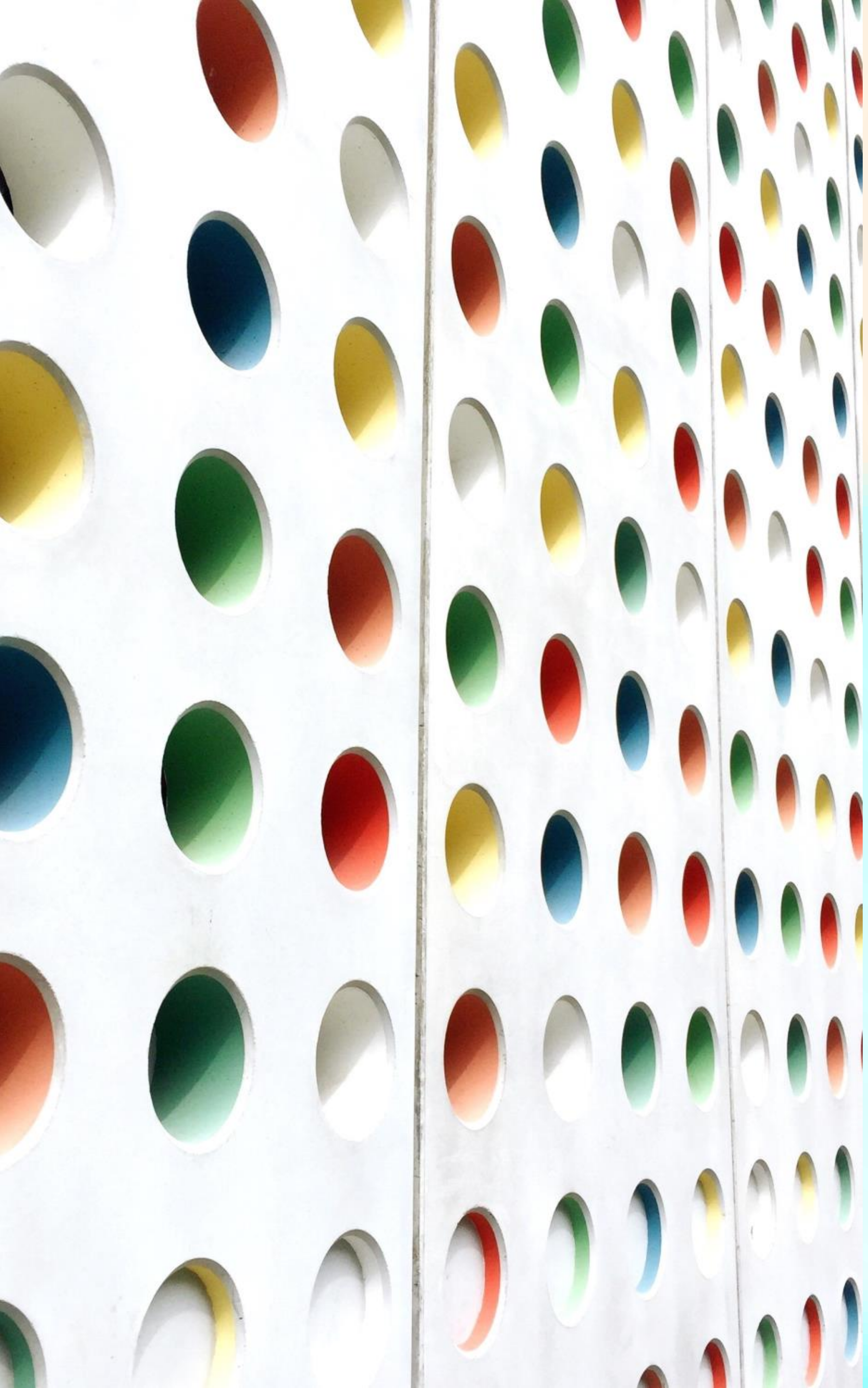


TICTACTOE

Team Members:
Wahid Mubarrat Bin Azhar
ID:220042136

Naybur Rahman Sinha
ID:220042128

- COURSE CODE DETAILS :

