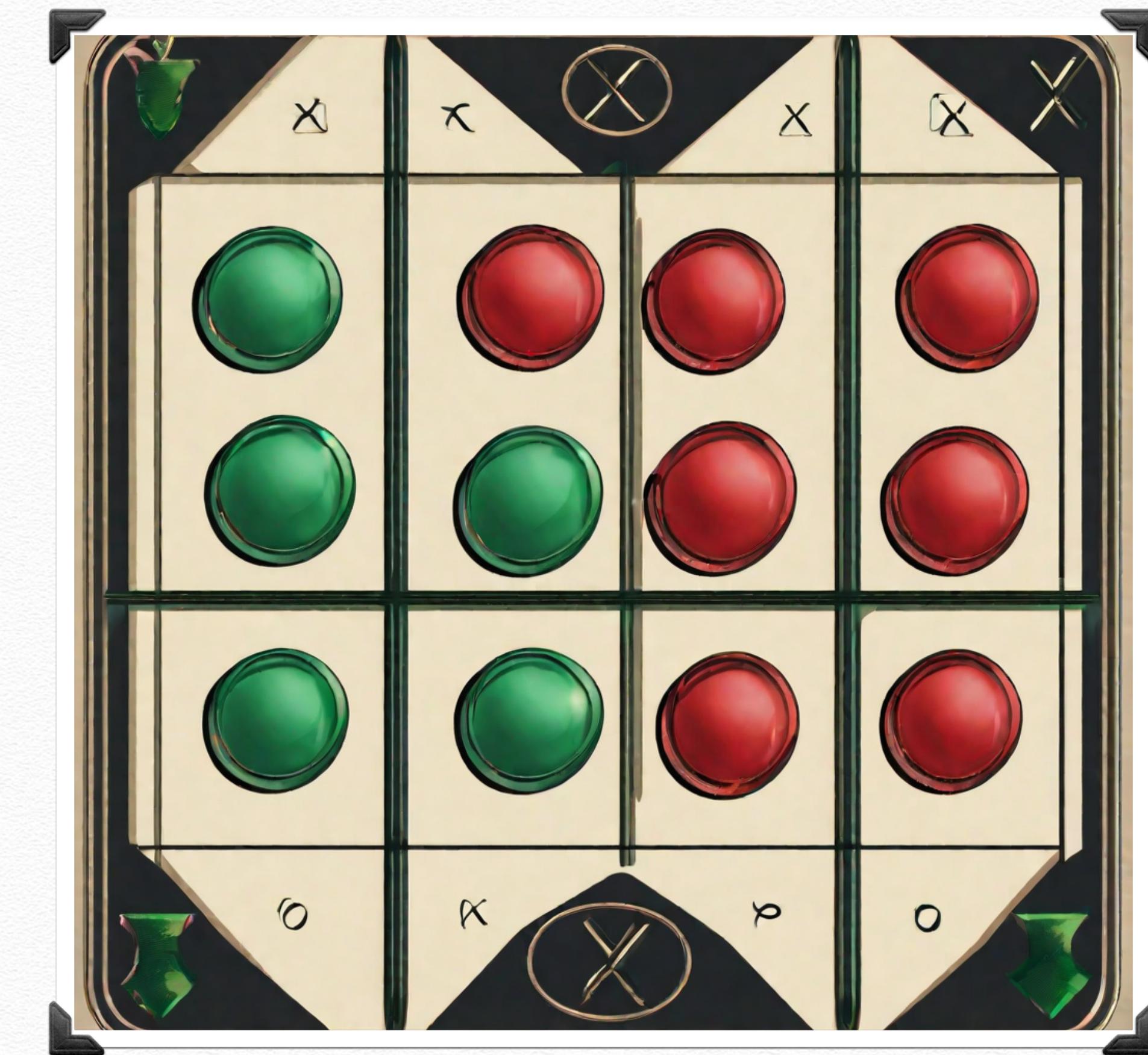


Tic Tac Toe Game with GUI

Presented by:
ISHA KUMARI

Introduction to Tic Tac Toe

Tic Tac Toe is a classic game played on a 3x3 grid. The objective of the game is to get three of your marks in a row, either horizontally, vertically, or diagonally. It is a game of strategy and critical thinking, and can be played by two players.



Game Rules

Players take turns placing their marks on a 3x3 grid. The first player to get three of their marks in a row wins. If all the spaces on the grid are filled and no player has won, the game is a draw.



User Interface Design

Game Board

The user interface should include a 3x3 grid to represent the game board. Each cell in the grid will correspond to a space on the Tic Tac Toe board.

Buttons

There should be a button for each space on the grid. These buttons will allow the user to make their move by clicking on the corresponding space.

