

## SCHOOL OF ENGINEERING AND TECHNOLOGY

FINAL ASSESSMENT FOR THE BSC (HONS) INFORMATION TECHNOLOGY; BSC (HONS) COMPUTER SCIENCE; BACHELOR of SOFTWARE ENGINEERING (HONS) YEAR 2

ACADEMIC SESSION 2024; SEMESTER 4

PRG2104: OBJECT ORIENTED PROGRAMMING

Project

DEADLINE: Week 14

### INSTRUCTIONS TO CANDIDATES

- This assignment will contribute 50% to your final grade.
- This is an individual assignment.


### IMPORTANT

The University requires students to adhere to submission deadlines for any form of assessment. Penalties are applied in relation to unauthorized late submission of work.

Coursework submitted after the deadline will be awarded 0 marks

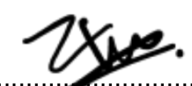
### Lecturer's Remark (Use additional sheet if required)

I Lim Zi Xuan (Name) 22031645 (Student ID) received the assignment and read the comments

  
..... (Signature/date)

### Academic Honesty Acknowledgement

"I Lim Zi Xuan (student name). verify that this paper contains entirely my own work. I have not consulted with any outside person or materials other than what was specified (an interviewee, for example) in the assignment or the syllabus requirements. Further, I have not copied or inadvertently copied ideas, sentences, or paragraphs from another student. I realize the penalties (*refer to page 16, 5.5, Appendix 2, page 44 of the student handbook diploma and undergraduate programme*) for any kind of copying or collaboration on any assignment."

  
..... (Student's signature / Date)

# Table of Contents

1.0 Introduction.....	4
1.1 Game Overview .....	4
1.2 Motivation of the Project .....	5
2.0 UML Diagrams .....	6
2.1 Class Diagram .....	6
2.2 Application Flow Chart.....	7
3.0 Code Structure Overview .....	8
4.0 System Functionalities .....	11
4.1 Home Page .....	11
4.1.1 Login Section .....	12
4.1.2 Difficulty Levels Selection .....	12
4.1.3 Navigation Buttons .....	13
4.2 Leaderboard Page.....	13
4.2.1 Filter Options .....	14
4.2.2 Top Player Labels .....	14
4.2.3 Detailed Results Table .....	15
4.3 Rules and Frequently Asked Questions (FAQs) Page .....	17
4.3.1 Game Rules .....	17
4.3.2 Frequently Asked Questions (FAQs) .....	17
4.3.3 Return Home Button.....	17
4.4 Game Pages (Easy, Intermediate, Hard Levels).....	18
4.4.1 Game Grid .....	19
4.4.2 Flip Card .....	20
4.4.3 Visualization of Game Status.....	20
4.5 Pause Game Function .....	21
4.6 Return Home Function.....	22
4.7 Game Over Dialog Box .....	22
4.8 Volume Button Control .....	22
4.9 Database Visualization with DataGrip .....	23
5.0 Personal Reflection .....	26
5.1 Applications of OOP Concepts .....	26
5.1.1 Inheritance .....	26

5.1.2 Parametric Polymorphism & Generic Programming .....	27
5.1.3 Collections .....	27
5.2 Challenges and Solutions .....	28
5.3 Project Strengths .....	28
5.4 Weaknesses & Areas of Improvement .....	28
6.0 Conclusion .....	29
7.0 References .....	30
7.1 Concept References .....	30
7.2 Background Music & Sound Effect Resources .....	30
7.3 Image Resources .....	30

**Video Presentation YouTube Link:** <https://www.youtube.com/watch?v=-D-FL23MTZA>

**GitHub Profile Link:** <https://github.com/limm-21>

**Enrolled GitHub Assignment Link:** <https://github.com/sunwaydcis/finalprojectv2-limm-21>

# 1.0 Introduction

## 1.1 Game Overview

The Memory Card Game is a single-player puzzle game developed using Scala 2.12.19 Library, ScalaFX 8 and Java 8 Library. The primary goal of the establishment of the game is to test and improve the player's memory by having them match pairs of cards within the least amount of moves and time. The game features three levels of difficulty which are easy, intermediate, and hard, each with an increasing number of cards to match. It allows players to track their performance after the game so as to improve their competency. Below are the key features or functionalities included in the Memory Card Game:

- i) Home Page to Track Usernames and Difficulty Levels
  - Serves as the beginning of the game where a player can navigate to the Leaderboard, Rules & FAQ pages or to start a new game, and to store player's username and difficulty levels.
- ii) Leaderboard to Store and Retrieve Players' Game Results
  - It shows the top 15 game results for each difficulty level with the least moves and time taken at the top of the rank.
- iii) Game Pages for Handling Flipping Cards and Match Checking Operations
  - Displays game grid according to various difficulty levels and enables card-related operations such as revealing and hiding cards, checking for match and updating game status.
- iv) Pause Game Function in the Game
  - Enables a player to pause ongoing game activities and later resume from where they stopped.
- v) Return Home Function in the Game
  - This allows players to leave the current game and go back to the home page. A confirmation dialog should be used to avoid any accidental taps that would lead to leaving.

vi) Handling Game Over Operations

- Displays a congratulatory message upon completing the game, showing the time taken and the number of moves.

vii) Enabling Background Sound Effects

- The game also includes background music with an option to turn it on or off, adding an engaging auditory experience for players.

## 1.2 Motivation of the Project

The development of the Memory Card Game was driven by several motivations. First and foremost, it mainly targets entertainment, while giving out an educational value by using the application to exercise and strengthen individual's memory and cognitive abilities. Users' short-term memory and concentration can be enhanced through the game, given that the game has to be played severally by the users. Furthermore, the project serves as an excellent opportunity to gain insightful technical knowledge in a way that the knowledge of Scala, ScalaFX and Java have been expanded and strengthened. It shows a clear overview on how to construct a complete and working application for the operating system desktop environment with an attractive Graphical User Interface (GUI). Another key motivation is to create an engaging and visually appealing game that incorporates a pleasant user experience. This involves designing interactive interfaces, smooth animations, and integrating sound effects so as to enhance player enjoyment. Moreover, the game is developed to be both educational and entertaining to provide players with a relaxing and enjoyable way to spend their time while improving their memory skills.

## 2.0 UML Diagrams

### 2.1 Class Diagram

Attached shows the full class diagram to show the relationship exists among all classes present in the code:

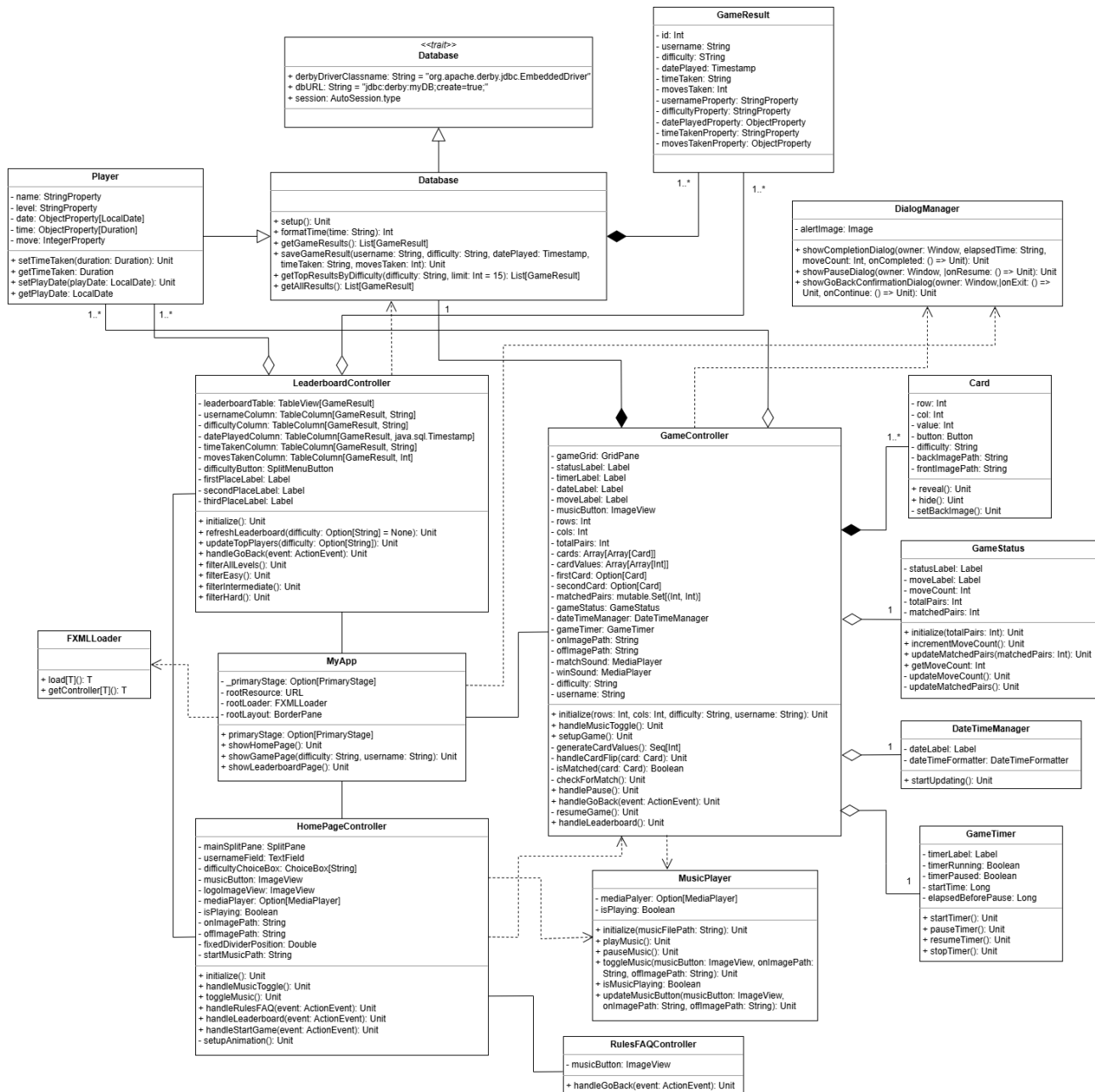


Figure 2.1.1 Complete Class Diagram

\*Complete Class Diagram in PDF form is attached in submitted zip file for better visualization

