interactivity

### DISCUSSION

C++ serves as a good starting point for novice or beginners. With knowledge of C++, children may be able to pick up other languages quickly



START GAME ►

PROJECT DEVELOPER: ONG ZHAN WEN  
SUPERVISED BY: MR LUKE LEE CHEE CHIEN

## PLAGIARISM CHECK RESULT

FYP2 2104076.docx

### ORIGINALITY REPORT

<b>5%</b>	<b>5%</b>	<b>1%</b>	<b>%</b>
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

### PRIMARY SOURCES

<b>1</b>	<b>eprints.utar.edu.my</b> Internet Source	<b>2%</b>
<b>2</b>	<b>www.freepressjournal.in</b> Internet Source	<b>1%</b>
<b>3</b>	<b>www.softwaretestinghelp.com</b> Internet Source	<b>&lt;1%</b>
<b>4</b>	<b>github.com</b> Internet Source	<b>&lt;1%</b>
<b>5</b>	<b>123dok.com</b> Internet Source	<b>&lt;1%</b>
<b>6</b>	<b>fict.utar.edu.my</b> Internet Source	<b>&lt;1%</b>
<b>7</b>	<b>wiingy.com</b> Internet Source	<b>&lt;1%</b>
<b>8</b>	<b>Yingyu Xu. "Deep reinforcement learning and imitation learning based on VizDoom", Proceedings of the 2022 6th International Conference on Electronic Information Technology and Computer Engineering, 2022</b> Publication	<b>&lt;1%</b>

9	<a href="http://publisher.uthm.edu.my">publisher.uthm.edu.my</a> Internet Source	<1 %
10	<a href="http://setkab.go.id">setkab.go.id</a> Internet Source	<1 %
11	<a href="http://forums.tomshardware.com">forums.tomshardware.com</a> Internet Source	<1 %
12	<a href="http://www.slideshare.net">www.slideshare.net</a> Internet Source	<1 %
13	<a href="http://www.dell.com">www.dell.com</a> Internet Source	<1 %
14	Yoshiki Ohira, Takahisa Uchida, Takashi Minato, Hiroshi Ishiguro. "A Dialogue System That Models User Opinions Based on Information Content", Multimodal Technologies and Interaction, 2022 Publication	<1 %

Exclude quotes Off  
Exclude bibliography On

Exclude matches Off

**Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)**

Form Number: FM-IAD-005 | Rev No.: 0 | Effective Date: | Page No.: 1 of 1



**FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY**

<b>Full Name(s) of Candidate(s)</b>	Ong Zhan Wen
<b>ID Number(s)</b>	2104076
<b>Programme / Course</b>	CS
<b>Title of Final Year Project</b>	Interactive Learning Adventures: Learn Coding Through Gamification

<b>Similarity</b>	<b>Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)</b>
<b>Overall similarity index: <u>5</u></b>  <b>% Similarity by source</b> Internet Sources: <u>5</u> % Publications: <u>1</u> % Student Papers: <u>0</u> %	
<b>Number of individual sources listed of more than 3% similarity: 0</b>	
<b>Parameters of originality required, and limits approved by UTAR are as Follows:</b> (i) Overall similarity index is 20% and below, and (ii) Matching of individual sources listed must be less than 3% each, and (iii) Matching texts in continuous block must not exceed 8 words <i>Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.</i>	

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

*Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.*

Signature of Supervisor

Name: Luke Lee Chee Chien

Date: 13/09/2024

Signature of Co-Supervisor

Name: \_\_\_\_\_

Date: \_\_\_\_\_



**UNIVERSITI TUNKU ABDUL RAHMAN**

**FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY  
(KAMPAR CAMPUS)**

**CHECKLIST FOR FYP2 THESIS SUBMISSION**

Student Id	21ACB04076
Student Name	Ong Zhan Wen
Supervisor Name	Mr Luke Lee Chee Chien

<b>TICK (√)</b>	<b>DOCUMENT ITEMS</b>
	Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

\*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

(Signature of Student)

Date: 13/09/2024