

TICTACTOE

Group: 2P

Subject: Object-Oriented Programming Lab

Semester: 2-2024

Prof: N.T.Nghĩa



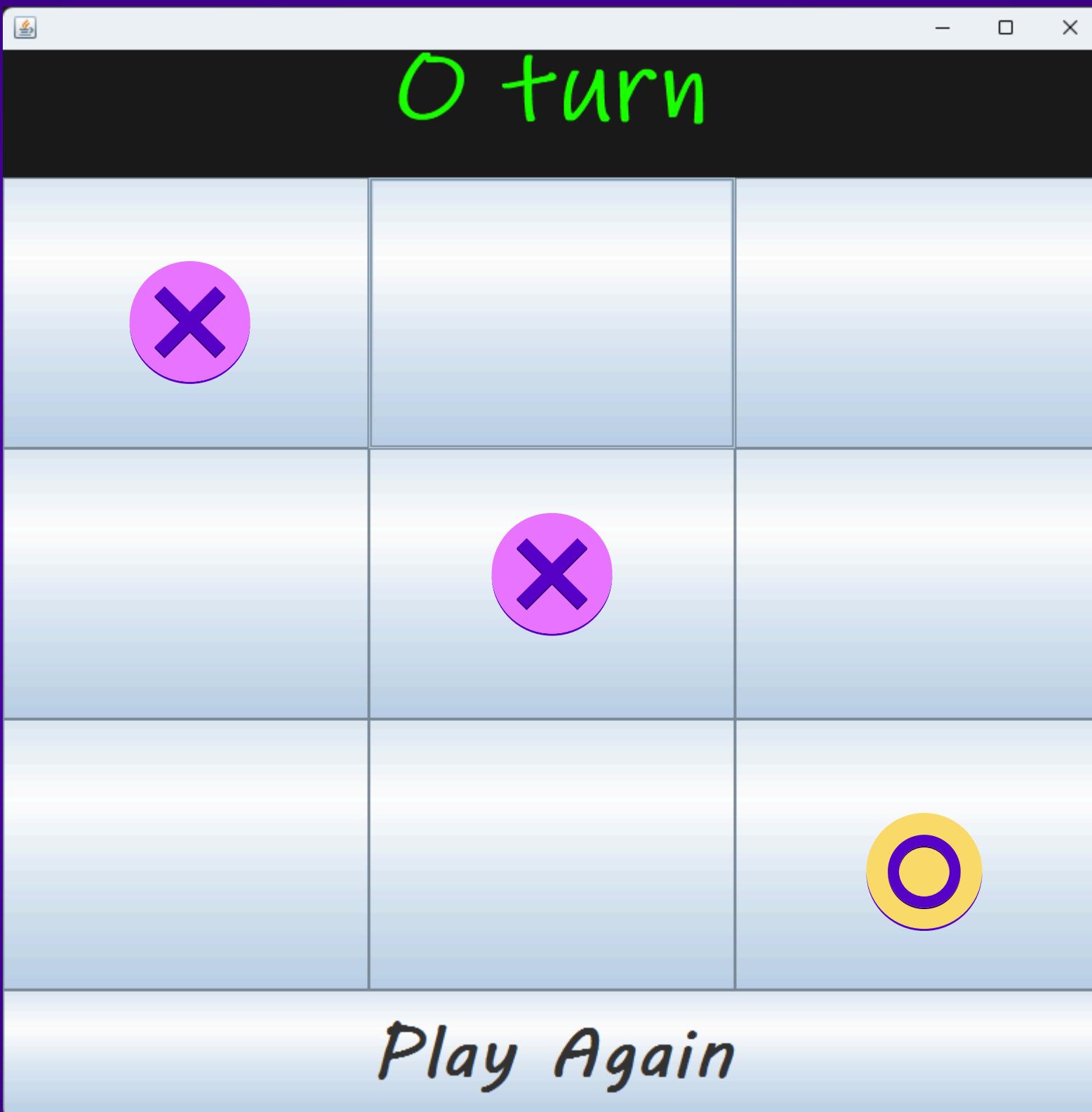
LET'S PLAY



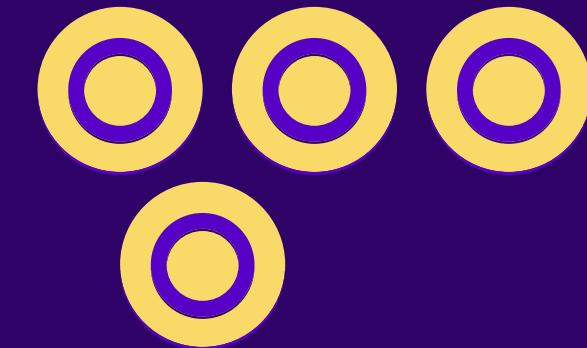
DRAG THE XS AND OS TO THE SPACE OF YOUR CHOICE.



Vũ Thành
Đạt
ITITIU21041



Nguyễn Thiện
Tín
EEEEIU21060



[1]

Contents

01

Introduction

02

Key Components

03

Implementation Details

04

Results

05

Conclusion

1. Introduction

- Two-player Tic-Tac-Toe game.
- Players take turns to mark X or O on a 3x3 grid.
- Check for winning conditions after each move.
- "Play Again" button to reset the game board.