## 2.2 Application Flow Chart

Attached shows the complete flow chart on the flow and logic when playing the Memory Card Game:

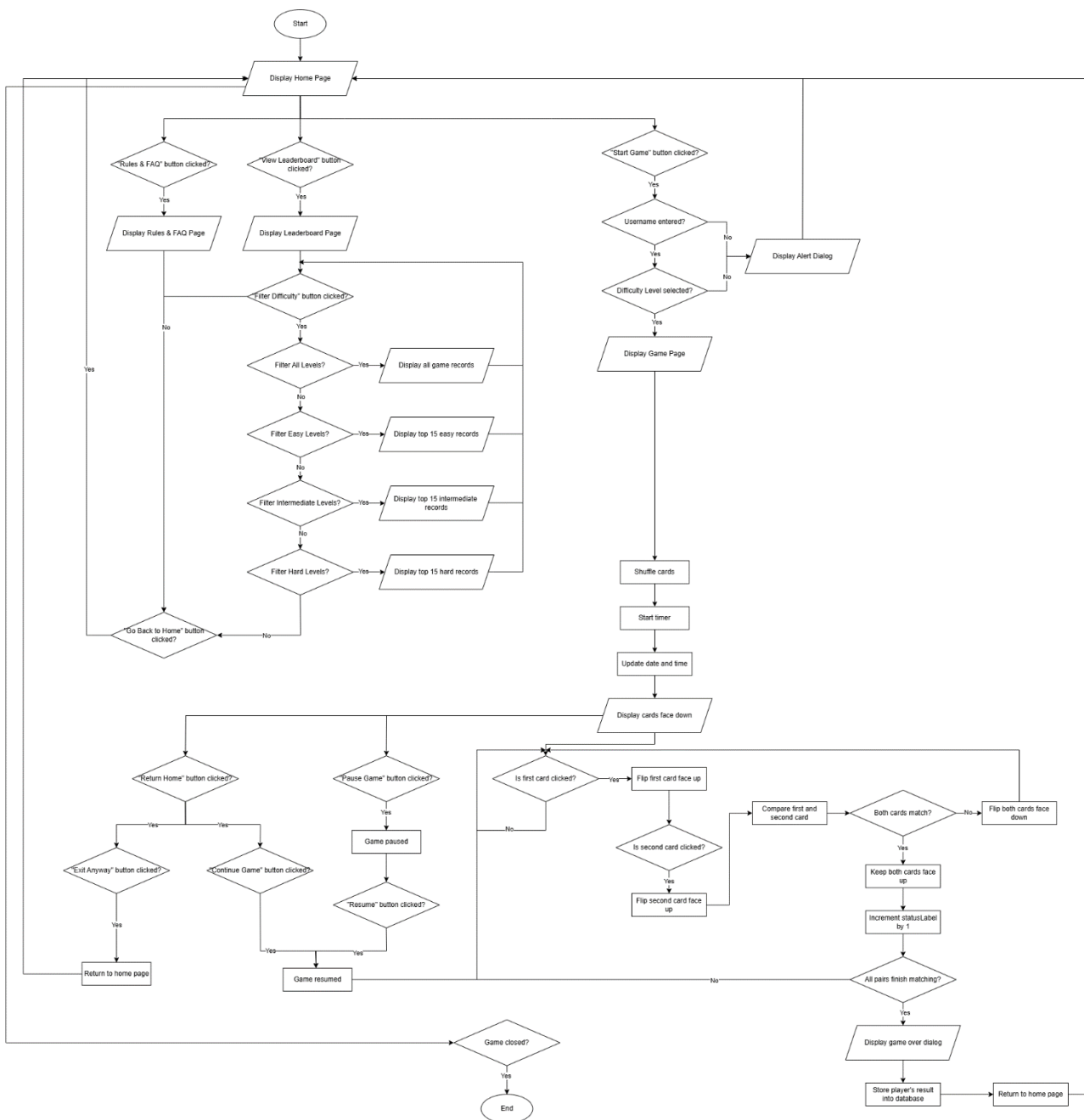


Figure 2.2.1 Complete Flow Chart

*\*Complete Flow Chart in PDF form is attached in submitted zip file for better visualization*

### 3.0 Code Structure Overview

The architecture of the Memory Card Game project is organized into 3 main packages or program logic, namely “model”, “util” and “view” classes which satisfies the common software design pattern (Model-View-Controller, MVC) in developing user interfaces. Each package is entitled to its respective responsibility, and it consists of various classes and objects to manage the game’s functionality and flow. Attached shows a clear overview of the functionalities of each class and the main role they play in contributing the overall structure of the code:

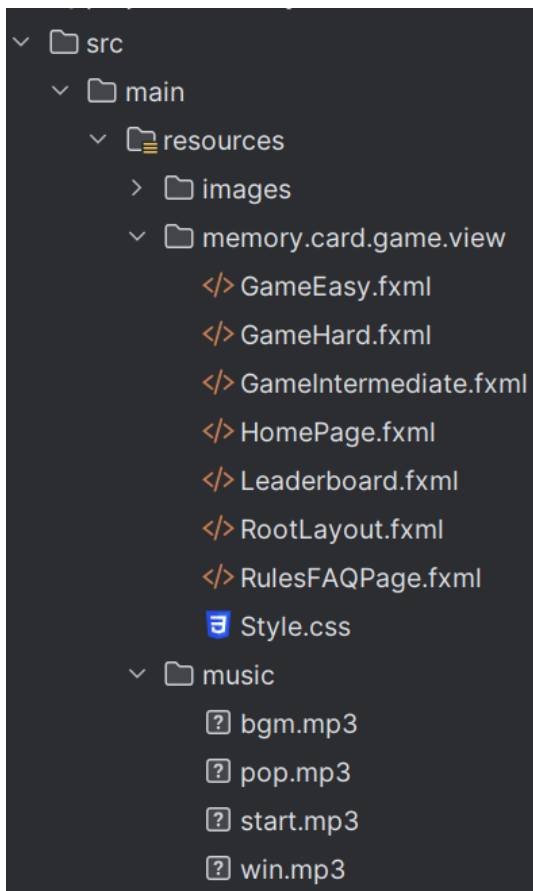


Figure 3.1 File Structure of the Project

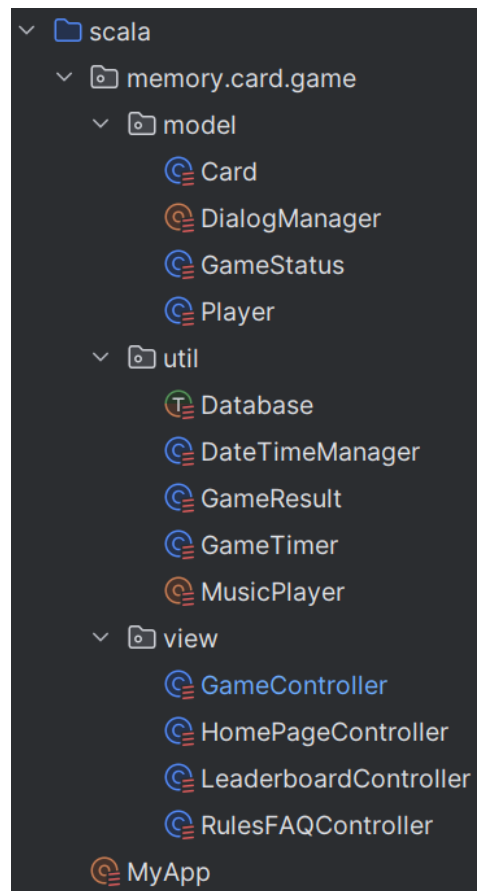


Figure 3.2 File Structure of the Project

Package	Classes & Objects	Responsibility / Functionality
Model	Card.scala	Represents a card in the game and responsible for handling card-related operations, such as revealing cards, hiding cards, checking matched pairs etc.