Display Area

The user interface should include a display area to show the current player and the game status. This area will update as the game progresses, indicating which player's turn it is and displaying the winner or a tie when the game is over.

Game Logic

Win Condition

The game checks for a win condition by examining the board after each move. If a player has three of their marks in a row, column, or diagonal, they win the game.

Switching Players

After each move, the game switches to the other player. This ensures that each player takes turns making a move.

Handling Invalid Moves

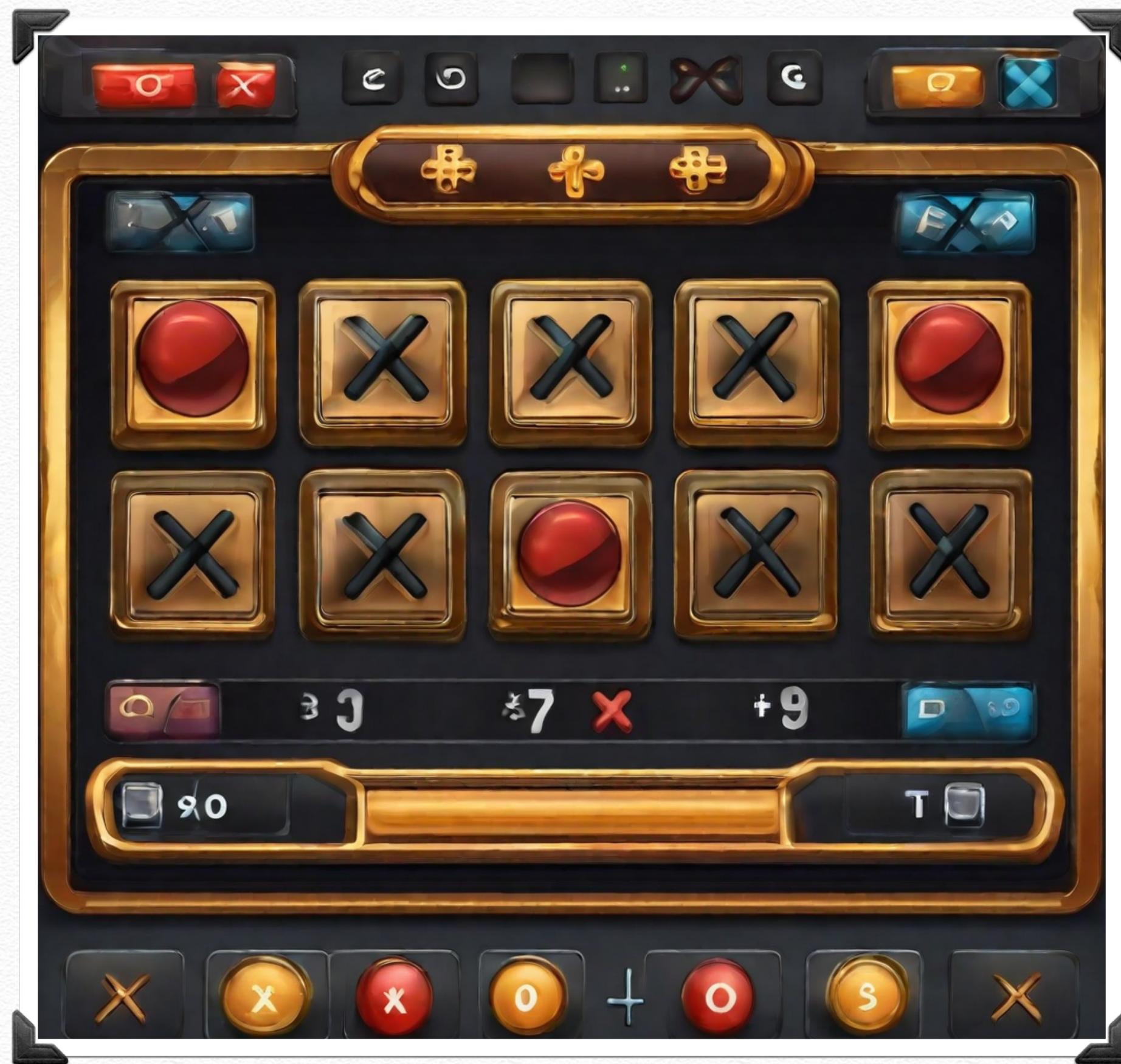
If a player makes an invalid move, such as selecting a cell that is already occupied or selecting a cell outside the board, the game will display an error message and prompt the player to make a valid move.