SWE 4202

APPLICATION OF OOC

OVERVIEW OF THE PROJECT

- **Create a console-based Tic Tac Toe game initially, and then expand it to have a graphical user interface (GUI) for better user interaction.**

- **Implement different game modes: Player vs Player (PvP), Player vs Computer(PvC).**

- **Using Algorithm for higher difficulty.**

- **Utilize Object-Oriented Programming concepts to design and structure the codebase, ensuring scalability and maintainability.**

DETAILS

- Can Play Player vs Player
- Can play against Computer
- Minimax algorithm used for ensuring maximum difficulty

INITIAL DESIGN

- Classes

- TicTacToeGame: Manages the game logic and flow
- PlayerVsPlayer : manages the game flow for PvP mode
- PlayerVsComputer: manages the game flow for PvC mode
- MenuPanel : Designs the main menu using JavaSwing

OOC CONCEPTS

- Inheritance : PlayerVsPlayer & PlayerVsComputer classes inherit from TicTacToeGame class
- Encapsulation: Data hiding within classes, methods for accessing/modifying data.
- Polymorphism: Different behavior of PvP and PvC through overriding methods
- Interface : TicTacToeGame class implements ActionListener

GITHUB REPOSITORY LINK

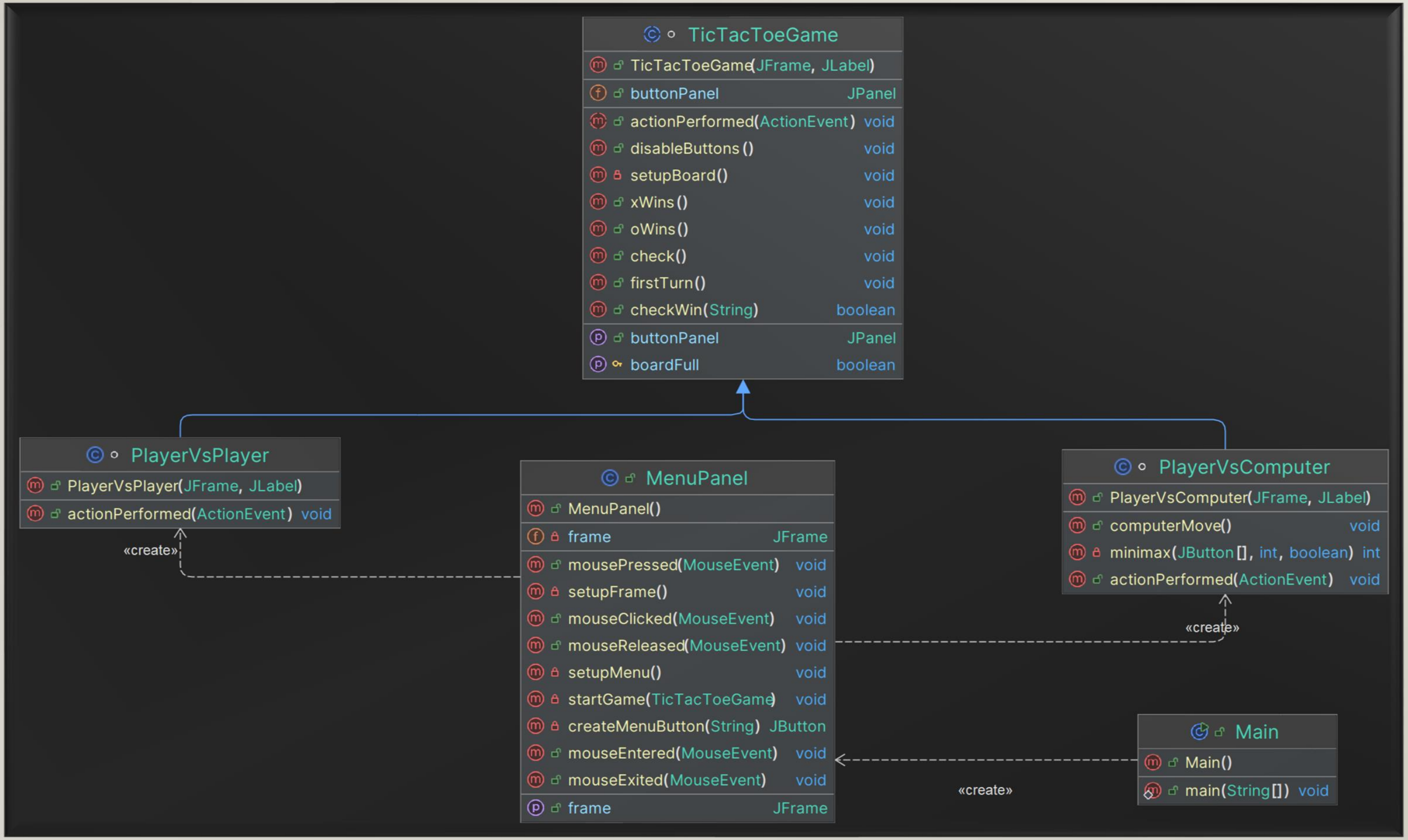
- https://github.com/SinhaWiz/TiCTaCToEproject

```
public void computerMove() {
      int bestScore = Integer.MIN_VALUE;
      int move = -1;
      for (int i = 0; i < 9; i++) {
         if (buttons[i].getText().equals("")) {
            buttons[i].setText("O");
            int score = minimax(buttons, 0, false);
            buttons[i].setText("");
            if (score > bestScore) {
               bestScore = score;
               move = i;
            }
         }
      }
      buttons[move].setForeground(new Color(87, 87, 189));
      buttons[move].setText("O");
      player1Turn = true;
      textfield.setText("X turn");
      check();
   }
```

```
public void xWins() {
   textfield.setText("X wins");
   disableButtons();
}
public void oWins() {
   textfield.setText("O wins");
   disableButtons();
}
public void disableButtons() {
   for (JButton button : buttons) {
      button.setEnabled(false);
   }
}
```

```
public void check() {
   if (checkWin("X")) {