	DialogManager.scala	Manages the display of different dialogs such as game completion, pause and exit confirmation dialog box
	GameStatus.scala	Handling the current status of the game, such as the number of successful match cards, number of moves taken etc.
	Player.scala	Represents the player with attributes such as username and its corresponding game results
Util	Database.scala	Manage database-related operations, such as setting up database, saving game results, committing game records, retrieve game results etc.
	DateTimeManager.scala	Provides utility functions related to displaying current date and time operations throughout the game
	GameResult.scala	Represents the result of a game, such as player name, difficulty level, time and moves taken, date played etc.
	GameTimer.scala	Manages and tracks, and captures the timer during the game to get the duration of gameplay
	MusicPlayer.scala	Handles the playback of background music and sound effects during the game
View	GameController.scala	Manages the logic for the game pages including game progress and button operations
	HomeController.scala	Handles the logic for home page such as navigation to other screens and exception handling before starting a new game
	LeaderboardController.scala	Manages the logic for displaying top game results and filtering game results based on difficulty levels
	RuesFAQController.scala	Handles logic for rules and FAQ page such as button navigation

View	GameEasy.fxml	Defines the layout of game screen for Easy difficulty level
	GameIntermediate.fxml	Defines the layout of game screen for Intermediate difficulty level
	GameHard.fxml	Defines the layout of game screen for Hard difficulty level
	HomePage.fxml	Defines the layout of home page that serves as the main entry of the game
	Leaderboard.fxml	Defines the layout of the leaderboard page that displays players' result
	RootLayout.fxml	Defines the root layout structure used across the game
	RulesFAQPage.fxml	Defines the layout for the rules and Frequently Asked Questions (FAQs) page
	Style.css	Includes the Cascading Style Sheet (CSS) styles used across the game
MyApp.scala		Serves as the main entry point of the entire Memory Card Game in setting up and launching the application by initializing the primary stage and its corresponding controller class

Table 3.1 Visualization of Code Structure

# 4.0 System Functionalities

In this section, various screenshots and together with the functionalities of the Memory Card Game application are attached and explained in detail to highlight all the key features and user interface elements. Each subsection focuses on a specific aspect of the application.

## 4.1 Home Page

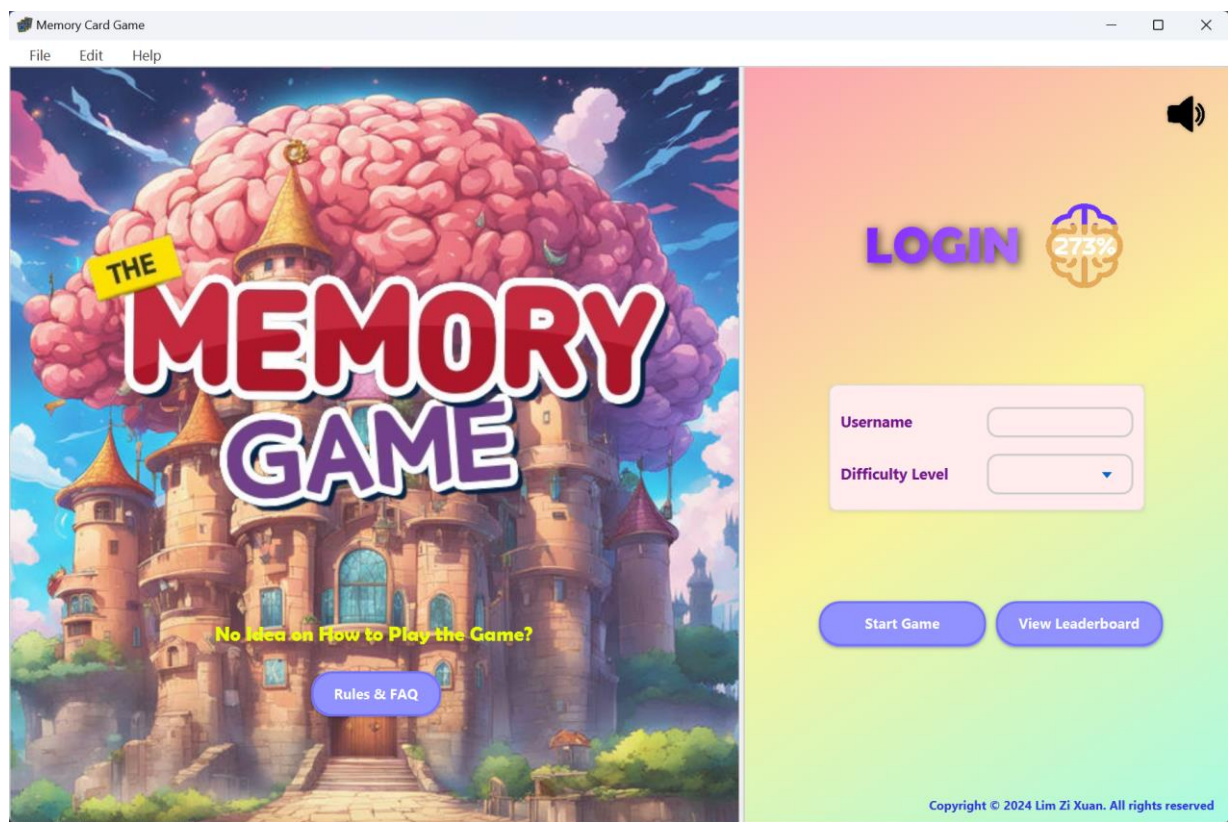


Figure 4.1.1 Overview of Home Page

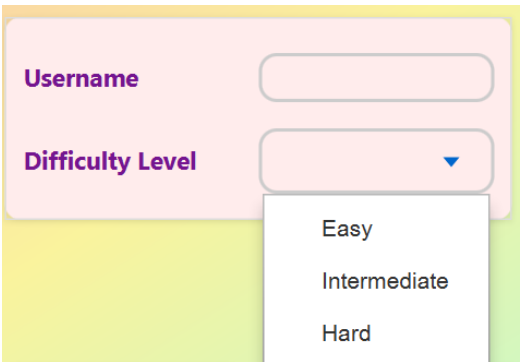


Figure 4.1.2 Difficulty Levels Selection

The Home Page serves as the main entry point to the Memory Card Game. Users can log in with their username, choose the desired difficulty levels which consist of Easy, Intermediate and

Hard levels to start the game. The page also provides easy and clear access to the Leaderboard Page as well as Rules and Frequently Asked Questions (FAQs) sections if the users require guidelines in playing the game. Key elements include:

#### 4.1.1 Login Section

Users are required to enter their username and the difficulty levels they wish to play in order to track their respective game result, otherwise there would be an alert box popping out to remind users to enter all required information before starting the game. The game will only be started if users enter all the required information. Attached shows the implementation of exception handling for required information before starting the game:

- i) Entering username without choosing difficulty levels

<div><div>Username</div><div>Lim</div><div>Difficulty Level</div><div></div></div>	<div><div>Input Error</div><div>Missing Difficulty Level</div><div>Please select a difficulty level to start the game.</div><div>OK</div></div>
--	---

Table 4.1.1 Handling Missing Difficulty Level

- ii) Choosing difficulty levels without entering usernames

<div><div>Username</div><div></div><div>Difficulty Level</div><div>Intermediate</div></div>	<div><div>Input Error</div><div>Missing Username</div><div>Please enter your username to start the game.</div><div>OK</div></div>
---	---

Table 4.1.2 Handling Missing Username

#### 4.1.2 Difficulty Levels Selection

The difficulty levels of the game leverage from Easy, Intermediate and Hard Levels to fulfill user satisfaction. Attached shows the detailed visualization of the Grid Pane ranging from different levels:

Difficulty Levels	Number of Rows	Number of Columns
Easy	4	4
Intermediate	4	6

Hard	6	6
------	---	---

Table 4.1.3 Visualization of Different Difficulty Levels

### 4.1.3 Navigation Buttons

To ensure the user-friendliness of the application, all the navigation buttons are functional in a way that a new page will be instantaneously loaded after clicking the button such as “Start Game”, “View Leaderboard” and “Rules & FAQ” buttons, and this feature creates seamless transitions for user to view comprehensive information that they might look for. By this way, these navigation buttons make it easy for users to find important information and track their performance and this contributes to a more engaging and user-friendly experience, since the users can quickly access key sections of the game without navigating through multiple menus.


