

Intelligent Briscola

Intelligent Systems Engineering - UNIBO

A.A. 2023/2024

Andrea Zammarchi - andrea.zammarchi3@studio.unibo.it

Introduction

Introduction

Enhancing Briscola with Intelligent Agents

Digital Version of Briscola:

- Briscola is a traditional Italian card game.
- Transition to a digital format for broader accessibility.

Main Aim:

- Develop a dynamic and challenging gameplay experience.
- Intelligent agents simulate human-like strategies.

Key Features:

- Different levels of bot intelligence (stupid, normal, intelligent).
- Real-time interaction through a user-friendly GUI.



Objectives

Objectives

Goals

- **Implement Multi-Agent System:** Develop different levels of intelligence for a card-playing bot.
- **Autonomous Bot Behaviors:** Allow bots to make strategic decisions based on game context and difficulty level.
- **Game Logic Implementation:** Ensure correct rules, scoring, and mechanics of Briscola.
- **User Interface Development:** Create an intuitive and engaging graphical user interface (GUI).
- **Thorough Testing:** Conduct extensive testing for correctness and effectiveness.

Objectives

Expected Outcomes

- **Functional Simulation:** A fully functional Briscola game where players compete against intelligent bots.
- **Realistic Bot Behavior:** Bots adapt strategies based on selected difficulty level.
- **User-Friendly GUI:** Visually appealing and easy-to-use interface.
- **Engaging Gameplay:** An enjoyable and challenging experience that encourages repeat play.

Game Rules

Game Rules

Understanding the Basics of Briscola

Basic Rules of Briscola

- Score points by winning tricks with high-value cards.
- Minimum 61 points to win, 60 points results in a tie.
- 40 cards, 4 suits: Clubs, Coins, Cups, Swords.

Card Values

- **Ace:** 11 points
- **Three:** 10 points
- **King:** 4 points
- **Horse:** 3 points
- **Jack:** 2 points
- **Other Cards:** 0 points

Gameplay Mechanics

- **Starting the Game:** Briscola suit is established at the beginning.
- **Playing a Trick:** Highest card of the suit led or highest Briscola suit card wins the trick.
- **Winning a Trick:** Winner of the trick draws first in the next turn.
- **Game End:** Match ends when all cards are played.

