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 2023; SEMESTER 3

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..... (Name) .....std. ID received the assignment and read the comments..... (Signature/date)

Academic Honesty Acknowledgement

"I **Cheryl Toh Qiao Rou** (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."

…….3/8/2023 (Student's signature / Date)

Table of Contents

1.0 Introduction

1.1 Game Overview

The project implements a traditional Chinese card game, Big Two, also known as Chor Dai Di. Big Two is a popular and strategic game played with a standard deck of 52 playing cards, involving 4 players in the game. The goal of the game is for players to strategically play their cards in valid combinations and be the first to get rid of all their cards. The game follows a hierarchical order of cards, with the highest-ranked card being the 2 of Spades, followed by Aces, down to 3s. The suits hold powers by having Spades as the highest power, followed by Hearts, Clubs and Diamond. Players take turns playing a valid card combination, and the next player must play a higher-ranking set of cards with the same combination or pass their turn. The player with the 3 of Diamonds starts as the "king" and can choose any valid combination to play initially.

The king decides combinations to be played in the round. Players take turns to play specific card combinations decided by the king, such as singles, doubles, triples, flushes, straights, full houses, and straight flushes, while attempting to beat the previous cards played. The king changes when all players including the previous king have passed their turn, the last player that dealt a valid combination will become the next king, having the privilege to decide what type of combo they want to play. Throughout the game, players must strategize and observe the cards played by their opponents to determine the most optimal time to play their best combinations.

Big Two offers an engaging and entertaining card gaming experience for players of all skill levels. This project effectively brings this traditional card game to life in a digital form, allowing players to enjoy and compete against each other in a fun and immersive gaming environment.

1.2 Motivation

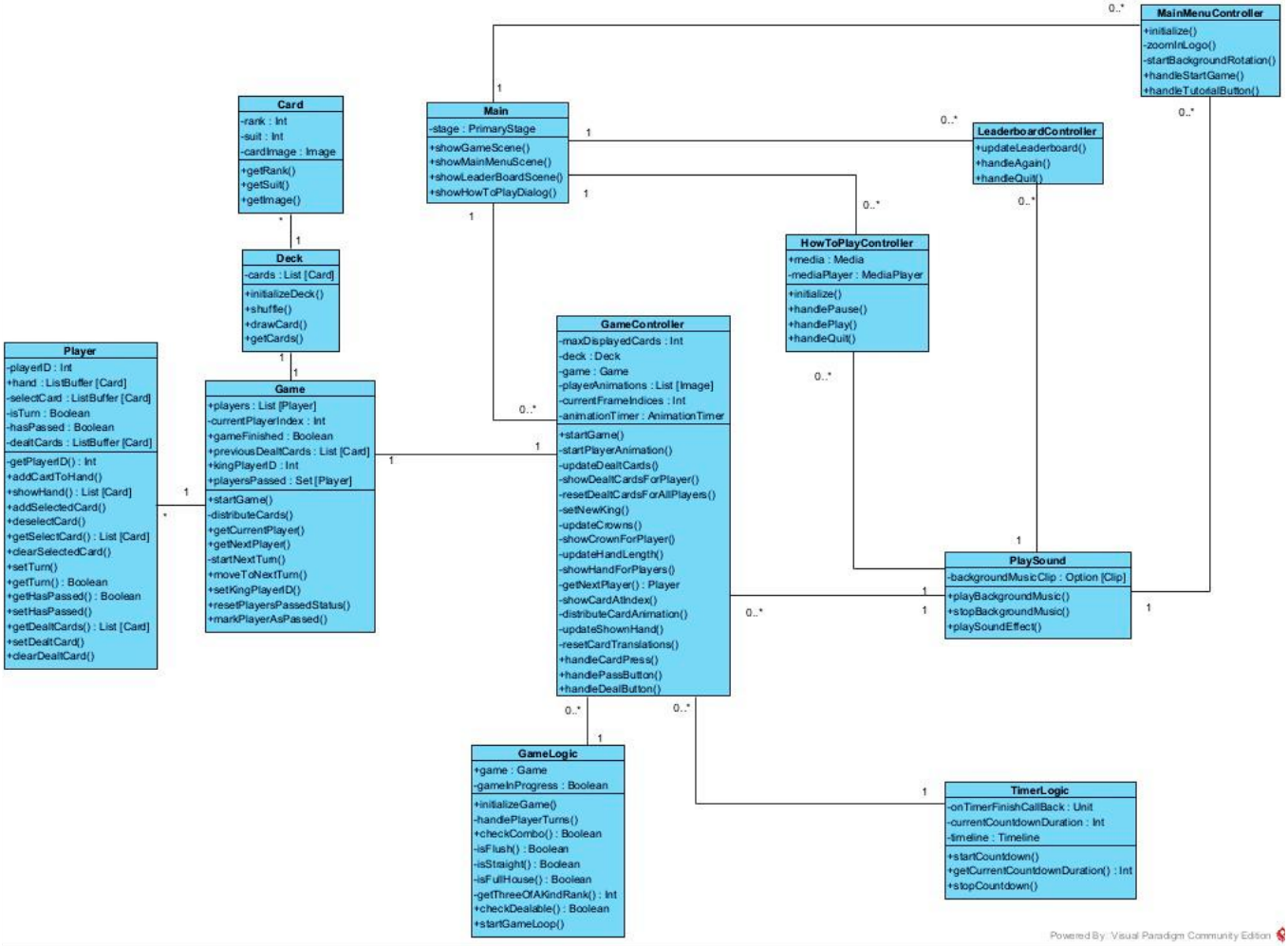
Digitalizing the gameplay and creating an interactive platform for Big Two was driven by a desire to share this game with a broader audience and recreate the sense of affinity and excitement. By bringing Big Two into the digital sphere, I aimed to provide players with a fresh and modern experience, offering a more visually captivating user interface that would increase the overall gaming enjoyment.