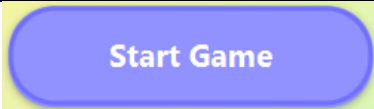
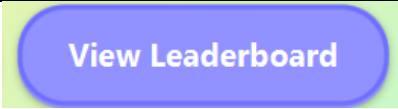
 <p>Direct to Rules &amp; Frequently Asked Questions (FAQ) Page</p>	 <p>Direct to Game Pages based on Difficulty Levels Chosen</p>	 <p>Direct to Leaderboard Page to Track their Score</p>
--	---	--

Table 4.1.4 Various Navigation Buttons

### 4.2 Leaderboard Page

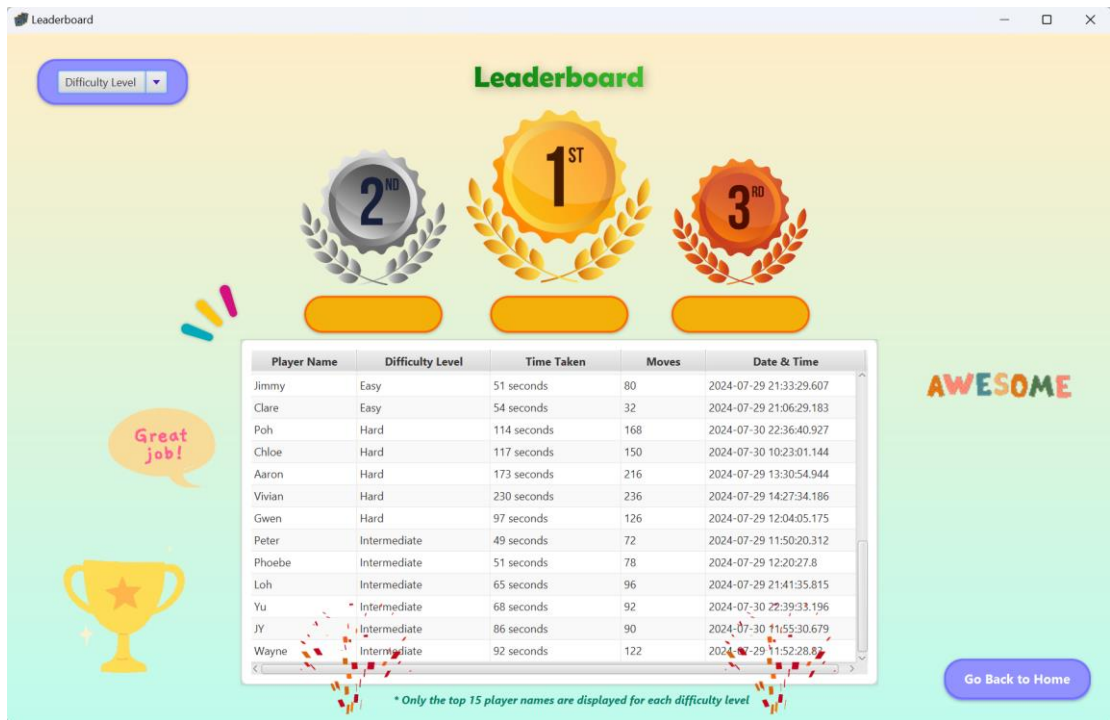


Figure 4.2.1 Overview of Leaderboard Page

The Leaderboard page displays the top player results based on the least number of moves taken and time taken for each difficulty level. This page also allows users to filter results by difficulty. Key features include:

4.2.1 Filter Options

Users are allowed to view the game results from 4 aspects, where they can view it for Easy, Intermediate, Hard or based on both difficulty levels. This allows the user to have a clear and transparent overview of the results from past players.

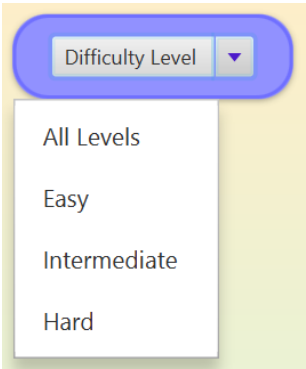




Figure 4.2.2 Visualization of Difficulty Level Filtering

4.2.2 Top Player Labels

The top player labels enable the system to display the top 3 players for each difficulty level. The labels’ names will keep updating from time to time to ensure the accuracy of the ranking.

Difficulties	Evidence (Top Player Names)		
Easy	 Lim	 Ken	 Adrian
Intermediate	 Phoebe	 Peter	 Loh

Hard	
------	--

Table 4.2.1 Visualization of Top Player Results (As of 31<sup>st</sup> July 2024)

### 4.2.3 Detailed Results Table

The game results acquired by each player will also be displayed in the form of a table which stores the values such as player name, difficulty level, time taken in seconds, number of moves taken as well as the date and time they played the game. This is to give an overview of other players and to ensure the transparency of the game. Only the top 15 players' results will be displayed for each difficulty level ("Easy", "Intermediate" and "Hard" choice) but the user can view every single game result under "All Levels" choice.

Difficulties	Evidence (Detailed Result Table)																																																																										
All Levels	<table><tr><th>Player Name</th><th>Difficulty Level</th><th>Time Taken</th><th>Moves</th><th>Date &amp; Time</th></tr><tr><td>Benjamin</td><td>Easy</td><td>38 seconds</td><td>46</td><td>2024-07-29 22:15:27.581</td></tr><tr><td>Dickson</td><td>Easy</td><td>38 seconds</td><td>60</td><td>2024-07-29 21:28:28.375</td></tr><tr><td>Jimmy</td><td>Easy</td><td>51 seconds</td><td>80</td><td>2024-07-29 21:33:29.607</td></tr><tr><td>Clare</td><td>Easy</td><td>54 seconds</td><td>32</td><td>2024-07-29 21:06:29.183</td></tr><tr><td>Poh</td><td>Hard</td><td>114 seconds</td><td>168</td><td>2024-07-30 22:36:40.927</td></tr><tr><td>Chloe</td><td>Hard</td><td>117 seconds</td><td>150</td><td>2024-07-30 10:23:01.144</td></tr><tr><td>Aaron</td><td>Hard</td><td>173 seconds</td><td>216</td><td>2024-07-29 13:30:54.944</td></tr><tr><td>Vivian</td><td>Hard</td><td>230 seconds</td><td>236</td><td>2024-07-29 14:27:34.186</td></tr><tr><td>Gwen</td><td>Hard</td><td>97 seconds</td><td>126</td><td>2024-07-29 12:04:05.175</td></tr><tr><td>Peter</td><td>Intermediate</td><td>49 seconds</td><td>72</td><td>2024-07-29 11:50:20.312</td></tr><tr><td>Phoebe</td><td>Intermediate</td><td>51 seconds</td><td>78</td><td>2024-07-29 12:20:27.8</td></tr><tr><td>Loh</td><td>Intermediate</td><td>65 seconds</td><td>96</td><td>2024-07-29 21:41:35.815</td></tr><tr><td>Yu</td><td>Intermediate</td><td>68 seconds</td><td>92</td><td>2024-07-30 22:39:33.196</td></tr></table>					Player Name	Difficulty Level	Time Taken	Moves	Date & Time	Benjamin	Easy	38 seconds	46	2024-07-29 22:15:27.581	Dickson	Easy	38 seconds	60	2024-07-29 21:28:28.375	Jimmy	Easy	51 seconds	80	2024-07-29 21:33:29.607	Clare	Easy	54 seconds	32	2024-07-29 21:06:29.183	Poh	Hard	114 seconds	168	2024-07-30 22:36:40.927	Chloe	Hard	117 seconds	150	2024-07-30 10:23:01.144	Aaron	Hard	173 seconds	216	2024-07-29 13:30:54.944	Vivian	Hard	230 seconds	236	2024-07-29 14:27:34.186	Gwen	Hard	97 seconds	126	2024-07-29 12:04:05.175	Peter	Intermediate	49 seconds	72	2024-07-29 11:50:20.312	Phoebe	Intermediate	51 seconds	78	2024-07-29 12:20:27.8	Loh	Intermediate	65 seconds	96	2024-07-29 21:41:35.815	Yu	Intermediate	68 seconds	92	2024-07-30 22:39:33.196
	Player Name	Difficulty Level	Time Taken	Moves	Date & Time																																																																						
	Benjamin	Easy	38 seconds	46	2024-07-29 22:15:27.581																																																																						
	Dickson	Easy	38 seconds	60	2024-07-29 21:28:28.375																																																																						
	Jimmy	Easy	51 seconds	80	2024-07-29 21:33:29.607																																																																						
	Clare	Easy	54 seconds	32	2024-07-29 21:06:29.183																																																																						
	Poh	Hard	114 seconds	168	2024-07-30 22:36:40.927																																																																						
	Chloe	Hard	117 seconds	150	2024-07-30 10:23:01.144																																																																						
	Aaron	Hard	173 seconds	216	2024-07-29 13:30:54.944																																																																						
	Vivian	Hard	230 seconds	236	2024-07-29 14:27:34.186																																																																						
	Gwen	Hard	97 seconds	126	2024-07-29 12:04:05.175																																																																						
	Peter	Intermediate	49 seconds	72	2024-07-29 11:50:20.312																																																																						
	Phoebe	Intermediate	51 seconds	78	2024-07-29 12:20:27.8																																																																						
	Loh	Intermediate	65 seconds	96	2024-07-29 21:41:35.815																																																																						
	Yu	Intermediate	68 seconds	92	2024-07-30 22:39:33.196																																																																						
(Only parts of the player results are captured in this picture. For detailed visualization, please refer to the actual game interface.)																																																																											