Design

Design

Architecture and Design Choices

Architecture Overview

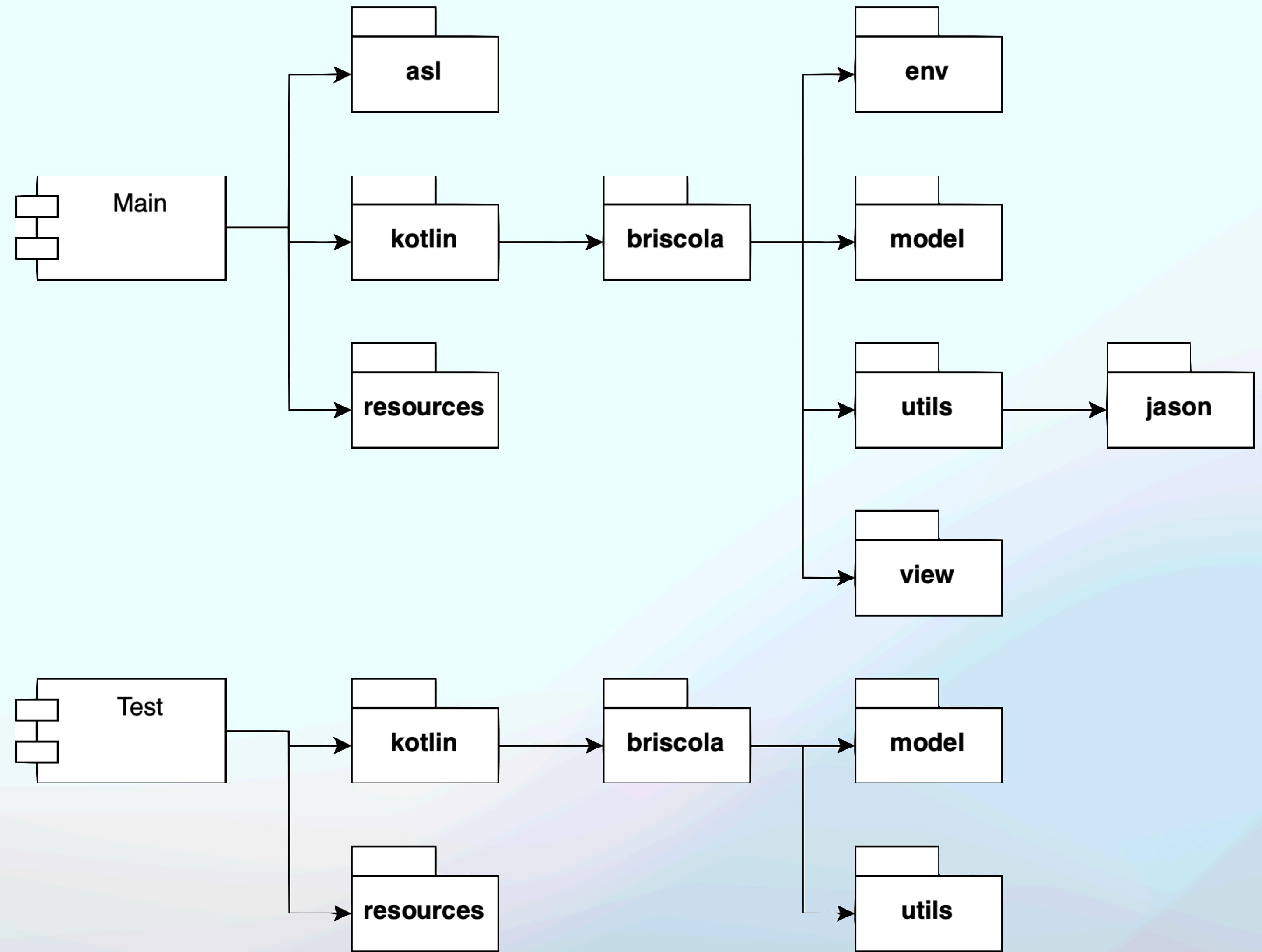
- **Centralized Multi-Agent System (MAS):**
Manages all agent activities and coordinates gameplay.
- **Core Components:**
 - **Kotlin:** Used for implementing the game logic.
 - **Jason:** Manages bot behaviors and decision-making.
 - **JavaFX:** Provides the graphical user interface.

Key Design Principles

- **Separation of Concerns:** Different packages for game logic, bot behaviors, and GUI.
- **Modularity:** Independent development and testing of components.
- **Scalability:** Easily add new features or intelligence levels without major refactoring.
- **Reusability:** Utility classes and functions for common tasks.