One of the main objectives of the project is to make Big Two more accessible to players who might not be familiar with physical poker cards or the rules of Big Two. By incorporating intuitive controls and interactive gameplay features, the digital version of Big Two aims to welcome newcomers and make them feel at ease while exploring the captivating world of Big Two. To achieve this, the project's focus was not solely on replicating the basic mechanics of the game but also on enhancing the player experience through a captivating and user-friendly interface. The goal was to create a visually appealing and immersive UI that would engage players and make their Big Two gaming sessions even more enjoyable.

Ultimately, the motivation behind this project goes beyond developing a digital card game; it is a heartfelt tribute to the cultural significance and emotional connections that games like Big Two hold in our lives. It is my hope that this digital adaptation will bring the same joy, laughter, and bonding moments to players as it did to me and my loved ones during those cherished gatherings.

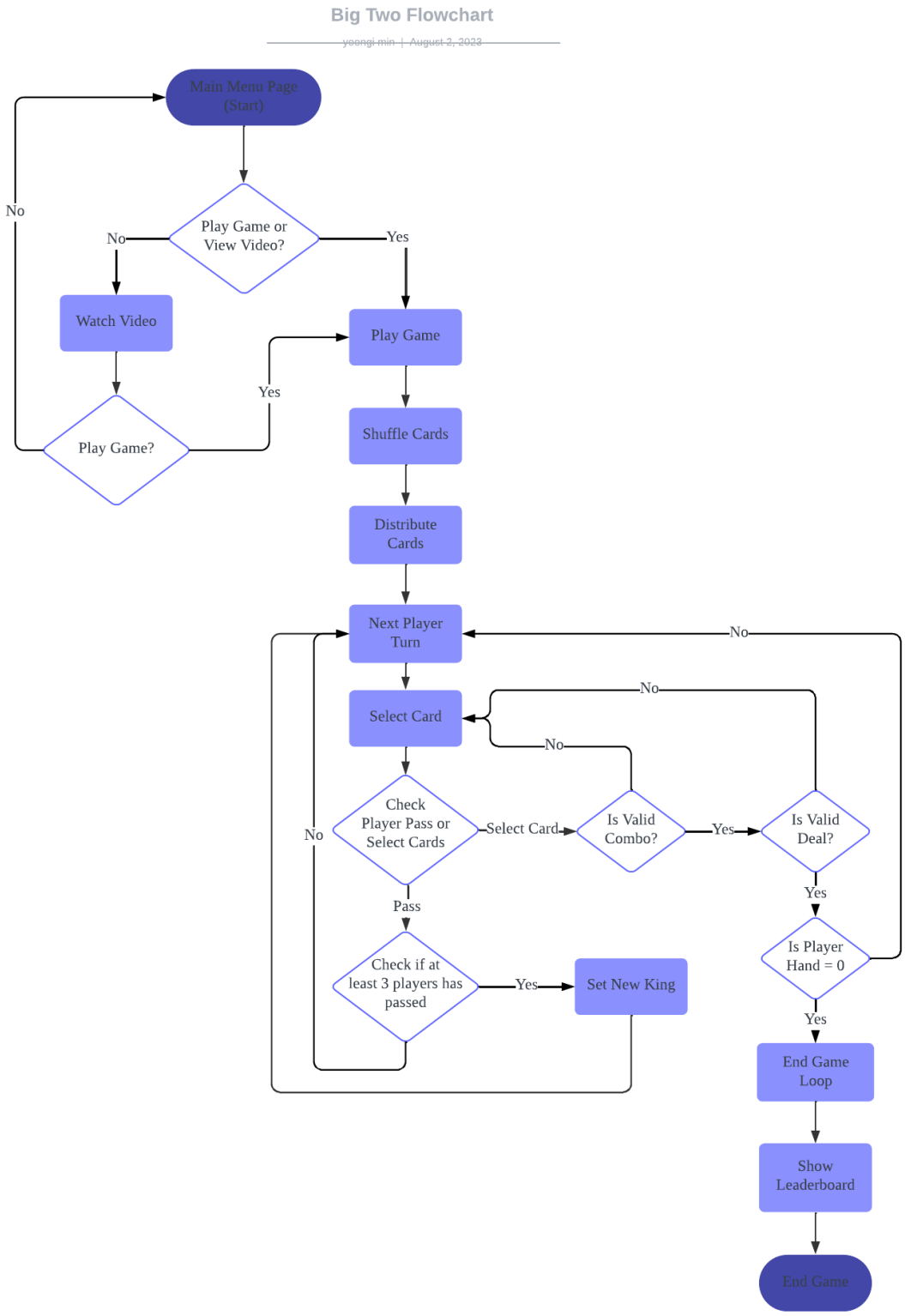
2.0 UML Diagrams

2.1 Class Diagram



Link to pdf version (for clearer view):
https://drive.google.com/file/d/1gLlEAh7EUBbYFU9OVfqk7spyVaKwfVMD/view?usp=sharing

2.2 Flow Chart



Link to pdf version (for clearer view):
https://drive.google.com/file/d/15cU5f82alwzyEWS_5gXyA-NIOR6P8i8p/view?usp=sharing

3.0 System GUI Usage

3.1 User Interface

The Big Two game's graphical user interface (GUI) was carefully designed to create an immersive and enjoyable gaming experience. The main menu greets players with an engaging rotating background and vibrant logo, setting a lively and inviting tone. The use of bold colors and cool visual effects on the buttons adds to the excitement and encourages players to explore different features of the game. The fast-paced tutorial video with modern animations not only provides essential information but also keeps players entertained and eager to learn more.