Easy	<table><tr><th>Player Name</th><th>Difficulty Level</th><th>Time Taken</th><th>Moves</th><th>Date &amp; Time</th></tr><tr><td>Ken</td><td>Easy</td><td>14 seconds</td><td>30</td><td>2024-07-29 12:15:47.866</td></tr><tr><td>Lim</td><td>Easy</td><td>18 seconds</td><td>32</td><td>2024-07-29 11:29:17.397</td></tr><tr><td>Adrian</td><td>Easy</td><td>20 seconds</td><td>34</td><td>2024-07-29 21:24:15.846</td></tr><tr><td>Jane</td><td>Easy</td><td>23 seconds</td><td>40</td><td>2024-07-29 11:39:54.606</td></tr><tr><td>Jing</td><td>Easy</td><td>25 seconds</td><td>46</td><td>2024-07-30 17:53:01.435</td></tr><tr><td>Jia</td><td>Easy</td><td>26 seconds</td><td>32</td><td>2024-07-29 14:11:25.126</td></tr><tr><td>Tommy</td><td>Easy</td><td>27 seconds</td><td>36</td><td>2024-07-29 21:09:00.82</td></tr><tr><td>Pooh</td><td>Easy</td><td>27 seconds</td><td>46</td><td>2024-07-29 11:35:09.053</td></tr><tr><td>Ying</td><td>Easy</td><td>29 seconds</td><td>50</td><td>2024-07-29 14:35:13.315</td></tr><tr><td>Danny</td><td>Easy</td><td>29 seconds</td><td>50</td><td>2024-07-29 12:05:13.685</td></tr><tr><td>Hoo</td><td>Easy</td><td>31 seconds</td><td>42</td><td>2024-07-31 16:01:33.205</td></tr><tr><td>Ivy</td><td>Easy</td><td>32 seconds</td><td>46</td><td>2024-07-29 11:32:59.478</td></tr><tr><td>Sebastian</td><td>Easy</td><td>32 seconds</td><td>50</td><td>2024-07-30 22:26:24.546</td></tr><tr><td>Tim</td><td>Easy</td><td>34 seconds</td><td>56</td><td>2024-07-29 14:13:00.675</td></tr><tr><td>Owen</td><td>Easy</td><td>35 seconds</td><td>58</td><td>2024-07-29 11:33:55.756</td></tr></table>	Player Name	Difficulty Level	Time Taken	Moves	Date & Time	Ken	Easy	14 seconds	30	2024-07-29 12:15:47.866	Lim	Easy	18 seconds	32	2024-07-29 11:29:17.397	Adrian	Easy	20 seconds	34	2024-07-29 21:24:15.846	Jane	Easy	23 seconds	40	2024-07-29 11:39:54.606	Jing	Easy	25 seconds	46	2024-07-30 17:53:01.435	Jia	Easy	26 seconds	32	2024-07-29 14:11:25.126	Tommy	Easy	27 seconds	36	2024-07-29 21:09:00.82	Pooh	Easy	27 seconds	46	2024-07-29 11:35:09.053	Ying	Easy	29 seconds	50	2024-07-29 14:35:13.315	Danny	Easy	29 seconds	50	2024-07-29 12:05:13.685	Hoo	Easy	31 seconds	42	2024-07-31 16:01:33.205	Ivy	Easy	32 seconds	46	2024-07-29 11:32:59.478	Sebastian	Easy	32 seconds	50	2024-07-30 22:26:24.546	Tim	Easy	34 seconds	56	2024-07-29 14:13:00.675	Owen	Easy	35 seconds	58	2024-07-29 11:33:55.756
	Player Name	Difficulty Level	Time Taken	Moves	Date & Time																																																																												
	Ken	Easy	14 seconds	30	2024-07-29 12:15:47.866																																																																												
	Lim	Easy	18 seconds	32	2024-07-29 11:29:17.397																																																																												
	Adrian	Easy	20 seconds	34	2024-07-29 21:24:15.846																																																																												
	Jane	Easy	23 seconds	40	2024-07-29 11:39:54.606																																																																												
	Jing	Easy	25 seconds	46	2024-07-30 17:53:01.435																																																																												
	Jia	Easy	26 seconds	32	2024-07-29 14:11:25.126																																																																												
	Tommy	Easy	27 seconds	36	2024-07-29 21:09:00.82																																																																												
	Pooh	Easy	27 seconds	46	2024-07-29 11:35:09.053																																																																												
	Ying	Easy	29 seconds	50	2024-07-29 14:35:13.315																																																																												
	Danny	Easy	29 seconds	50	2024-07-29 12:05:13.685																																																																												
	Hoo	Easy	31 seconds	42	2024-07-31 16:01:33.205																																																																												
	Ivy	Easy	32 seconds	46	2024-07-29 11:32:59.478																																																																												
	Sebastian	Easy	32 seconds	50	2024-07-30 22:26:24.546																																																																												
	Tim	Easy	34 seconds	56	2024-07-29 14:13:00.675																																																																												
Owen	Easy	35 seconds	58	2024-07-29 11:33:55.756																																																																													
(Only parts of the player results are captured in this picture. For detailed visualization, please refer to the actual game interface.)																																																																																	
Intermediate	<table><tr><th>Player Name</th><th>Difficulty Level</th><th>Time Taken</th><th>Moves</th><th>Date &amp; Time</th></tr><tr><td>Peter</td><td>Intermediate</td><td>49 seconds</td><td>72</td><td>2024-07-29 11:50:20.312</td></tr><tr><td>Phoebe</td><td>Intermediate</td><td>51 seconds</td><td>78</td><td>2024-07-29 12:20:27.8</td></tr><tr><td>Loh</td><td>Intermediate</td><td>65 seconds</td><td>96</td><td>2024-07-29 21:41:35.815</td></tr><tr><td>Yu</td><td>Intermediate</td><td>68 seconds</td><td>92</td><td>2024-07-30 22:39:33.196</td></tr><tr><td>JY</td><td>Intermediate</td><td>86 seconds</td><td>90</td><td>2024-07-30 11:55:30.679</td></tr><tr><td>Wayne</td><td>Intermediate</td><td>92 seconds</td><td>122</td><td>2024-07-29 11:52:28.83</td></tr></table>	Player Name	Difficulty Level	Time Taken	Moves	Date & Time	Peter	Intermediate	49 seconds	72	2024-07-29 11:50:20.312	Phoebe	Intermediate	51 seconds	78	2024-07-29 12:20:27.8	Loh	Intermediate	65 seconds	96	2024-07-29 21:41:35.815	Yu	Intermediate	68 seconds	92	2024-07-30 22:39:33.196	JY	Intermediate	86 seconds	90	2024-07-30 11:55:30.679	Wayne	Intermediate	92 seconds	122	2024-07-29 11:52:28.83																																													
	Player Name	Difficulty Level	Time Taken	Moves	Date & Time																																																																												
	Peter	Intermediate	49 seconds	72	2024-07-29 11:50:20.312																																																																												
	Phoebe	Intermediate	51 seconds	78	2024-07-29 12:20:27.8																																																																												
	Loh	Intermediate	65 seconds	96	2024-07-29 21:41:35.815																																																																												
	Yu	Intermediate	68 seconds	92	2024-07-30 22:39:33.196																																																																												
	JY	Intermediate	86 seconds	90	2024-07-30 11:55:30.679																																																																												
Wayne	Intermediate	92 seconds	122	2024-07-29 11:52:28.83																																																																													
Hard	<table><tr><th>Player Name</th><th>Difficulty Level</th><th>Time Taken</th><th>Moves</th><th>Date &amp; Time</th></tr><tr><td>Poh</td><td>Hard</td><td>114 seconds</td><td>168</td><td>2024-07-30 22:36:40.927</td></tr><tr><td>Chloe</td><td>Hard</td><td>117 seconds</td><td>150</td><td>2024-07-30 10:23:01.144</td></tr><tr><td>Aaron</td><td>Hard</td><td>173 seconds</td><td>216</td><td>2024-07-29 13:30:54.944</td></tr><tr><td>Vivian</td><td>Hard</td><td>230 seconds</td><td>236</td><td>2024-07-29 14:27:34.186</td></tr><tr><td>Gwen</td><td>Hard</td><td>97 seconds</td><td>126</td><td>2024-07-29 12:04:05.175</td></tr></table>	Player Name	Difficulty Level	Time Taken	Moves	Date & Time	Poh	Hard	114 seconds	168	2024-07-30 22:36:40.927	Chloe	Hard	117 seconds	150	2024-07-30 10:23:01.144	Aaron	Hard	173 seconds	216	2024-07-29 13:30:54.944	Vivian	Hard	230 seconds	236	2024-07-29 14:27:34.186	Gwen	Hard	97 seconds	126	2024-07-29 12:04:05.175																																																		
	Player Name	Difficulty Level	Time Taken	Moves	Date & Time																																																																												
	Poh	Hard	114 seconds	168	2024-07-30 22:36:40.927																																																																												
	Chloe	Hard	117 seconds	150	2024-07-30 10:23:01.144																																																																												
	Aaron	Hard	173 seconds	216	2024-07-29 13:30:54.944																																																																												
	Vivian	Hard	230 seconds	236	2024-07-29 14:27:34.186																																																																												
Gwen	Hard	97 seconds	126	2024-07-29 12:04:05.175																																																																													