2.Key Components

1. **JFrame**: Main window to hold all components.

2. **JPanel**: Containers for organizing the layout.

- **title_panel**: Displays the game title and current player's turn.
- **button_panel**: Holds the 3x3 grid of buttons.
- **bottom_panel**: Contains the "Play Again" button.

3. **JLabel**: Displays game status and current player's turn.

4. **JButton**: Represents each cell in the Tic-Tac-Toe grid and the "Play Again" button.

3. Implementation Details

1. Frame Setup:

- Created a JFrame named frame with a size of 800x800 pixels.
- Set the background color to dark gray (RGB: 50, 50, 50).

3. Implementation Details

2. Title Panel:

- Created a JPanel named title_panel with a BorderLayout.
- Added a JLabel named textfield to display the game title and current turn.

3. Implementation Details

3. Button Panel:

- Created a JPanel named button_panel with a 3x3 GridLayout.
- Added nine JButtons to the panel, each representing a cell in the Tic-Tac-Toe grid.
- Set font and styles for the buttons to ensure visibility and aesthetics.

3.Implementation Details

4. Bottom Panel:

- Created a JPanel named `bottom_panel` with a `BorderLayout`.
- Added a JButton named `playAgainButton` for resetting the game.

3.Implementation Details

5. Event Handling:

- Implemented the `ActionListener` interface to handle button clicks.
- Defined actions for marking X or O, switching turns, and checking win conditions.
- Added functionality to reset the game board when the "Play Again" button is pressed.