      xWins();
   } else if (checkWin("O")) {
      oWins();
   } else if (isBoardFull()) {
      textfield.setText("DRAW -_- ");
   }
}
protected boolean isBoardFull() {
   for (JButton button : buttons) {
      if (button.getText().equals("")) {
         return false;
      }
   }
   return true;
}
```

```
 private int minimax(JButton[] board, int depth, boolean
isMaximizing) {
      if (checkWin("O")) {
         return 1;
      }
      if (checkWin("X")) {
         return -1;
      }
      if (isBoardFull()) {
         return 0;
      }
      if (isMaximizing) {
         int bestScore = Integer.MIN_VALUE;
         for (int i = 0; i < 9; i++) {
            if (board[i].getText().equals("")) {
               board[i].setText("O");
               int score = minimax(board, depth + 1, false);
               board[i].setText("");
               bestScore = Math.max(score, bestScore);
            }
         }
         return bestScore;
      } else {
         int bestScore = Integer.MAX_VALUE;
         for (int i = 0; i < 9; i++) {
            if (board[i].getText().equals("")) {
               board[i].setText("X");
               int score = minimax(board, depth + 1, true);
               board[i].setText("");
               bestScore = Math.min(score, bestScore);
            }
         }
         return bestScore;
      }
   }
}
```


TicTacToeGame
TicTacToeGame(JFrame, JLabel)
buttonPanel JPanel
actionPerformed(ActionEvent) void
disableButtons() void
setupBoard() void
xWins() void
oWins() void
check() void
firstTurn() void
checkWin(String) boolean
buttonPanel JPanel
boardFull boolean
PlayerVsPlayer
PlayerVsPlayer(JFrame, JLabel)
actionPerformed(ActionEvent) void
«create»
MenuPanel
MenuPanel()
frame JFrame
mousePressed(MouseEvent) void
setupFrame() void
mouseClicked(MouseEvent) void
mouseReleased(MouseEvent) void
setupMenu() void
startGame(TicTacToeGame) void
createMenuButton(String) JButton
mouseEntered(MouseEvent) void
mouseExited(MouseEvent) void
frame JFrame
PlayerVsComputer
PlayerVsComputer(JFrame, JLabel)
computerMove() void
minimax(JButton[], int, boolean) int
actionPerformed(ActionEvent) void
«create»
Main
Main()
main(String[]) void
«create»

thank you!