Table 4.2.2 Detailed Game Result Table (As of 31<sup>st</sup> July 2024)



## 4.3 Rules and Frequently Asked Questions (FAQs) Page

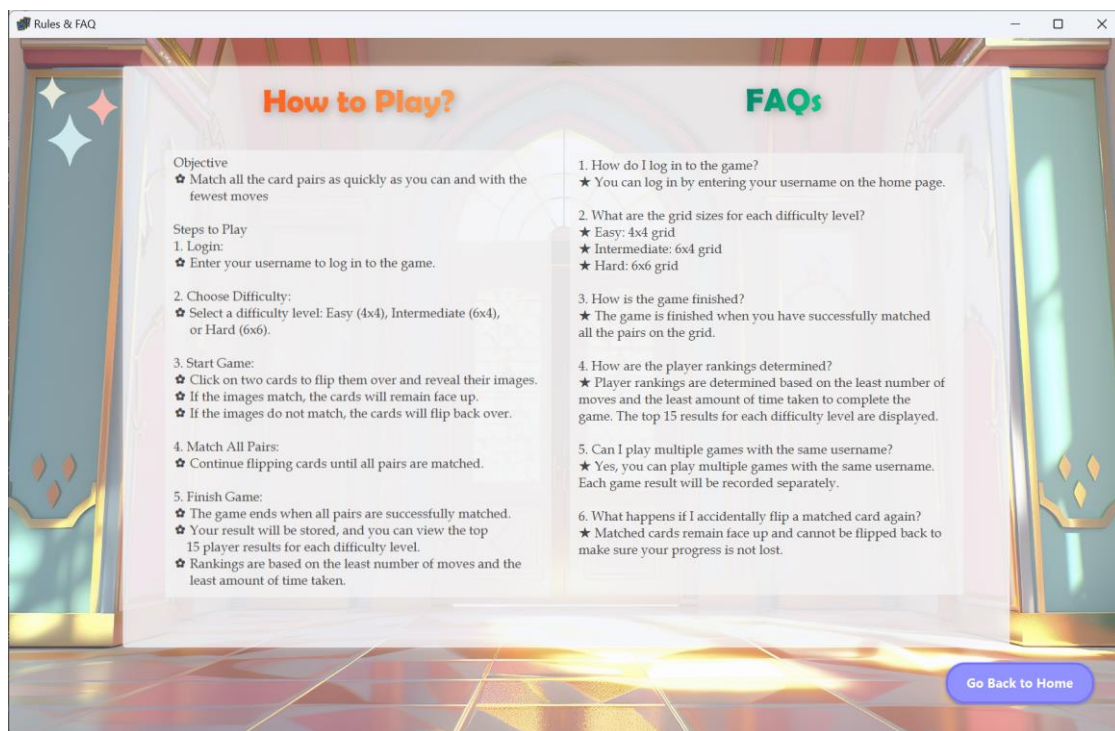


Figure 4.3.1 Overview of Rules & FAQ Page

The Rules and FAQ page provides a comprehensive overview of the game's rules and frequently asked questions. By having this page, this helps new players to understand better how to play and clarifies common inquiries. Key features include:

### 4.3.1 Game Rules

The game rules section allows users to gain detailed instructions and insight on how to play the game.

### 4.3.2 Frequently Asked Questions (FAQs)

The frequently asked questions section provides answers to common inquiries about gameplay, scoring system and other important features.

### 4.3.3 Return Home Button

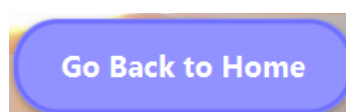


Figure 4.3.2 Return Home Button

For ease of use in terms of user experience, a “Go Back to Home” button is implemented to allow users to return to Home Page after finish reading the game rules to allow seamless transition.

#### 4.4 Game Pages (Easy, Intermediate, Hard Levels)



Figure 4.4.1 Overview of Game Page (Easy Mode)



Figure 4.4.2 Overview of Game Page (Intermediate Mode)



Figure 4.4.3 Overview of Game Page (Hard Mode)

The game pages are presented according to different grid sizes of cards. Players are required to flip the cards to find their corresponding matching pairs. The game will only stop and finish after players have matched all pairs of cards present in the grid. Key features include:

#### 4.4.1 Game Grid

The game grid is represented in different sizes according to the difficulty levels users have chosen. Attached is the detailed visualization of game grid in different levels:




		
Easy (4 rows x 4 columns)	Intermediate (4 rows x 6 columns)	Hard (6 rows x 6 columns)

Table 4.4.1 Visualization of Game Grid



4.4.2 Flip Card

Users are required to flip cards by clicking on the back view of cards (represented in multiple, purple-colored question mark pictures) to match the corresponding pairs. Appropriate logic is also implemented to prevent the matched pair of cards from flipping back facing down so as to avoid any accidental taps when the users are playing the game to ensure users’ satisfaction.

The important aspect to be taken into account is that the randomized distribution of cards for every single game round, which means the distribution of cards are to be entirely different in every new game round, no matter it is in Easy, Intermediate or Hard level, so as to ensure the fairness and the difficulty throughout the game.

4.4.3 Visualization of Game Status

The game pages provide clear status of the progress in terms of 4 critical aspects, which are the number of success matched pairs, game timer, current date and time as well as the number of moves.


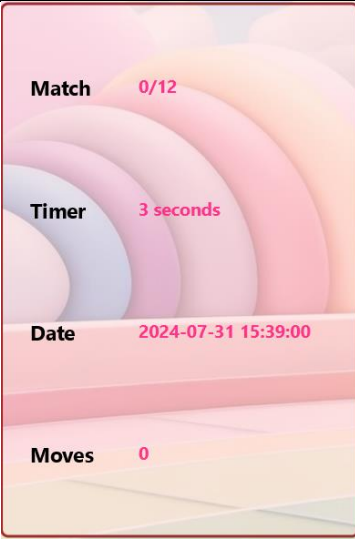

 The visualization for the Easy level shows a blue background with green leaves on the left. It displays 'Match' as 0/8, 'Timer' as 207 seconds, 'Date' as 2024-07-31 15:37:57, and 'Moves' as 7.	 The visualization for the Intermediate level shows a background of pink and orange curved shapes. It displays 'Match' as 0/12, 'Timer' as 3 seconds, 'Date' as 2024-07-31 15:39:00, and 'Moves' as 0.	 The visualization for the Hard level shows a pink desk with a lamp and a plant. It displays 'Match' as 0/18, 'Timer' as 2 seconds, 'Date' as 2024-07-31 15:39:22, and 'Moves' as 0.
Easy	Intermediate	Hard

Table 4.4.2 Game Status for Each Level

4.4.3.1 Success Matched Pairs

The successful matched pairs are represented according to the number of cards sets loaded in each difficulty level. This is to give a clear progress on the number of pairs they have matched successfully and this ensures the user-friendliness of the interface. This information will then be captured to be stored in the system database.

#### 4.4.3.2 Game Timer

A timer is initialized on the game page in order to track the players' performance in terms of the speed of finishing up the whole game, and it is captured in second's format. The result with the least time taken will be ranked at the top. This information will then be captured to be stored in the system database.

#### 4.4.3.3 Current Date and Time

Current date and time will be displayed when users are playing the game to provide up to date information. This information will be captured to be stored in the system database.

#### 4.4.3.4 Moves Taken

The number of moves taken is essential in determining the rank of the player. The player result with the least moves taken will be ranked at the top. This information will be captured to be stored in the system database.

### 4.5 Pause Game Function

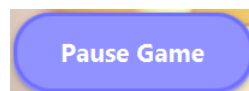


Figure 4.5.1 Pause Game Button Displayed in Each Game Page

Players can feel free to pause the game at any time if they wish to. The pause dialog box allows the user to resume the game when they are ready again to ensure the players can take a break in between without losing their progress. When the "Pause Game" button is pressed, an alert dialog will be displayed to show the current status of the game, and the timer will stop counting until the player clicks the "Resume" button in the alert dialog box.

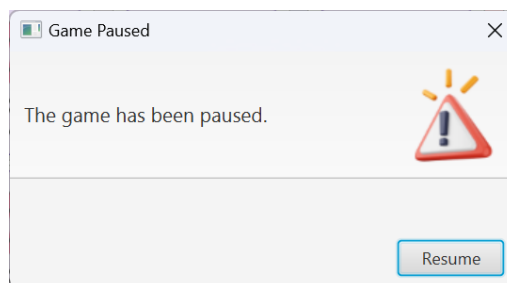


Figure 4.5.2 Alert Dialog when Pausing Game

## 4.6 Return Home Function

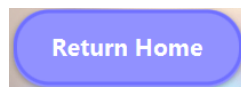


Figure 4.6.1 Return Home Button Displayed in Each Level

The “Return Home” function allows player to exit the current game and go back to the home page. A confirmation dialog will be displayed to ensure that the players are aware they will lose their progress if they exit the game by clicking “Exit Anyway” button in the dialog box.