Figure 1: Logo color palette

On the game page, a deliberate shift to a slower pace complements the strategic nature of the game. Traditional Chinese music plays in the background, creating a serene and meditative ambiance. The bubbly and round UI fonts add a touch of playfulness, reflecting the game's traditional roots. As players progress, the transition page and leaderboard maintain a minimalist vibe, striking a balance between the fast-paced main menu and the strategic game page. The use of light blue backgrounds introduces a calming tone, giving players a moment to reflect on their progress.

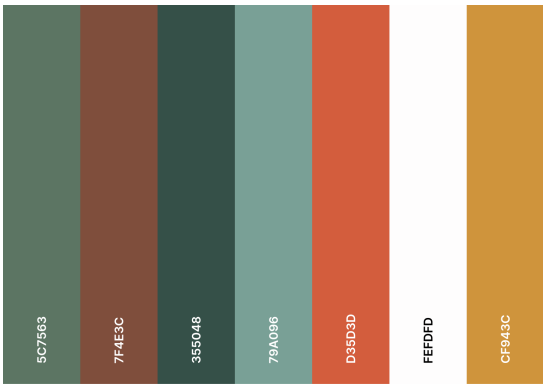


Figure 2: Game Page color palette

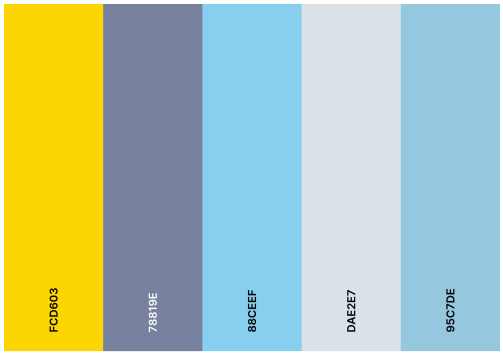


Figure 3: Transition and Leaderboard color palette

The seamless integration of contrasting elements, from the high-energy main menu to the more subdued and strategic game page, creates a cohesive and captivating user experience. It successfully blends modern aesthetics with traditional charm, catering to a diverse audience and providing a delightful journey into the world of Big Two.

3.2 User Experience

In terms of user experience, the flow of the game is designed to be smooth and enjoyable, allowing players to fully immerse themselves in the strategic gameplay of Big Two. The turn-based nature of the game ensures that players can take their time to strategize their moves, adding a layer of depth and critical thinking to each turn. After each player's turn, a transition prompt is introduced, creating a natural pause in the gameplay. This transition gives the next player ample time to prepare and strategize for their upcoming turn. It also adds a sense of anticipation and suspense as players eagerly await the next move.

Incorporating sound effects for each move made is another crucial element that enhances the user experience. When players make a move, corresponding sound effects accompany their actions. For example, when selecting a card, a satisfying sound effect signifies a card is selected. These sound cues not only provide auditory feedback to players but also serve as positive reinforcements, making the gameplay more engaging and immersive. Additionally, sound effects during transitions and when cards are dealt contribute to the overall atmosphere of the game, enriching the gaming experience.

By focusing on the flow of the game, strategic pacing, user-friendly interface, and engaging sound effects, the user experience of Big Two is carefully crafted to be both exciting and immersive. Whether players are enjoying the fast-paced action of the main menu or diving into the more thoughtful gameplay of the actual game, the user experience is designed to captivate and entertain, ensuring that players have a memorable and enjoyable time playing Big Two.

4.0 Architecture and Functionalities

4.1 Code Architecture Overview

The Big Two game follows a structured code architecture with different packages: view, controllers, entities, and utils. Each package contains specific classes that serve distinct functions in the game. Below is an overview of the classes within each package and their respective functionalities:

Package	Classes/Traits/Objects/Abstract classes	Functionality
View	FXML files	Define the layout and appearance of the graphical user interface (GUI) components.
Controllers	MainMenuController	Manages the logo and background animations and user interface interactions.
	HowToPlayController	Manages the video player logic and user interface interactions.
	GameController	Manages the game logic and user interface interactions.
	LeaderBoardController	Handles the leaderboard display and updates.
Util	GameLogic	Implements game-related logic
	ComboLogic	Contains case objects which verifies if a given list of cards matches the specific combo type
	DealableLogic	Contains case objects that implement the logic for determining if card combination is dealable
	TimerLogic	Manages timers for player turns and countdowns.
	PlaySound	Handles sound effects during the game.
Entities	Card	Represents a playing card with a rank and suit.
	Deck	Represents a deck of playing cards and its operations.
	Game	Manages the overall game, players, and turns.
	Player	Represents a player with a hand of cards and their actions.

4.2 Program Functionalities

1. **Card Distribution**
   The Game class handles the distribution of cards to players at the beginning of each round. It ensures that each player receives the appropriate number of cards from the deck, setting up the game for fair and balanced gameplay.