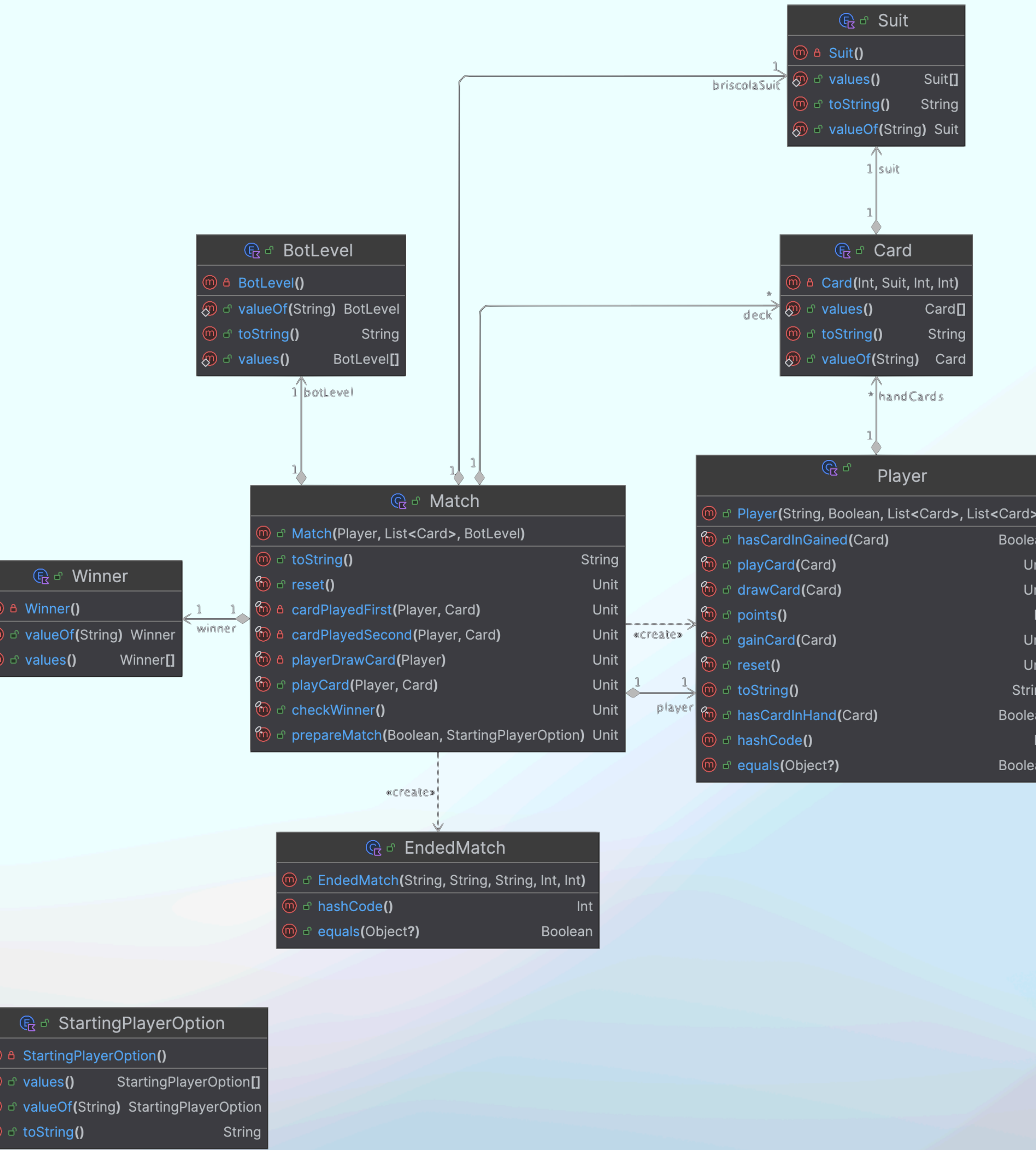
Design

Package Diagram



Design

Model Class Diagram



Bot Intelligence Levels

Bot Intelligence Levels

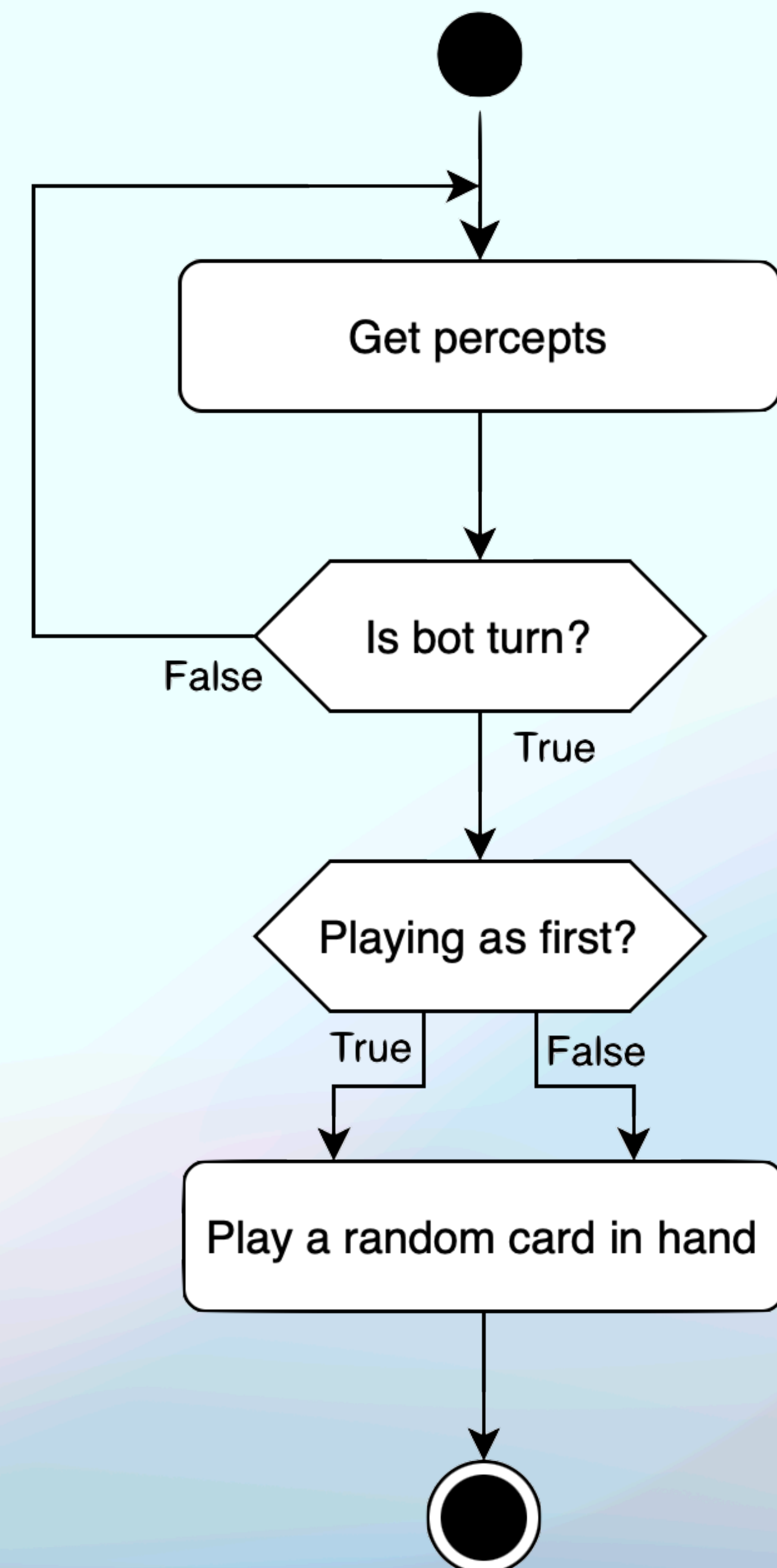
Stupid Bot

Strategy:

- Plays random cards without considering value or suit.

Characteristics:

- Unpredictable and lacks strategic decision-making.
- Provides an easy challenge for beginners.



