



# TICTACTOE

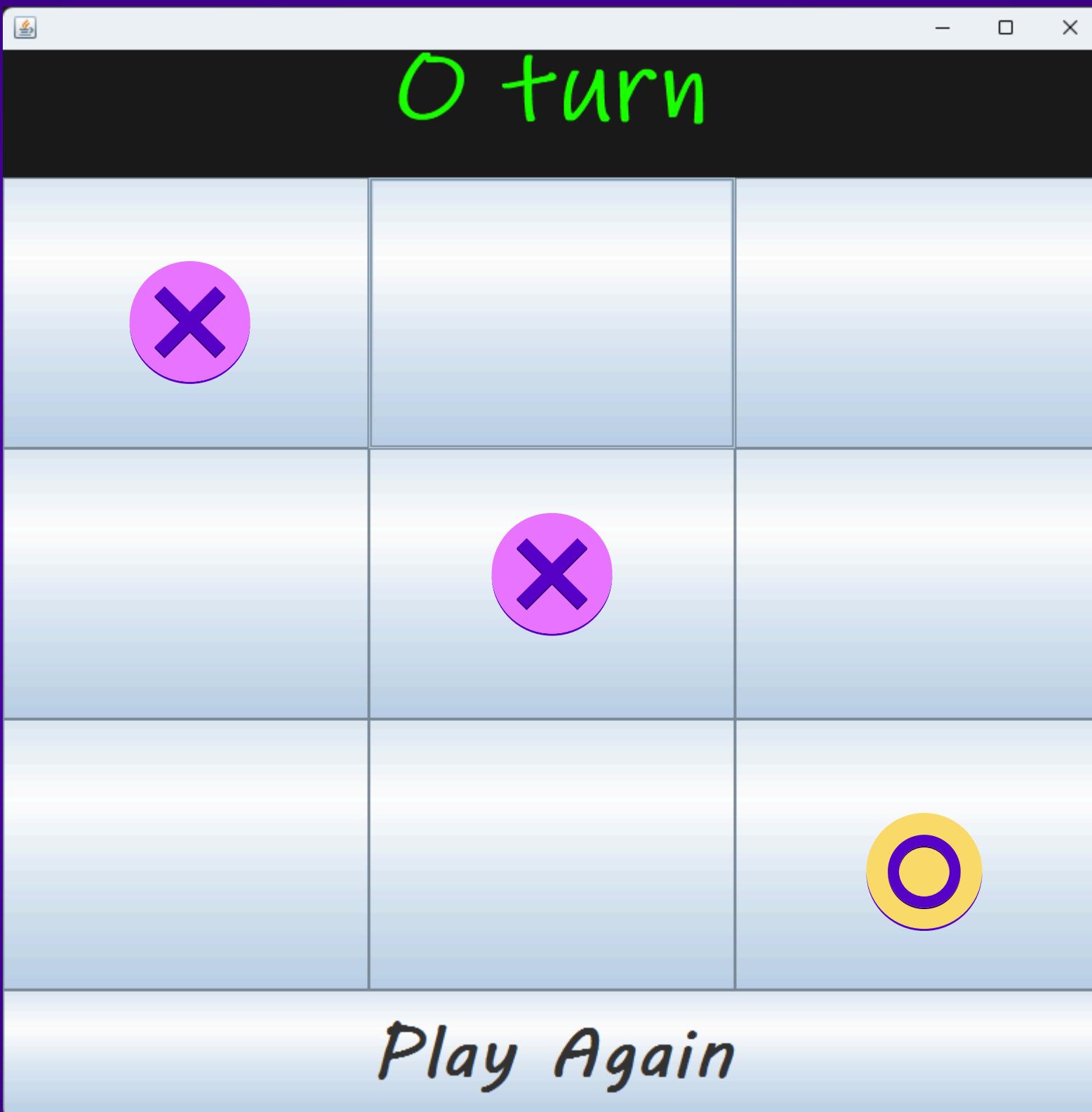
LET'S PLAY



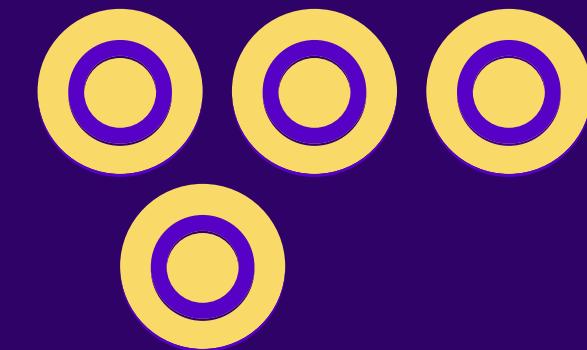
**DRAG THE XS AND OS TO THE SPACE OF YOUR CHOICE.**



Vũ Thành Đạt  
ITITU21041



Nguyễn Thiện Tín  
EEEEIU21060



**LET'S PLAY**

# TICTACTOE

The project involves creating a two-player Tic-Tac-Toe game where players take turns to mark X or O on a 3x3 grid. The game checks for winning conditions after each move and announces the winner. Additionally, a "Play Again" button is provided to reset the game board for a new round.

# **Develop a simple Tic-Tac-Toe game using Java with a graphical user interface (GUI).**

## **Overview**

- 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.

# 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.

# 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).

# 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.

# 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.

# Implementation Details:

## 4. Bottom Panel:

- Created a JPanel named bottom\_panel with a BorderLayout.
- Added a JButton named playAgainButton for resetting the game.

# 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.



*Play Again*

# 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.

# SCORE BOARD

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



# CONCLUSION

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



# TICTACTOE CHAMPION



**THANK YOU  
FOR PLAYING!**