Implementation in Java

To create a Tic Tac Toe game with a GUI in Java, you will need to perform the following steps:

1. Create the game board: Initialize a 3x3 grid to represent the Tic Tac Toe board. You can use a 2D array or a list of lists to store the state of each cell.
2. Handle user input: Implement a method to handle user input for each move. This can be done using a mouse listener or button click event handler. Validate the input and update the game board accordingly.
3. Update the GUI: After each move, update the GUI to reflect the current state of the game board. This can be done by updating the text or image of each cell on the GUI.



Testing and Debugging

Testing Different Scenarios

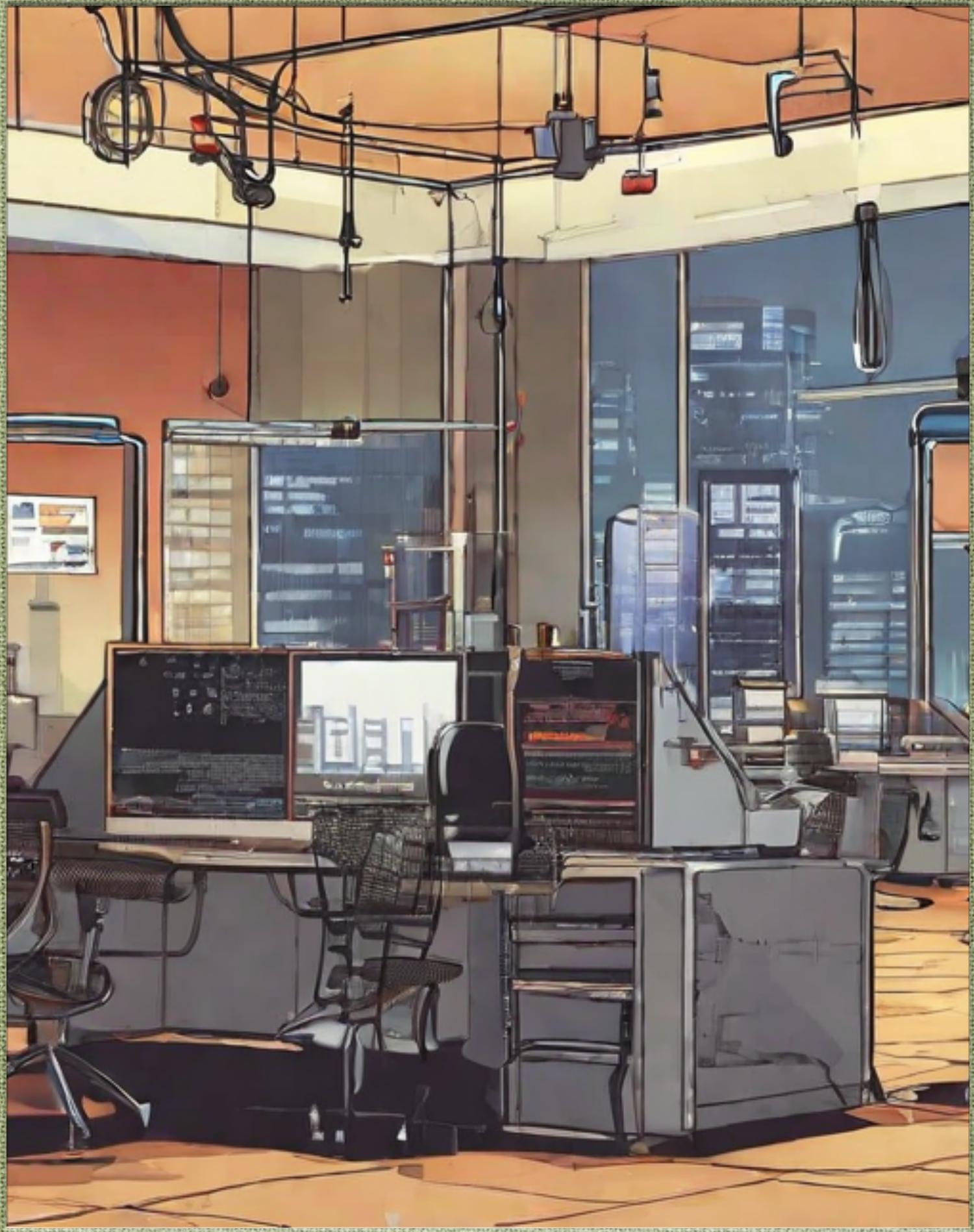
During the testing phase, we will simulate different scenarios to ensure that the Tic Tac Toe game functions correctly. This includes testing different combinations of moves, checking for winning conditions, and verifying that the game ends correctly.

Handling Edge Cases

We will also test the game with edge cases in mind, such as invalid input, out-of-bounds moves, and handling unexpected user behavior. By testing these edge cases, we can ensure that the game handles errors and exceptions gracefully.