Bot Intelligence Levels

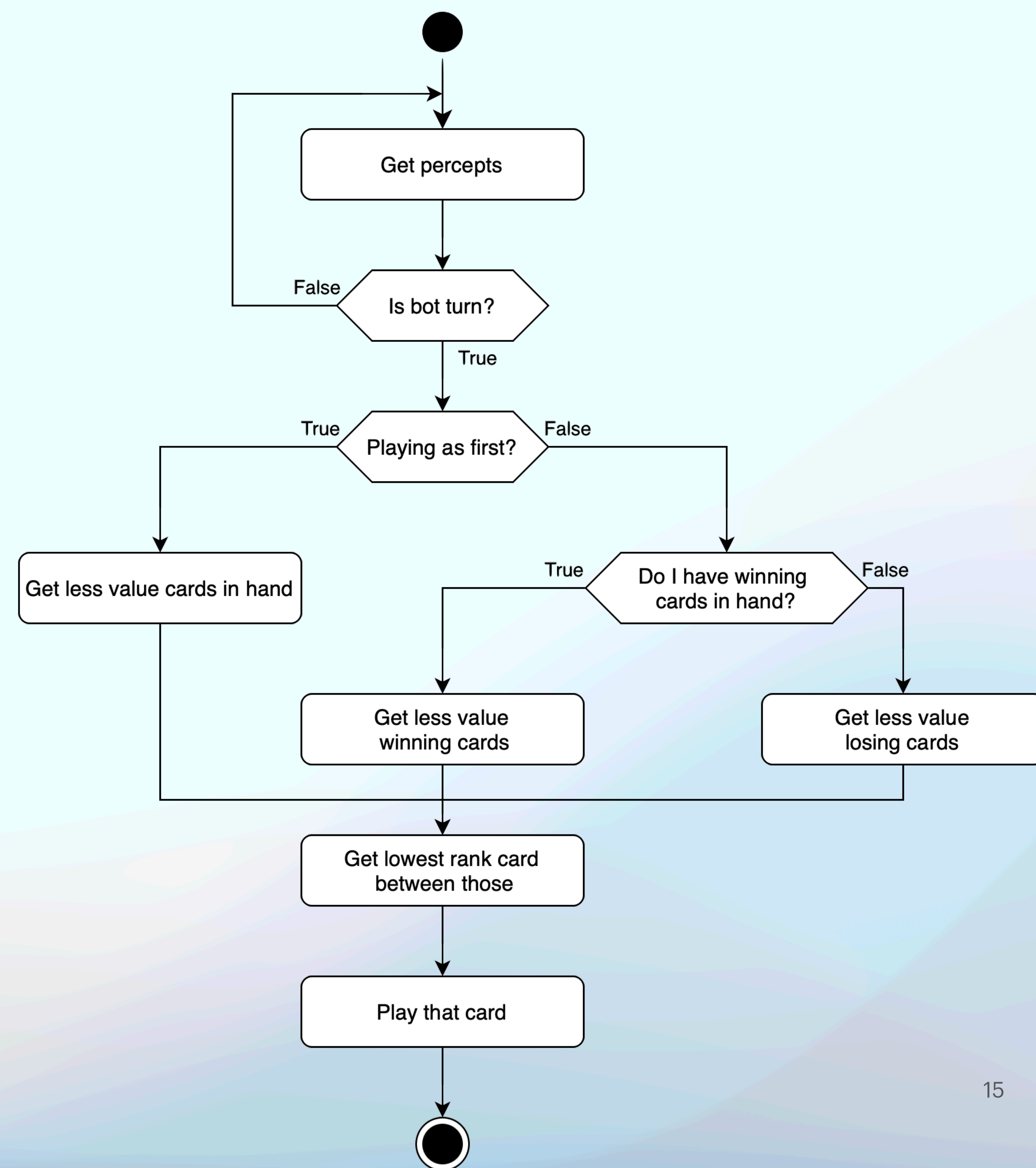
Stupid Bot

Strategy:

- Plays the lowest value card, prioritizing non-briscola suits.
- As the second player, tries to play the lowest card that can win the trick.

Characteristics:

- Basic level of strategy balancing between conserving valuable cards and winning tricks.
- Suitable for intermediate players.



