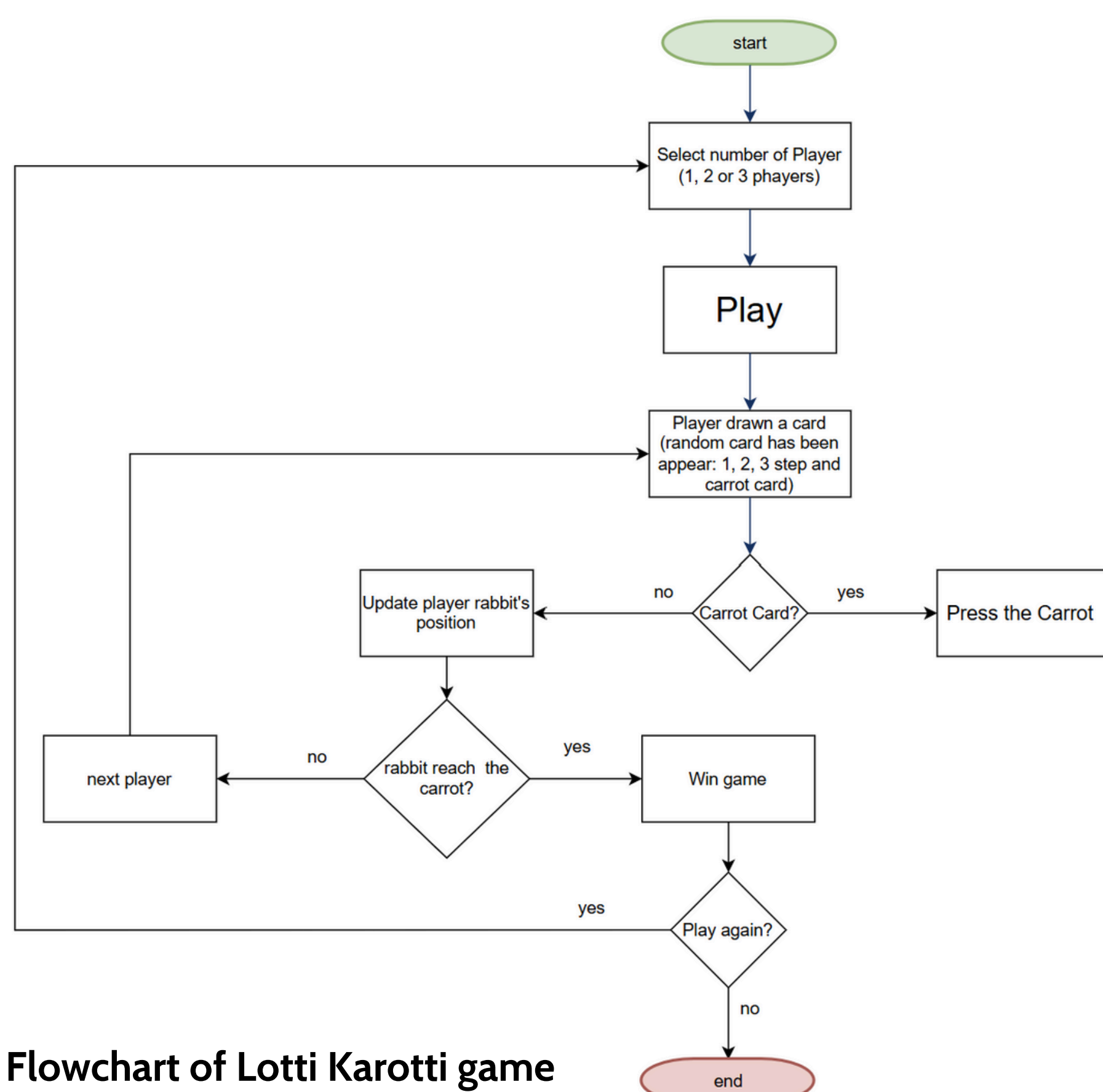


## LOTTI KAROTTI

HUYNH THI KIM QUYNH\_1584541, DUONG BAO NGOC\_1589562, SUJUNG LEE\_1365537

### ABSTRACT

The Lotti Karotti board game project was implemented by Java programming language and JavaFX. The goal was to create a digital adaptation of the classic board game, providing an interactive and visually engaging experience while retaining the original game's appeal.



Flowchart of Lotti Karotti game

### KEY ALGORITHM

The **Lotti Karotti** game integrates advanced algorithms to ensure dynamic and engaging gameplay. Below are three of the most critical algorithms:

#### 1. Turn Management Algorithm

A round-robin scheduling algorithm ensures fair player rotation, skipping players who cannot perform valid moves (e.g., no available rabbits). Validation mechanisms enforce game rules, such as requiring players to draw a card before moving, ensuring orderly gameplay.

#### 2. Pathfinding and Movement Algorithm

This algorithm calculates valid paths for rabbits based on constraints like occupied positions and the steps specified by drawn cards. It also handles end-game scenarios, such as detecting when a rabbit reaches the carrot, triggering the win condition[1].

#### 3. Randomization Algorithm[2]

Variability is introduced through random sampling algorithms. These determine player actions by drawing cards in the Card class and dynamically generating board holes in the Board class, adding unpredictability and challenge. These algorithms collectively support seamless gameplay, maintaining fairness, adaptability, and a rich user experience.

### GUI DETAILS



At the beginning, the user chooses how many players they want to play with. The "Start" button appears after two seconds.