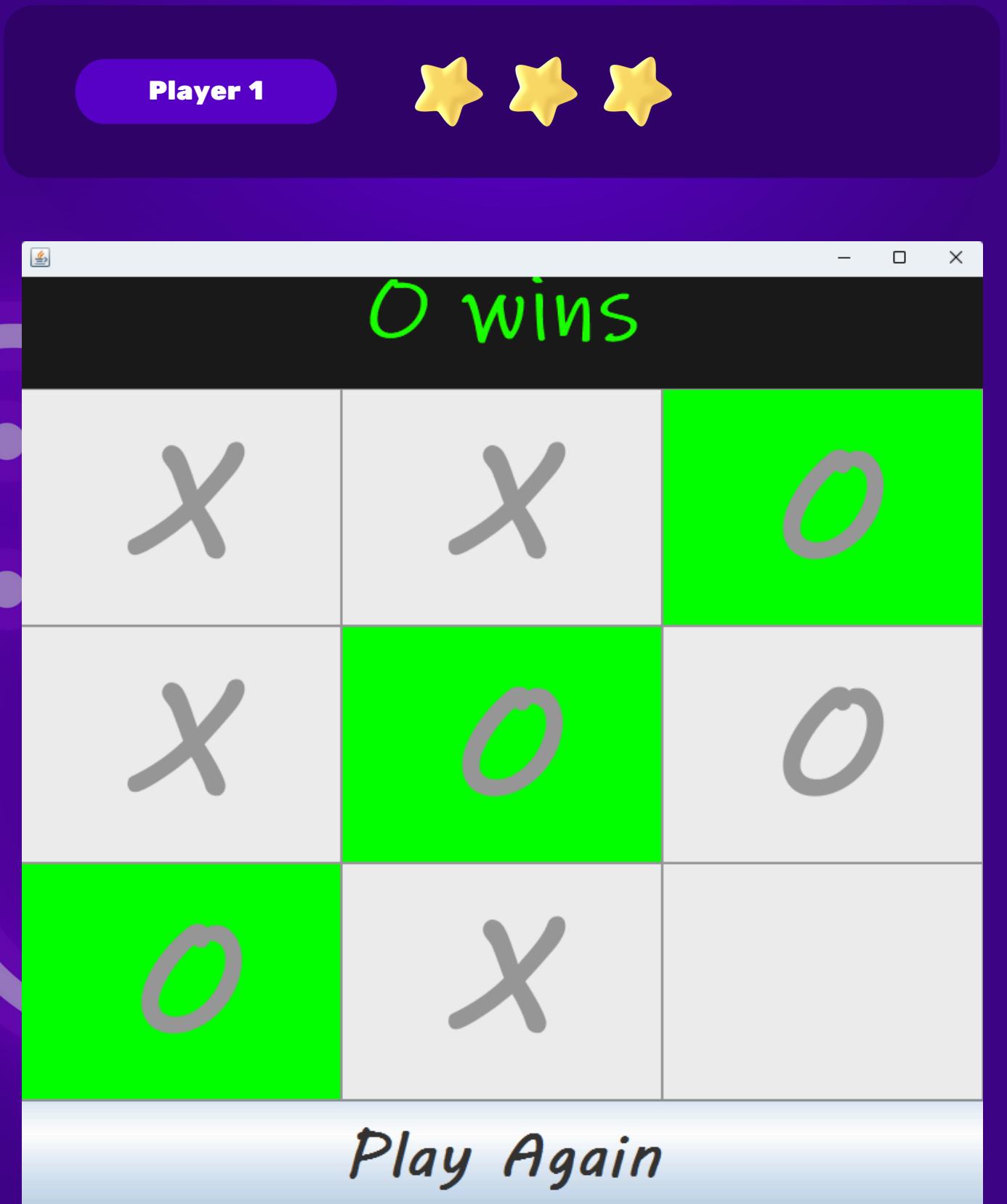
3. Implementation Details

6. Game Logic:

- Used a boolean variable `player1_turn` to track the current player.
- Implemented the `check` method to verify winning conditions after each move.
- Defined methods `xWins` and `oWins` to handle the win scenarios and disable further moves.
- Added a `resetGame` method to clear the board and enable buttons for a new game.

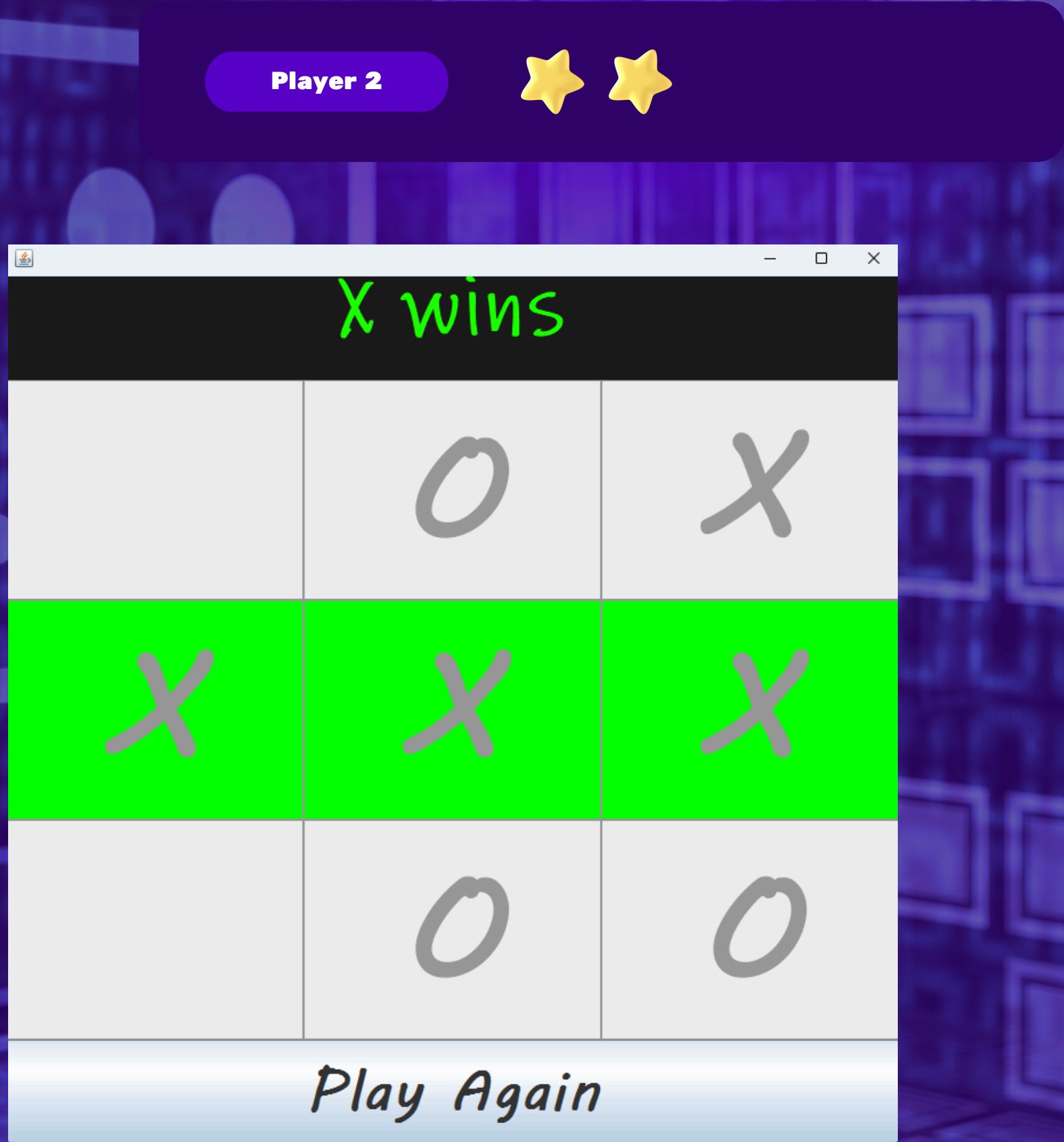
4.RESULT

- Alternate turns between two players.
- Determine the exact winning conditions.
- Show winner.
- The "Play Again" button resets the game board.



5.CONCLUSION

- Simple GUI programming in Java.
- Basic components: JFrame, JPanel, JLabel, JButton.
- Event handling and game logic.
- Enhanced user experience with "Play Again" button.





**THANK YOU
FOR
LISTENING!**