2. **Combo Validation**
   The GameLogic class includes the functionality to validate whether the selected cards by a player form a valid combo that can be played as a move. If the combo is not valid, the player is not allowed to proceed, ensuring that the game rules are followed.

3. **Deal-ability Check**
   The GameLogic class implements the functionality to determine if the selected cards by a player can be dealt on top of the previous cards played in the round. It checks the current combo and compares it with the previous one, allowing or disallowing the deal based on the game rules.

4. **Background Music and Sound Effects**
   The PlaySound object provides functionalities to play background music throughout the game and sound effects when certain actions occur. It enhances the user experience and adds immersion to the game environment.

5. **Player Turns and Timer**
   The GameLogic class handles player turns and ensures that each player gets a fair chance to play their cards. The TimerLogic class is responsible for managing the countdown timer for each player's turn. Players have a limited time to make their move, creating a sense of urgency and excitement during gameplay.

6. **User Interface**
   The View package encompasses various functionalities related to the user interface, including the main menu, tutorial video, and game interface. The main menu allows players to start the game or access the tutorial. The tutorial video teaches players the game rules, and the game interface displays player hands and the cards played during gameplay.

5.0 System Screenshots

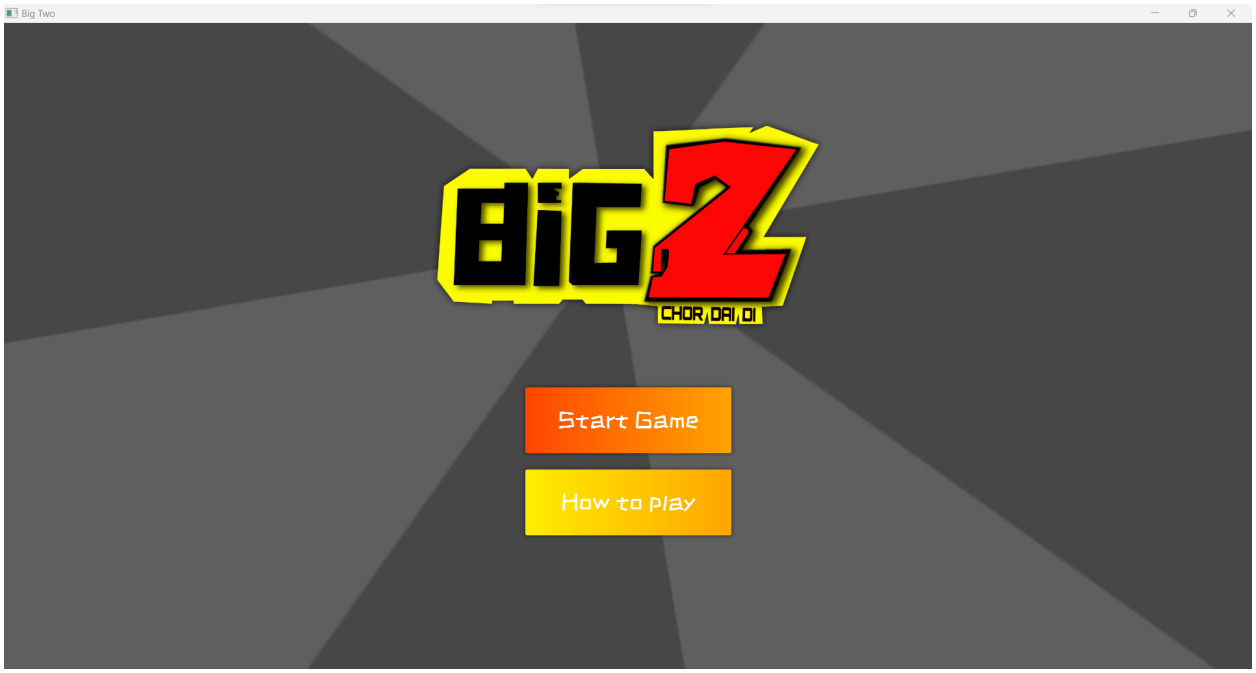


Figure 4: Main Menu page screenshot

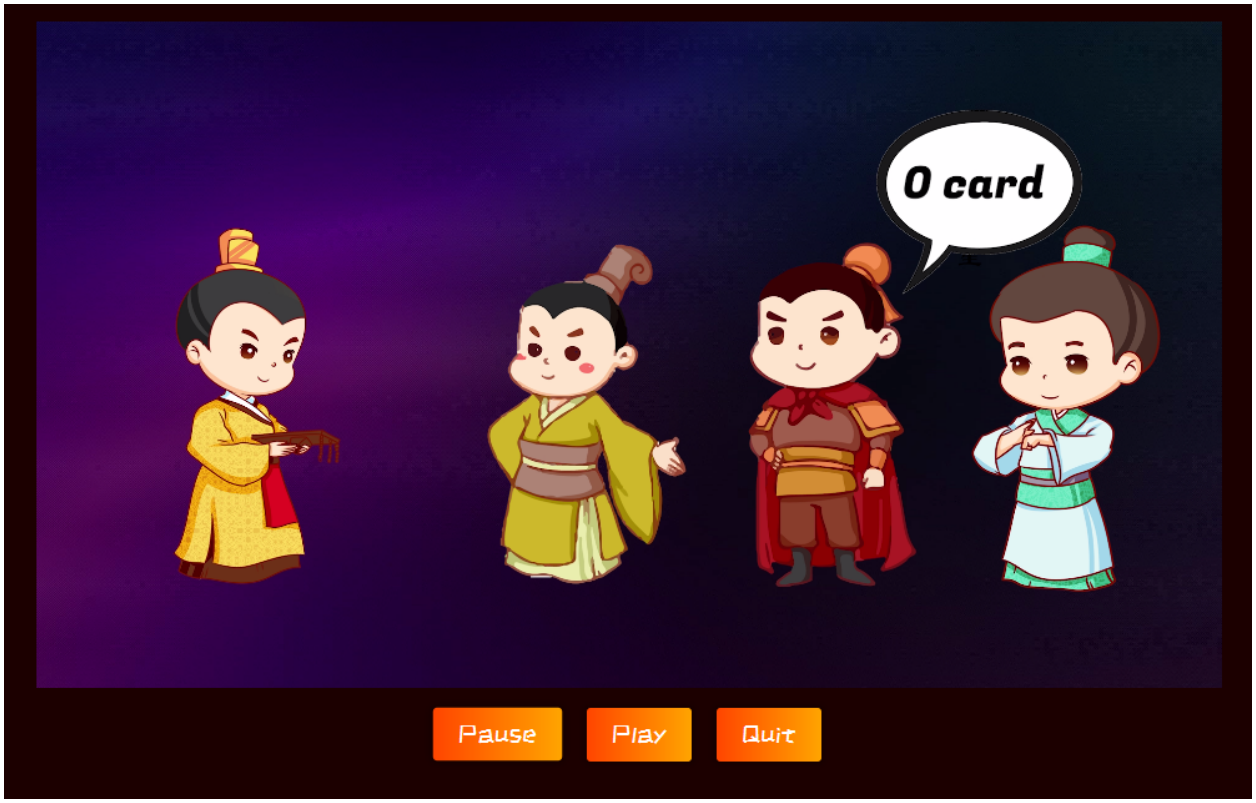


Figure 5: How to Play Dialog screenshot

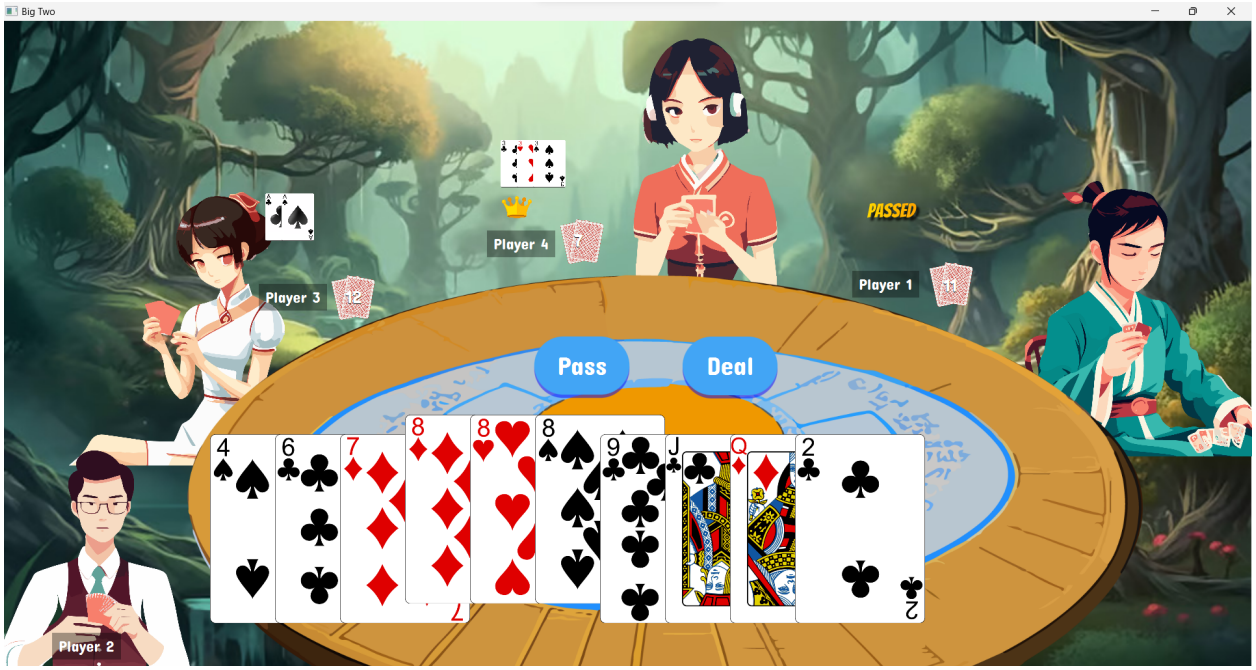


Figure 6: Game page screenshot

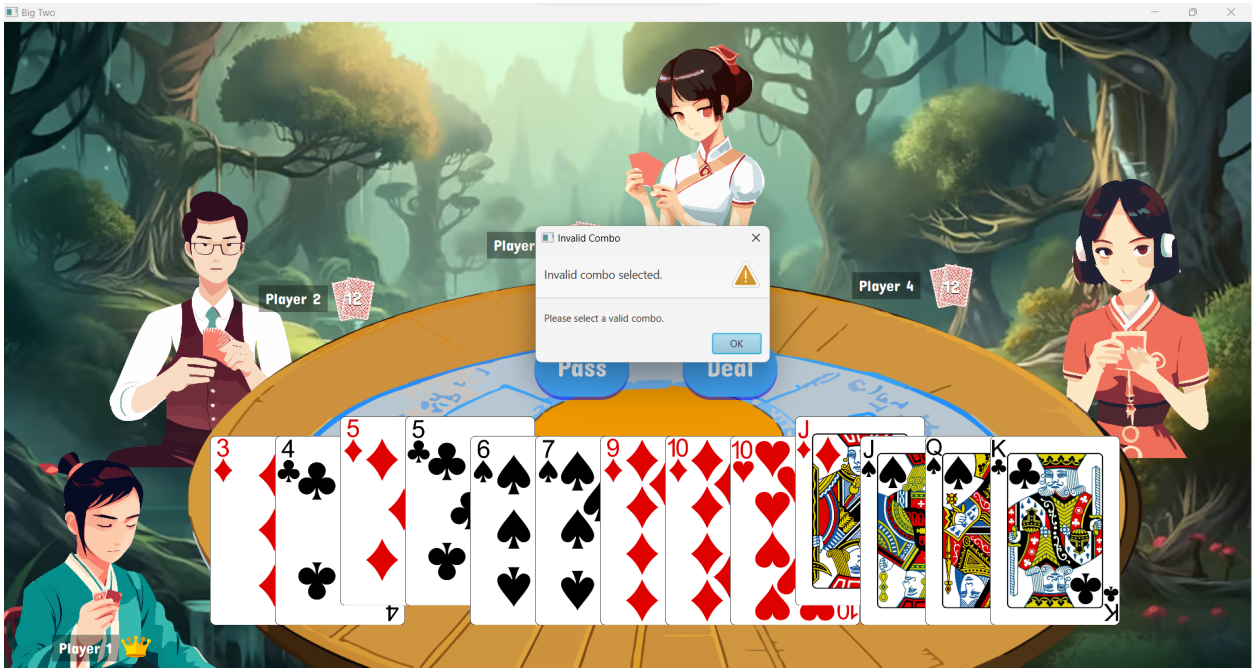


Figure 7: Invalid combination error dialog screenshot

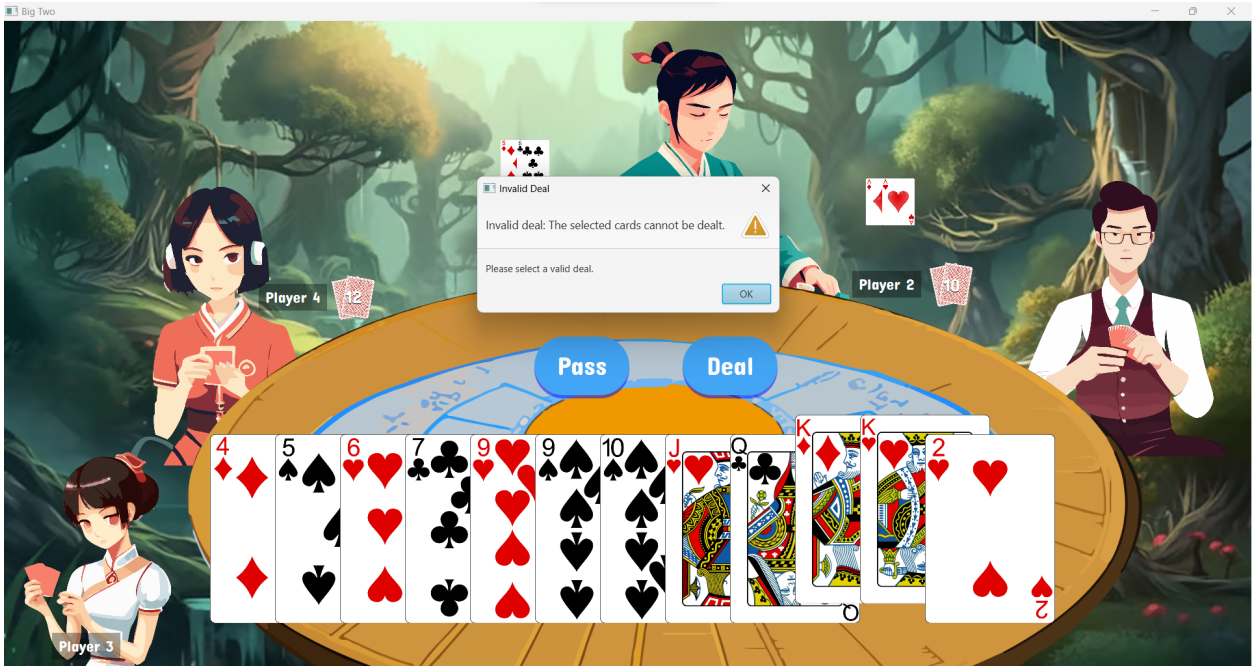


Figure 8: Invalid deal error dialog screenshot

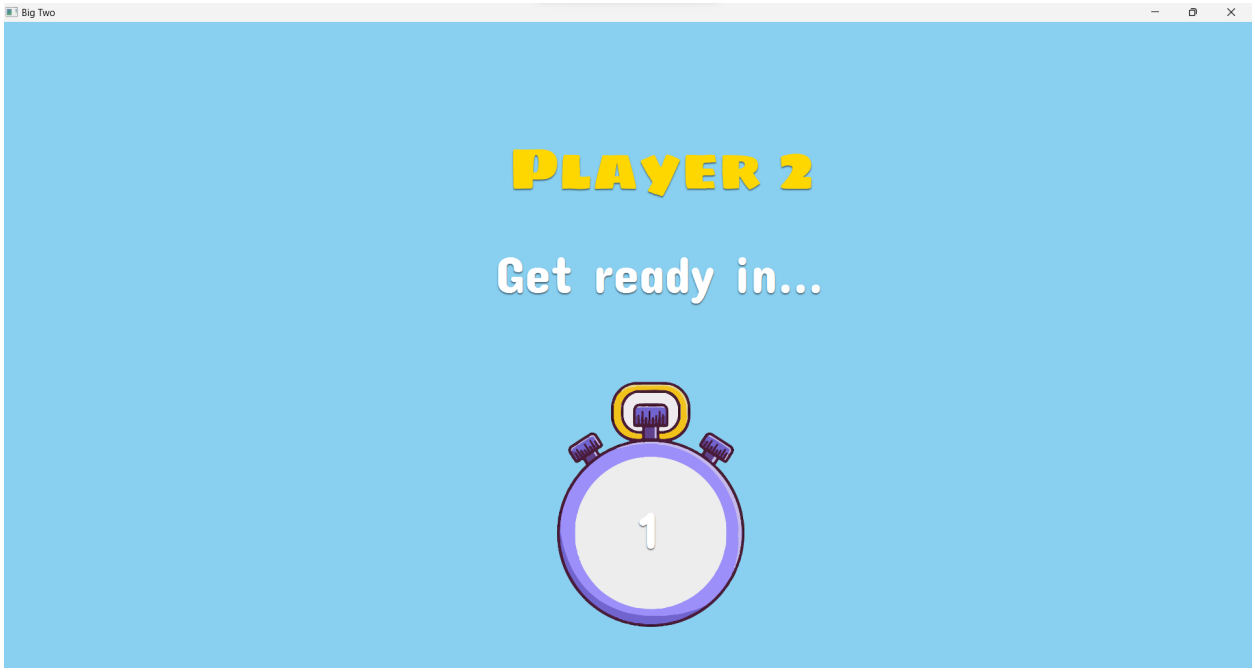


Figure 9: Transitions pane screenshot

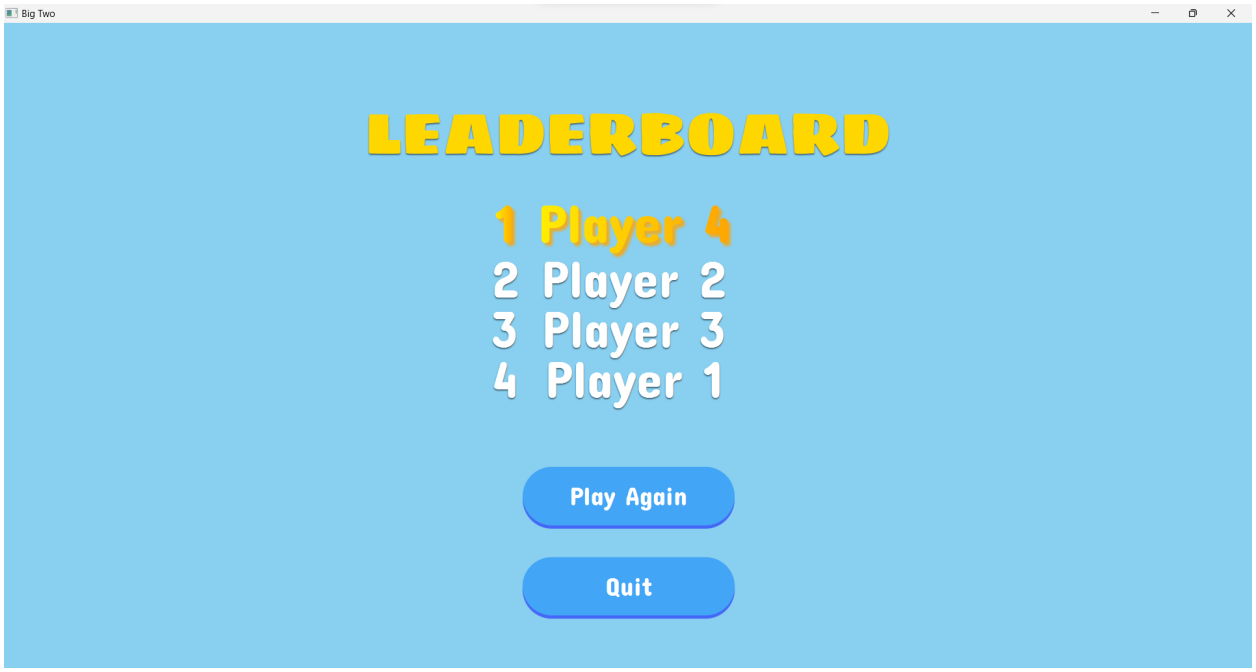


Figure 10: Leaderboard page screenshot

6.0 Personal Reflection

6.1 Application of OOP Concepts

6.1.1 Inheritance

Inheritance is a key principle of object-oriented programming that allows classes to derive characteristics and behaviors from a parent class. The *DealableLogic* trait defines a common interface for different deal-ability checks based on the previous round's and current round's card combinations. The objects *SingleDealable*, *DoubleDealable*, *TripleDealable*, *FlushDealable*, *StraightDealable*, *FullHouseDealable*, and *StraightFlushDealable* extend the *DealableLogic* trait, inheriting the method signature. Each object provides its implementation of the isDealable method, handling specific deal-ability conditions based on the combination types.