Bot Intelligence Levels

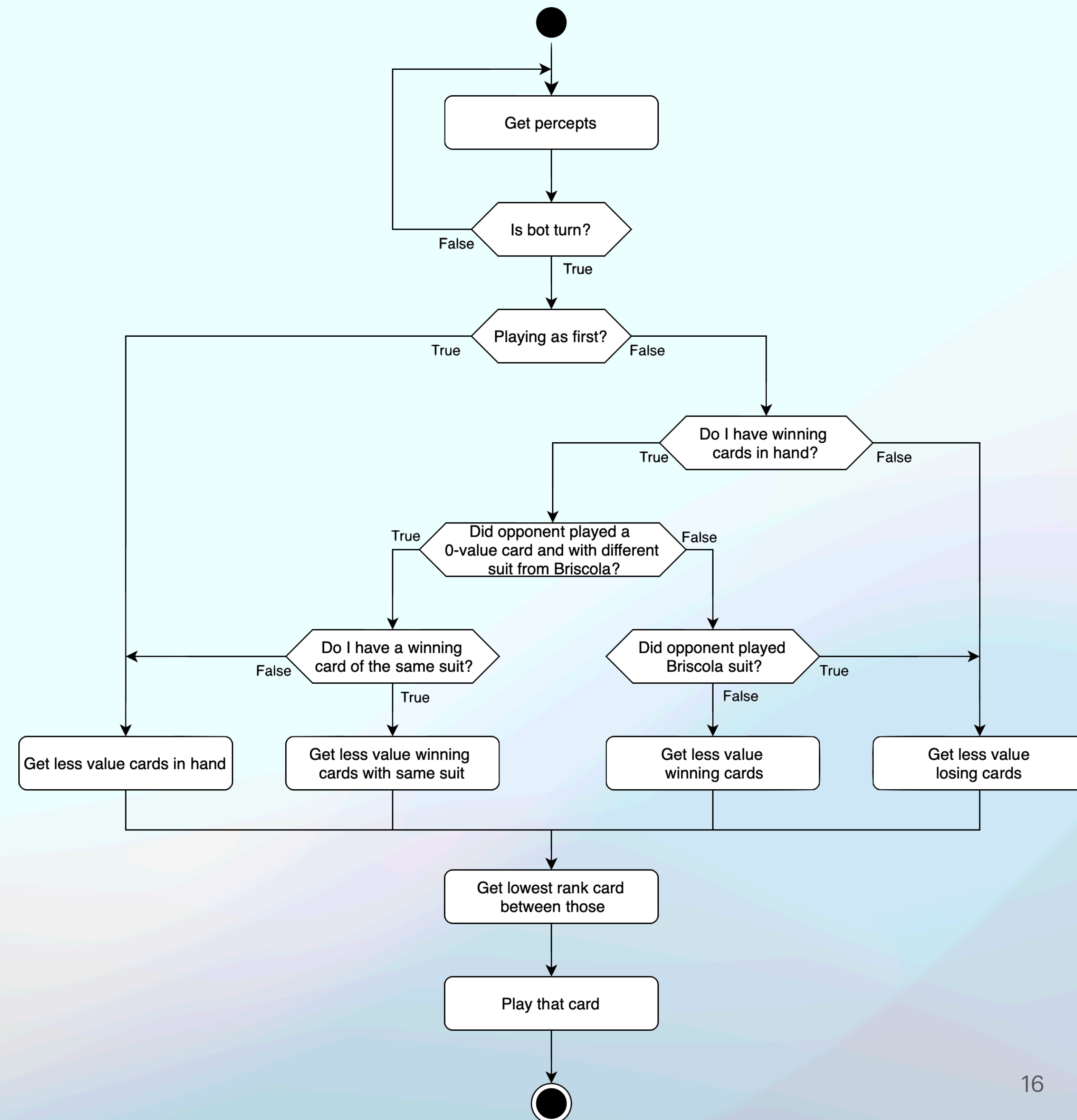
Stupid Bot

Strategy:

- Plays the lowest value card when leading, conserving stronger cards.