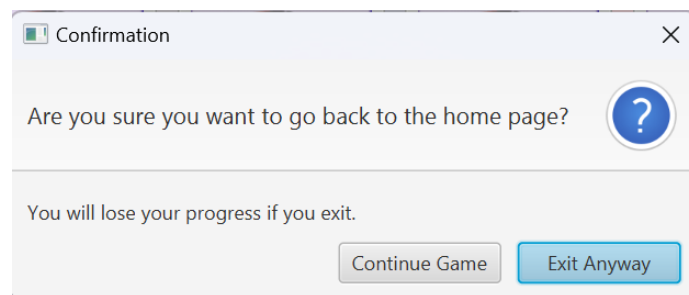


Figure 4.6.2 Confirmation Dialog After Clicking Return Home Button

## 4.7 Game Over Dialog Box

After the players have finished matching all the pairs of cards, the game over dialog will be displayed to congratulate the players and display the time taken and moves taken to provide a clear overview of their performance. The player’s results are then saved into the system database and displayed on the Leaderboard page. After that, the game will direct the player to the Home Page to start a new game if they wish to.

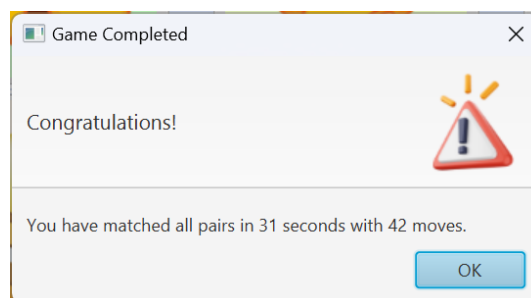


Figure 4.7.1 Dialog Box Upon Finishing the Game

## 4.8 Volume Button Control

To ensure the users’ satisfaction and create more engaging moment throughout the game, the background music is implemented, and users are allowed to control the background music by

pressing the On and Off icon displayed at the top left or top right corner of the screens. The icon is implemented in Home Page and respective Game Pages to enhance the user experience when the game is ongoing. Also, the icon perfectly reflects the status of the background music to make sure of its accuracy.


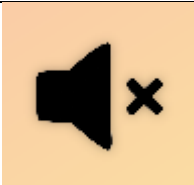
	
Background Music On (by default)	Background Music Off

Table 4.8.1 Visualization of Volume Button

### 4.9 Database Visualization with DataGrip

For better visualization for players’ game results, the Apache Derby database used in this Memory Card Game project has been connected to the DataGrip software which serves as a powerful Integrated Development Environment (IDE) for cross-platform database visualization. Reason being that is due to the functionality of displaying only the top 15 players’ game results based on each difficulty level, thus this approach efficiently improves the ease of obtaining necessary information. Below shows the detailed view of game result by connecting myDB Apache Derby database into DataGrip:

	
---	--

Table 4.9.1 Documents Created for Cross-platform Database Connection

Besides, attached shows the necessary information such as username and password for Apache Derby database creation and DataGrip connection:

```

trait Database {
    val derbyDriverClassname = "org.apache.derby.jdbc.EmbeddedDriver"
    val dbURL = "jdbc:derby:myDB;create=true;"
    Class.forName(derbyDriverClassname)

    ConnectionPool.singleton(dbURL, "user", "pass", ConnectionPoolSettings(
        initialSize = 5,
        maxSize = 20,
        connectionTimeoutMillis = 3000L,
        validationQuery = "SELECT 1 FROM SYS.SYSSCHEMAS"
    ))

    implicit val session: AutoSession.type = AutoSession
}

```

Figure 4.9.1 Codes Representing the Creation of Database

For illustration purpose, the attached shows the players' game record based on difficulty levels of Easy. Only the top 15 players are to be shown in the Leaderboard page based on the time and moves taken. The remaining records from 16<sup>th</sup> place onwards will be kept in the DataGrip Database.

Player Name	Difficulty Level	Time Taken	Moves	Date & Time
Ken	Easy	14 seconds	30	2024-07-29 12:15:47.866
Lim	Easy	18 seconds	32	2024-07-29 11:29:17.397
Adrian	Easy	20 seconds	34	2024-07-29 21:24:15.846
Jane	Easy	23 seconds	40	2024-07-29 11:39:54.606
Jing	Easy	25 seconds	46	2024-07-30 17:53:01.435
Jia	Easy	26 seconds	32	2024-07-29 14:11:25.126
Tommy	Easy	27 seconds	36	2024-07-29 21:09:00.82
Pooh	Easy	27 seconds	46	2024-07-29 11:35:09.053
Ying	Easy	29 seconds	50	2024-07-29 14:35:13.315
Danny	Easy	29 seconds	50	2024-07-29 12:05:13.685
Hoo	Easy	31 seconds	42	2024-07-31 16:01:33.205
Ivy	Easy	32 seconds	46	2024-07-29 11:32:59.478
Sebastian	Easy	32 seconds	50	2024-07-30 22:26:24.546
Tim	Easy	34 seconds	56	2024-07-29 14:13:00.675
Owen	Easy	35 seconds	58	2024-07-29 11:33:55.756

Figure 4.9.2 Top 15 Players Shown in Leaderboard Page (As of 31<sup>st</sup> July 2024)

Below shows the actual ranking ranging from the 1<sup>st</sup> place to the 21<sup>st</sup> place (as the total number of players playing the Easy Level are 21 as of 31<sup>st</sup> July 2024). There are extra 6 game records which are not displayed in the Leaderboard Page, given that there will only be top 15 players will be displayed.



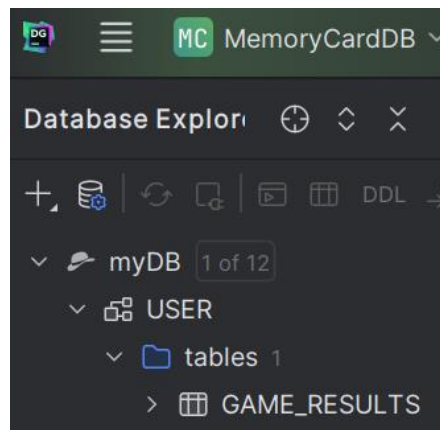


Figure 4.9.3 Database Structure Shown in the DataGrip with GAME\_RESULTS TABLE

WHERE		ORDER BY TIME_TAKEN, MOVES_TAKEN					
ID	USERNAME	DIFFICULTY	DATE_PLAYED	TIME_TAKEN	MOVES_TAKEN		
1	901 Ken	Easy	2024-07-29 12:15:47.866000000	14 seconds			30
2	1 Lim	Easy	2024-07-29 11:29:17.397000000	18 seconds			32
3	2001 Adrian	Easy	2024-07-29 21:24:15.846000000	20 seconds			34
4	401 Jane	Easy	2024-07-29 11:39:54.606000000	23 seconds			40
5	2701 Jing	Easy	2024-07-30 17:53:01.435000000	25 seconds			46
6	1301 Jia	Easy	2024-07-29 14:11:25.126000000	26 seconds			32
7	1801 Tommy	Easy	2024-07-29 21:09:00.820000000	27 seconds			36
8	301 Pooh	Easy	2024-07-29 11:35:09.053000000	27 seconds			46
9	1601 Ying	Easy	2024-07-29 14:35:13.315000000	29 seconds			50
10	801 Danny	Easy	2024-07-29 12:05:13.685000000	29 seconds			50
11	3101 Hoo	Easy	2024-07-31 16:01:33.205000000	31 seconds			42
12	101 Ivy	Easy	2024-07-29 11:32:59.478000000	32 seconds			46
13	2801 Sebastian	Easy	2024-07-30 22:26:24.546000000	32 seconds			50
14	1401 Tim	Easy	2024-07-29 14:13:00.675000000	34 seconds			56
15	201 Owen	Easy	2024-07-29 11:33:55.756000000	35 seconds			58
16	1901 Roy	Easy	2024-07-29 21:22:05.180000000	37 seconds			58
17	1101 Hannah	Easy	2024-07-29 12:49:13.042000000	37 seconds			60
18	2401 Benjamin	Easy	2024-07-29 22:15:27.581000000	38 seconds			46
19	2101 Dickson	Easy	2024-07-29 21:28:28.375000000	38 seconds			60
20	2201 Jimmy	Easy	2024-07-29 21:33:29.607000000	51 seconds			80
21	1701 Clare	Easy	2024-07-29 21:06:29.183000000	54 seconds			32

Figure 4.9.4 Complete Game Records for Easy Level Shown in DataGrip (GAME\_RESULTS)

Based on the evidence shown in Figure 4.9.3, the records are sorted based on the SQL command “ORDER BY TIME\_TAKEN, MOVES\_TAKEN” so as to rank the top 15 players. The 16<sup>th</sup> place onwards which holds the IDs of 1901 (Roy), 1101 (Hannah), 2401 (Benjamin), 2101 (Dickson), 2201 (Jimmy), 1701 (Clare) are not displayed on the Leaderboard Page.