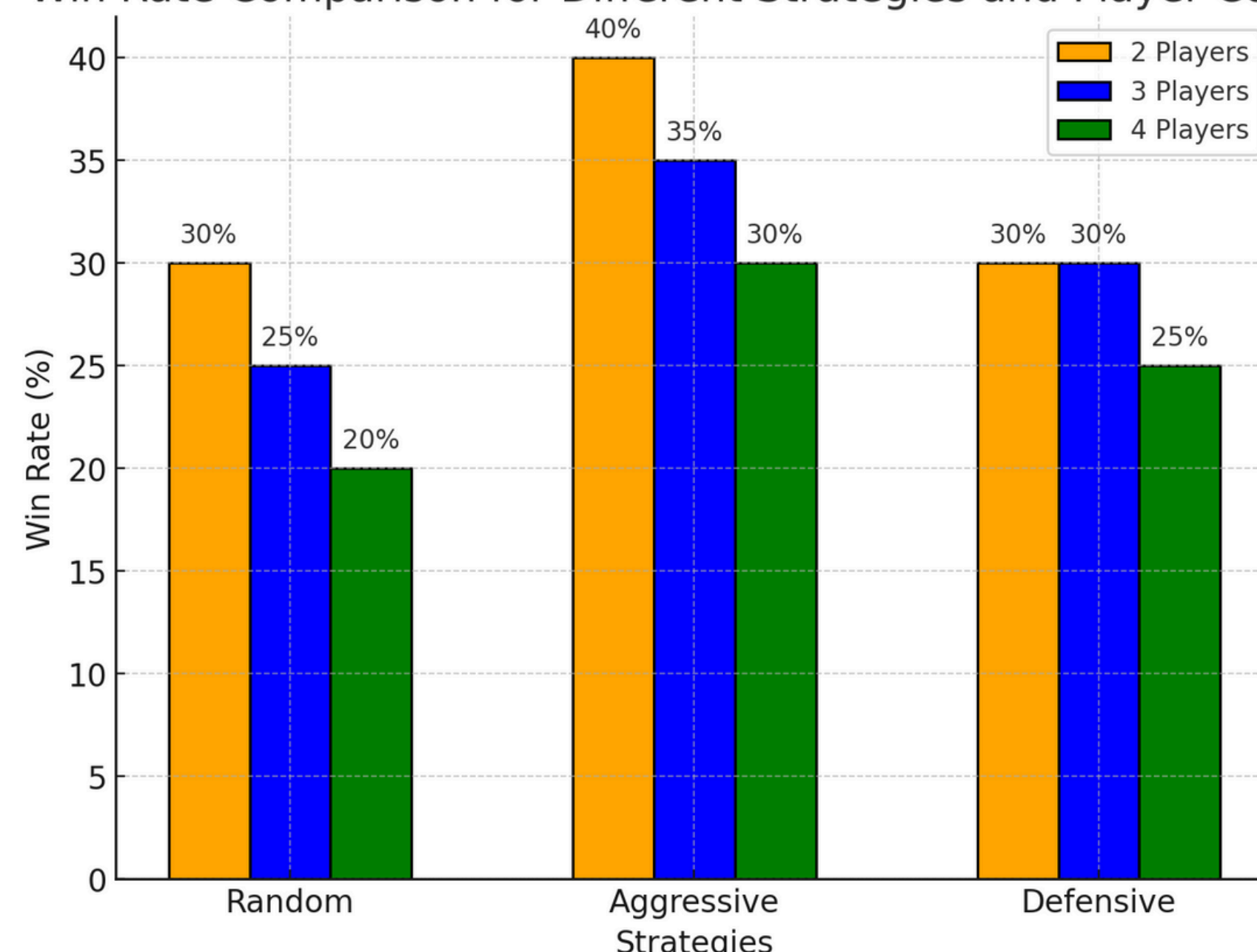
The bunny's path is shown by the blue circle. It contains "Draw Card", "Rules", "Quit" and a special "Carrot" button. The number of players is shown in the upper right corner.



The "Play Again" and "Quit" buttons are located in the winning view. The winning rabbit will be displayed in the middle.

### STATISTICAL TESTS

Win Rate Comparison for Different Strategies and Player Counts



According to the figure, the Aggressive strategy routinely wins the most games, especially when there are just two players (40%) since it has a far higher chance of obstructing the opponent. This method becomes somewhat less effective as the number of participants increases because of the increasing competition and unpredictability. The defensive strategy, on the other hand, is consistent across player counts (30% for two and three players, but lower to 25% for four), suggesting that it performs well under balanced circumstances but falters with additional players. In every scenario, the Random approach has the lowest victory rate, falling from 30% (2 players) to 20% (4 players), confirming the notion that lower results are the result of a lack of strategic forethought [3].

### CONCLUSION

Through organized task management, brainstorming, and collaboration using version control and bug-tracking tools, the team overcame these obstacles. The result is an innovative and entertaining game that recreates the essence of the original board game while opening the door to future enhancements, such as multiplayer features or new gameplay elements.

### REFERENCE

- [1] "javafx.animation (JavaFX 2.2)." Accessed: Jan. 24, 2025. [Online]. Available: <https://docs.oracle.com/javafx/2/api/javafx/animation/package-summary.html>
- [2] "Random (Java Platform SE 8 )." Accessed: Jan. 24, 2025. [Online]. Available: <https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>
- [3] manaiaclemim, "Lotti Karotti or Funny Bunny!," Tame The Board Game. Accessed: Jan. 27, 2025. [Online]. Available: <https://tametheboardgame.com/2016/12/21/lotti-karotti-or-funny-bunny/>