Fixing Issues and Bugs

If any issues or bugs are discovered during testing, we will use the debugging process to identify and fix them. This may involve analyzing the code, stepping through it line by line, and using debugging tools to track down the source of the problem. Once identified, we will make the necessary changes to resolve the issue and ensure the game functions correctly.



Future Improvements

Improved User Interface

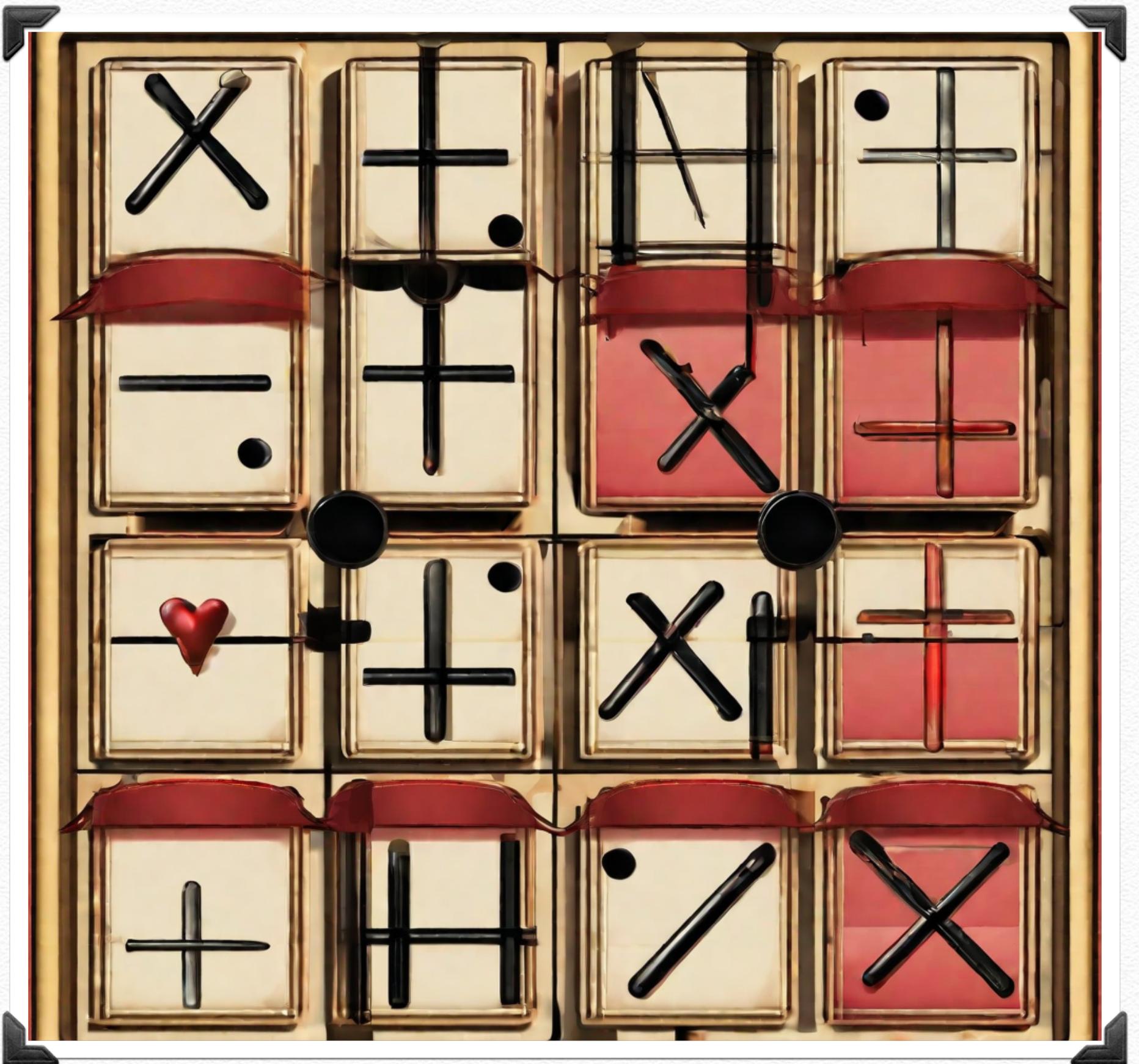
Enhance the visual design of the game with more appealing colors, fonts, and graphics.

Additional Features

Implement a computer player with different difficulty levels to challenge the player.

Optimized Game Logic

Improve the efficiency and performance of the game logic to reduce processing time and enhance the overall gameplay experience.



THANKYOU!