6.1.2 Polymorphism

Polymorphism is a powerful object-oriented programming concept that allows different classes to be treated as instances of a common superclass or interface. I applied polymorphism in the game through the use of traits and case objects. In the *GameLogic* object, the *checkCombo* function demonstrates the usage of polymorphism. By importing *ComboLogic._*, we gain access to the trait and its case objects. The function then uses pattern matching to identify the specific card combination among *Single*, *Double*, *Triple*, *Flush*, *Straight*, *Full* House, and *Straight Flush*. This polymorphic approach enhances code readability and maintainability as it abstracts the specific combo checks, making the code more modular and extensible.

6.1.3 Encapsulation

Encapsulation, a fundamental OOP concept, was applied throughout the project to protect the internal state of objects and ensure data integrity. Private attributes and methods were utilized to restrict direct access to class internals, promoting a more secure and maintainable codebase. Access to object properties was provided through well-defined getter methods, allowing controlled interaction with the objects.

6.1.4 Abstract Class

An abstract class in Scala is a class that cannot be instantiated directly. The abstract class *ComboLogic* in the provided code serves as a blueprint for different card combination objects. It defines abstract methods *checkCombo* and *comboType*, which must be implemented by its subclasses. Each object representing a specific card combination extends the ComboLogic abstract class and provides its implementation of the abstract methods.

6.2 Challenges and Solutions

During the development of the Big Two game, I encountered several challenges, and I implemented various solutions to overcome them.

1. **Linking Transition Scene and Game Page with Score Tracking**
   One of the challenges was smoothly transitioning from the main menu to the game interface while maintaining score tracking throughout the gameplay. I addressed this by creating a transition scene as an AnchorPane within the game page, sharing the same root and controller (gameController). This approach allowed for a seamless flow between the main menu and the game page while ensuring that the score tracking remained consistent across the transition.

2. **Updating Labels for Opponent's Dealt Cards and Passed Label**
   Updating the labels for opponent's dealt cards and the "passed" label was another challenge that required careful handling. After extensive debugging and testing, I successfully implemented the logic to update these labels accurately. The solution involved efficiently accessing and updating the necessary data structures within the game logic to reflect the opponents' moves and inform the players about the game state.

3. **AI Player Decision and Multiplayer Threading**
   Implementing the decision-making process for the AI player proved to be a complex task. I also faced the challenge of implementing multiplayer functionality with threading and synchronization requirements, which could be time-consuming given the project's constraints. As a result, I decided to focus on enhancing the single-player experience and creating a smooth transition scene instead of implementing full multiplayer functionality. This decision allowed me to provide a more polished and enjoyable single-player experience within the given time frame.

Overall, these challenges pushed me to think critically and creatively, and I am pleased with the solutions I implemented. Despite the constraints, the game now offers a smooth user experience, intuitive gameplay, and an engaging single-player mode. Future developments could focus on expanding the multiplayer capabilities and AI decision-making for an even more comprehensive gaming experience.

6.3 Strengths of the Project

1. **User-Friendly UI**
   One of the significant strengths of the project is its user-friendly and visually appealing user interface (UI). The contrast in UI design between the fast-paced main menu and the slower-paced game page caters to different player preferences, creating a well-rounded user experience.

2. **Game Logic**
   The game logic implementation is a strong aspect of the project. The various classes, such as Card, Deck, Player, and Game, effectively encapsulate their functionalities, promoting code modularity and reusability. The abstraction of card combination checks, such as flush, straight, full house, and straight flush, simplifies the code and enhances maintainability.

3. **Well-structured code**
   The code architecture and organization of classes demonstrate a high level of organization and clarity. Each package, such as Entities, View, Controllers, and Util, serves a distinct purpose, contributing to a coherent project structure. This well-structured codebase facilitates collaboration among team members, as well as future maintenance and updates.

4. **Extensibility**
   The project exhibits potential for extensibility. The code's modular design allows for straightforward integration of additional features and enhancements in the future. For instance, the game can be expanded to support multiplayer functionality and improved AI player strategies. The project's scalability ensures that it can evolve with new game modes, rules, or visual elements without compromising the existing functionalities.

6.4 Weaknesses of the Project

1. **Game State Persistence**
   A notable weakness of the project is the lack of game state persistence. Currently, if the application is closed or interrupted, the game progress is not saved, and players have to start over from the beginning. Implementing a save and load feature would greatly enhance the user experience, allowing players to continue their games from where they left off.

2. **Player Interaction**
   While the game offers a compelling single-player experience and allows multiplayer gameplay, the lack of real-time multiplayer functionality limits the depth of social engagement and live competition among human players. Implementing a real-time multiplayer feature would enhance the game's overall excitement and sense of community.

3. **Longer Rendering Time**
   With extensive UI and styling, the game might take longer to load all resources before appearing depending on hardware specifications and computational power. A loading page with good threading and synchronization can be introduced in the future in order to provide a more fluent gameplay experience to the users.

7.0 Conclusion

7.1 Summary

In summary, the project successfully implements the traditional Chinese card game, Big Two, offering players an engaging and strategic gaming experience. The project effectively utilizes object-oriented programming (OOP) concepts to create modular and organized code, ensuring encapsulation and abstraction in class design for maintainability and extensibility.The user-friendly UI and captivating visuals, combined with sound effects, contribute to an immersive gaming experience. The code architecture demonstrates well-structured and organized classes, making the project extensible and maintainable for future enhancements. However, the project could benefit from improvements in game state persistence, player interaction, and the addition of AI opponents to offer a more comprehensive gaming experience. Overall, the project achieves its objective of bringing the beloved Big Two game to a digital platform, preserving the cultural significance and enjoyment of this traditional card game.