- Analyzes opponent's card and plays the optimal card to win or conserve based on the game state.

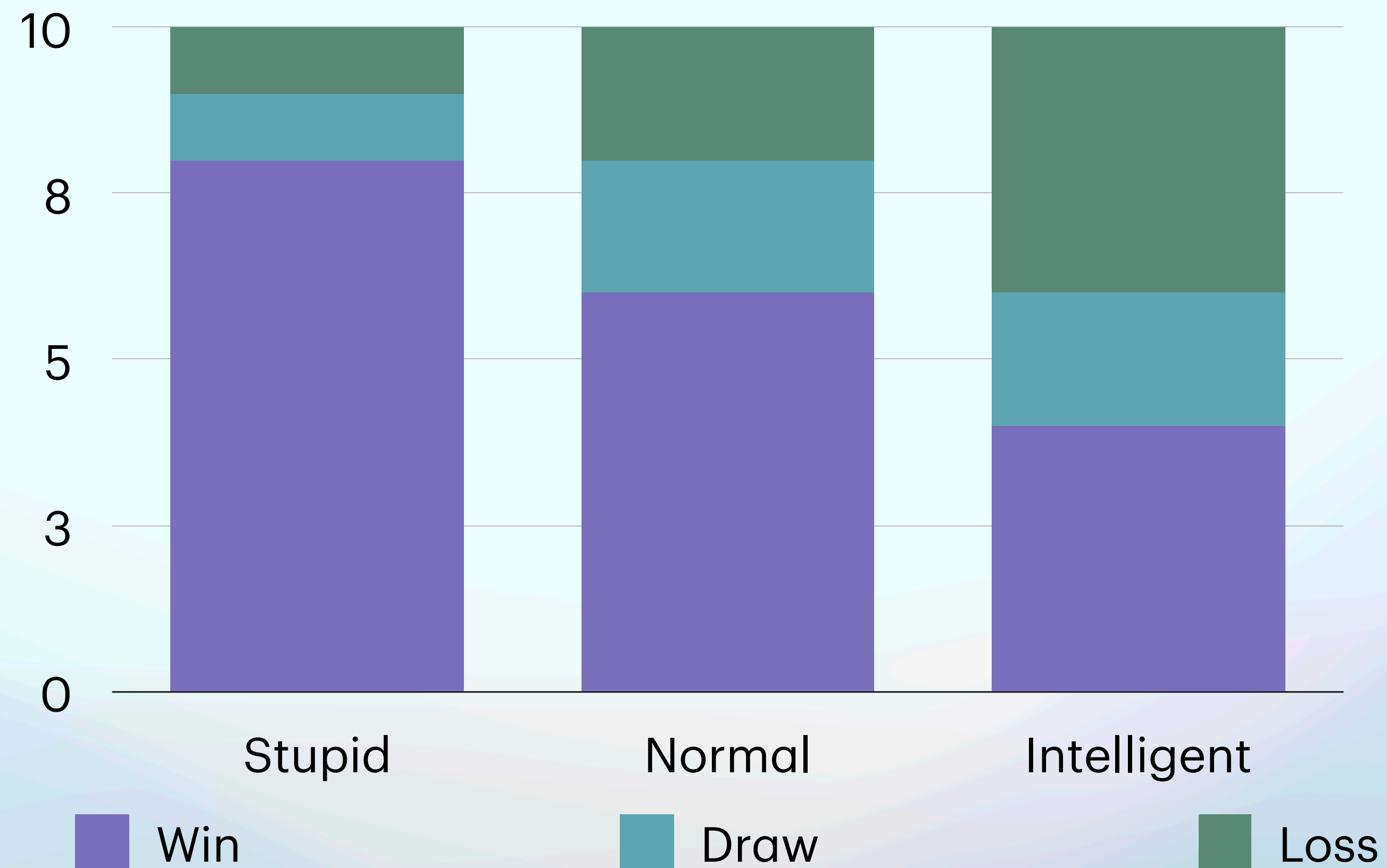
Characteristics:

- Sophisticated decision-making considering card values, suits, and game context.
- Provides a challenging experience for advanced players.



Bot Intelligence Levels

Strategies for Different Difficulty Levels



Development and Testing

Development and Testing

Tools, Strategies and Results

Development Tools

- **Kotlin:** Used for implementing the core game logic.
- **Jason:** Manages the autonomous behavior of the bots.
- **JavaFX:** Provides the graphical user interface.
- **Gradle:** Used for building and running the project.
- **GitHub Actions:** CI/CD pipelines for automatic testing and deployment.

Testing Strategy

- **Unit Testing:** Comprehensive unit tests for critical components using Kotlin Test. Focus on utils and model packages.
- **Integration Testing:** Verifying the interaction between different components.
- **System Testing:** Monitoring agent states and debugging.
- **Usability Testing:** Gathering user feedback to identify and resolve usability issues.

33

Unit Tests

100%

Pass Rate Unit Tests

92%

Model Test Coverage

Usage Examples

Usage Examples

Exploring the Game Views



Usage Examples

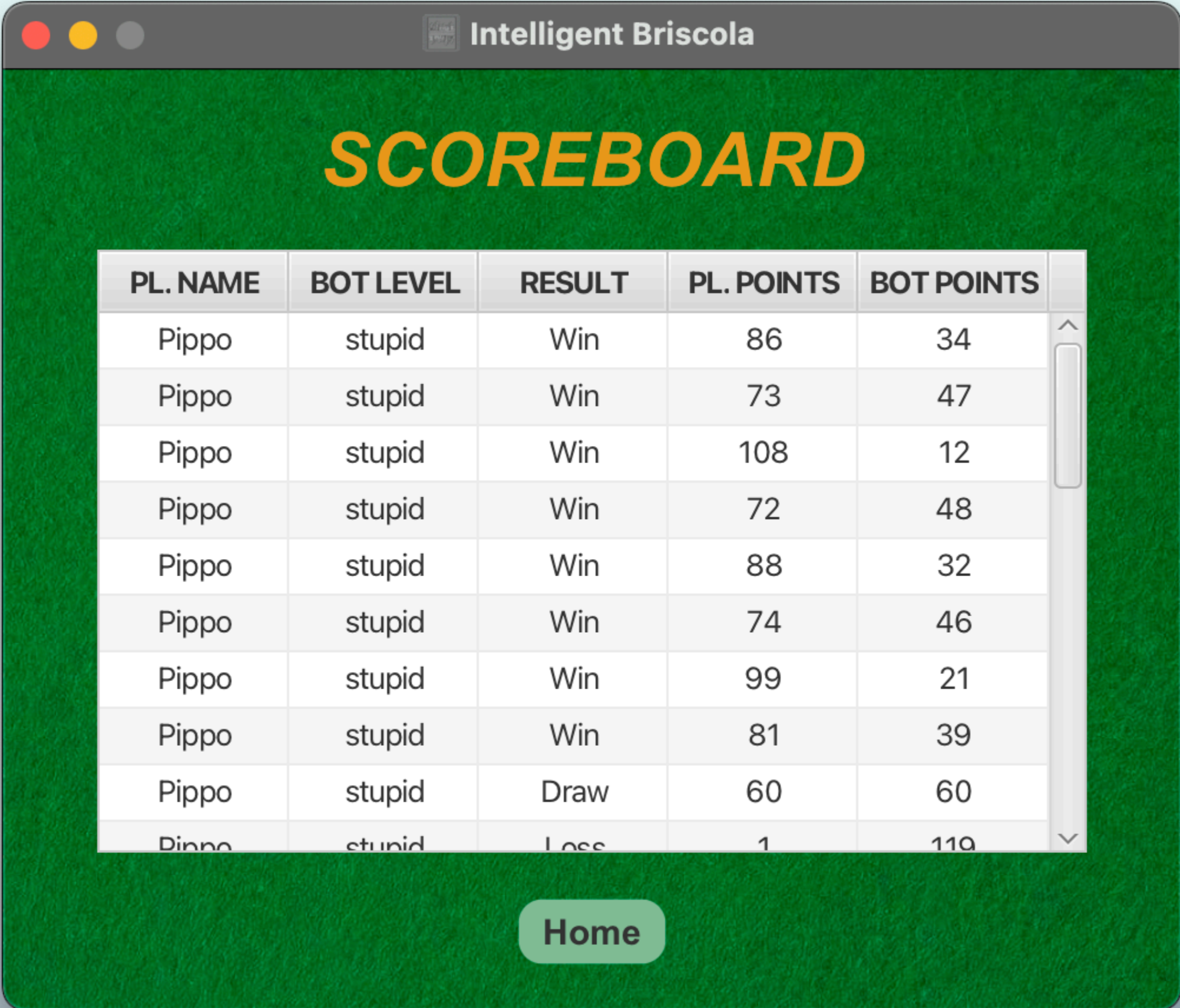
Exploring the Game Views



Match View

Usage Examples

Exploring the Game Views

A screenshot of a software application window titled "Intelligent Briscola". The window has a green textured background. At the top center, the word "SCOREBOARD" is written in large, bold, orange capital letters. Below this is a table with five columns: "PL. NAME", "BOT LEVEL", "RESULT", "PL. POINTS", and "BOT POINTS". The table contains ten rows of data. A vertical scrollbar is on the right side of the table. At the bottom center of the window, there is a green rounded rectangular button with the word "Home" in white text.

PL. NAME	BOT LEVEL	RESULT	PL. POINTS	BOT POINTS
Pippo	stupid	Win	86	34
Pippo	stupid	Win	73	47
Pippo	stupid	Win	108	12
Pippo	stupid	Win	72	48
Pippo	stupid	Win	88	32
Pippo	stupid	Win	74	46
Pippo	stupid	Win	99	21
Pippo	stupid	Win	81	39
Pippo	stupid	Draw	60	60
Pippo	stupid	Loss	1	119

Scoreboard

Conclusions

Conclusions

Summary and Future Work

Summary

- **Successful Implementation:**

- Developed a fully functional digital version of Briscola.
- Implemented different levels of bot intelligence for challenging gameplay.

- **User Experience:**

- Ensured a smooth and responsive gaming experience through thorough testing.

- **Technical Achievements:**

- Utilized a multi-agent system with Kotlin and Jason.

Future Work

- **Enhanced AI Strategies:**

- Develop more advanced strategies for the intelligent bot based on previous plays.

- **2v2 Support:**

- Introduce a mode for 2v2 gameplay to expand the game's capabilities.

- **Additional Features:**

- Explore adding new game modes, customization options, and user statistics tracking.

The End