## 5.0 Personal Reflection

### 5.1 Applications of OOP Concepts

Throughout the development of Memory Card Game project, there are several Object-oriented Programming (OOP) concepts applied to reduce the redundancy of the codes, including inheritance, parametric polymorphism, generic programming as well as collections.

#### 5.1.1 Inheritance

```
trait Database {  
    val derbyDriverClassname = "org.apache.derby.jdbc.EmbeddedDriver"  
    val dbURL = "jdbc:derby:myDB;create=true;"  
    Class.forName(derbyDriverClassname)  
  
    ConnectionPool.singleton(dbURL, "user", "pass", ConnectionPoolSettings(  
        initialSize = 5,  
        maxSize = 20,  
        connectionTimeoutMillis = 3000L,  
        validationQuery = "SELECT 1 FROM SYS.SYSSCHEMAS"  
    ))  
  
    implicit val session: AutoSession.type = AutoSession  
}  
  
object Database extends Database {
```

```
class Player(  
    usernames: String,  
    difficultyLevels: String,  
    playDate: LocalDate,  
    timeTaken: Duration,  
    scores: Int  
    ) extends Database {  
    var name = new StringProperty(this, "name", usernames)  
    var level = new StringProperty(this, "level", difficultyLevels)  
    var date = new ObjectProperty[LocalDate](this, "date", playDate)  
    var time = new ObjectProperty[Duration](this, "time", timeTaken)  
    var move = new IntegerProperty(this, "move", scores)
```

Table 5.1.1.1 Implementation of Inheritance

The object-oriented programming technique of inheritance, especially the use of trait-based inheritance is demonstrated in the code, where a Database trait is defined to encapsulate the common functionality related to database operations. The Database object then extends the Database trait. By extending the Database trait, the Database object inherits all the fields and

methods defined in the trait such as the `derbyDriverClassname`, `dbURL` and connection pool setup. Using this technique allows Database object to use these members directly without having to redefine them. Apart from that, the Player class also demonstrates inheritance by extending the Database object as well to handle game-related operations such as storing retrieving player's game results.

### 5.1.2 Parametric Polymorphism & Generic Programming

This project code also includes the aspect of parametric polymorphism which in turns achieves the generic programming principles through the use of generic types and methods that abstract over different concrete implementations. The Database object uses generic List and Option types in its methods. For example, the `getGameResults`, `getTopResultsByDifficulty` and `getAllResults` methods all return `List[GameResult]` which serve as a generic collection type. Besides, the Option type used in GameController is used to represent the possibility of a value being present or absent in a type-safe manner. For instance, `firstCard` and `secondCard` are of type `Option[Card]` which is a generic type. The `generateCardValues` method uses the Seq type, and it is also a generic type that can be instantiated as different concrete sequence types like List, Vector, or Array. The method `generateCardValues` returns a `Seq[Int]` which means it works with any type that extends Seq.

### 5.1.3 Collections

<pre>private val matchedPairs: mutable.Set[(Int, Int)] = mutable.Set()</pre>
<pre>private def generateCardValues(): Seq[Int] = {   val totalCards = rows * cols   val numPairs = totalCards / 2   (1 to numPairs).flatMap(x =&gt; Seq(x, x)) }</pre>

Table 5.1.3.1 Visualization for Scala Collection

Scala collections have been implemented in this project, where the `mutable.Set[(Int, Int)]` for `matchedPairs` is used in GameController class. This implementation allows the compiler to dynamically load and add the card pairs based on different difficulty levels, which are Easy, Intermediate and Hard. Besides, the `generateCardValues()` method also uses Scala's

collection methods such as flatMap and Seq to create a sequence of card values for loading the card images.

## 5.2 Challenges and Solutions

Throughout the journey of developing this Memory Card Game project, there were several challenges I have encountered and successfully resolved. First and foremost, the integration of external libraries for database access library such as ScalikeJDBC were fairly complex to be implemented because of the compatibility and dependency issues. This issue was resolved by observing various detailed documentation together with constant testing to ensure its functionality. From this case, I learnt a lesson that thorough research and regular testing should always be held whenever possible to ensure every task will be going smoothly without any glitches. Besides, developing the game logic to handle all related operations that might have in the game, such as handling the matching of cards, flipping cards to reveal or hide them and constantly updating the game status were complex. This issue was overcome by dividing down a complex problem into multiple tiny and manageable parts and initiating enough logical thinking to fully master the logic of the program.

## 5.3 Project Strengths

When it comes to the strength of the project, the Memory Card Game provides a fun and effective way to strengthen players' memory skills and cognitive functions. Players can improve their short-term memory and concentration through repeated play. Besides, this project also showcases an insightful technical application through the implementation of Scala, ScalaFX and Java libraries and this provides a deep learning experience when it comes to the demonstration on how to build a fully functional desktop application with attractive Graphical User Interface (GUI). Apart from that, this Memory Card Game also features an engaging and interactive user experience with visually appealing design and smooth animations. The integration of background music and sound effects when dealing with different operations, for example starting a new game, successfully matching a pair of cards and completing a game has made the game more enjoyable.

## 5.4 Weaknesses & Areas of Improvement

While having strengths for this project, this project also contains some limitations and the areas of improvement to give a better experience for users. One of the limitations is that the

current version of Memory Card Game does not allow users to choose their desired theme or to customize their own interface. The issue could be overcome by introducing more theme option in terms of the background interface and card themes to create more interactive user interface and enhance user satisfaction. Besides, the current game primarily relies on the mouse input to handle most of the operations such as pausing game, resuming game, returning to home page, muting the background music etc. by enabling user to use keyboard input for various operation could efficiently improve the accessibility and make the game more user-friendly. For instance, the user-friendliness could be improved if there allows the use of “P” key on keyboard to pause the game, “M” key to mute or unmute the background music and many more.

## 6.0 Conclusion

The Memory Card Game was a highly comprehensive project that aimed to instill educational purpose while at the same time engaging the user to create a practical desktop application that features a rather attractive Graphical User Interface (GUI). The development of this project became not only a good example of usage Scala, ScalaFX and Java knowing and skills, but also a good lesson of practical software designing, user interface developing and database management. In a nutshell, this project has been itself a rather useful learning experience which emphasizes the necessity of planning, continuous learning, and adaptability from the aspect of software development. Indeed, the Memory Card Game project achieved the laid down goals to offer education and entertainment and lays the strong groundwork to offer future advancements and enhancements as well as for other projects.

## 7.0 References

### 7.1 Concept References

[1] “Apache Derby: Datagrip,” DataGrip Help,

<https://www.jetbrains.com/help/datagrip/apache-derby.html#connect-to-derby-database>

[2] “SCALIKEJDBC,” ScalikeJDBC, <https://scalikejdbc.org/documentation/2.x/>

[3] C. Teck Min, “AddressApp Online Recording ,” Microsoft Teams,

[https://imailsunwayedu.sharepoint.com/:v:/r/sites/PRG2104OOPAPR2024/Shared%20Documents/Practical%206/Recordings/OOP%20Practical%20G6-20240717\\_081253-Meeting%20Recording.mp4?csf=1&web=1&e=gdXH6U](https://imailsunwayedu.sharepoint.com/:v:/r/sites/PRG2104OOPAPR2024/Shared%20Documents/Practical%206/Recordings/OOP%20Practical%20G6-20240717_081253-Meeting%20Recording.mp4?csf=1&web=1&e=gdXH6U)

[4] GeeksforGeeks, “Memory game in Java,” GeeksforGeeks,

<https://www.geeksforgeeks.org/memory-game-in-java/>

[5] T. Kai Wei, “AddressApp Online Recording ,” Microsoft Teams,

[https://imailsunwayedu.sharepoint.com/:v:/r/sites/PRG2104OOPAPR2024/Shared%20Documents/Practical%201/Recordings/Meeting%20in%20\\_Practical%201\\_-20240716\\_100843-Meeting%20Recording.mp4?csf=1&web=1&e=d8lmS4](https://imailsunwayedu.sharepoint.com/:v:/r/sites/PRG2104OOPAPR2024/Shared%20Documents/Practical%201/Recordings/Meeting%20in%20_Practical%201_-20240716_100843-Meeting%20Recording.mp4?csf=1&web=1&e=d8lmS4)

### 7.2 Background Music & Sound Effect Resources

[6] “Background Music for gaming videos: Universal production music,” Background Music For Gaming Videos | Universal Production Music,

<https://www.universalproductionmusic.com/en-se/discover/collections/gaming>

[7] “What are you looking for?,” Tunetank, <https://tunetank.com/discover/themes/gaming/>

[8] “Free video games music MP3 download - pixabay,” Pixabay,

<https://pixabay.com/music/search/genre/video%20games/>

### 7.3 Image Resources

[9] <https://www.deviantart.com/dreamup>

[10] <https://www.pexels.com/>

[11] <https://www.canva.com/ai-image-generator/>