OOPs - 4

Introduction

In this lecture we will try to create game logics by playing around different classes. We will create a Tic-Tac-Toe game and we will discuss Othello game. It will be a java console application. You can refer to course videos to understand the game rules.

Tic-Tac-Toe

In this game, two players will be played and you have one print board on the screen where from 1 to 9 numbers will be displayed or you can say it box number. Now, you have to choose X or O for the specific box number. For example, if you have to select any number then for X or O will be shown on the print board, and turn for next will be there. The task is to create a Java program to implement a 3×3 Tic-Tac-Toe game for two players.

How to Play the Game :

- Both the players choose either **X** or **O** to mark their cells.
- There will be a 3×3 grid with numbers assigned to each of the 9 cells.
- The player who chose **X** begins to play first.
- He enters the cell number where he wishes to place **X**.
- Now, both **O** and **X** play alternatively until any one of the two wins.
- **Winning criteria:** Whenever any of the two players has fully filled one row/ column/ diagonal with his symbol (X/ O), he wins and the game ends.

- If neither of the two players wins, the game is said to have ended in a **draw**.

Let us now code this game

Board class

```
public class Board {
	private char board[][];
	private int boardSize = 3;
	private char p1Symbol, p2Symbol;
	private int count;
	public final static int PLAYER_1_WINS = 1;
	public final static int PLAYER_2_WINS = 2;
	public final static int DRAW = 3;
	public final static int INCOMPLETE = 4;
	public final static int INVALID = 5;



	public Board(char p1Symbol, char p2Symbol){
		board = new char[boardSize][boardSize];
		for(int i =0; i < boardSize; i++){
			for(int j =0; j < boardSize; j++){
				board[i][j] = ' ';
			}
		}
		this.p1Symbol = p1Symbol;
		this.p2Symbol = p2Symbol;
	}
	public int move(char symbol, int x, int y) {

		if(x < 0 || x >= boardSize || y < 0 || y >= boardSize ||
                                                board[x][y] != ' '){
			return INVALID;
		}

		board[x][y] = symbol;
		count++;
		// Check Row
		if(board[x][0] == board[x][1] && board[x][0] == board[x][2]){
```

```
                return symbol == p1Symbol ? PLAYER_1_WINS :PLAYER_2_WINS;
            }
            // Check Col
            if(board[0][y] == board[1][y] && board[0][y] == board[2][y]){
                return symbol == p1Symbol ? PLAYER_1_WINS :PLAYER_2_WINS;
            }
            // First Diagonal
            if(board[0][0] != ' ' && board[0][0] == board[1][1] &&
                                            board[0][0] == board[2][2]){
                return symbol == p1Symbol ? PLAYER_1_WINS :PLAYER_2_WINS;
            }
            // Second Diagonal
            if(board[0][2] != ' ' && board[0][2] == board[1][1] &&
                                            board[0][2] == board[2][0]){
                return symbol == p1Symbol ? PLAYER_1_WINS :PLAYER_2_WINS;
            }
            if(count == boardSize * boardSize){
                return DRAW;
            }
            return  INCOMPLETE;

        }
        public void print() {
            System.out.println("---------------");
            for(int i =0; i < boardSize; i++){
                for(int j =0; j < boardSize; j++){
                    System.out.print("| " + board[i][j] + " |");
                }
                System.out.println();
            }
            System.out.println();
            System.out.println("---------------");
        }
}
```

Player class

```
public class Player {

	private String name;
	private char symbol;

	public Player(String name, char symbol){
		setName(name);
		setSymbol(symbol);
	}

	public void setName(String name) {

		if(!name.isEmpty()) {
			this.name = name;
		}
	}

	public String getName() {
		return this.name;
	}

	public void setSymbol(char symbol) {
		if(symbol != '\0') {
			this.symbol = symbol;
		}
	}

	public char getSymbol() {
		return this.symbol;
	}
}
```

TicTacToe class (Main class)

```
import java.util.Scanner;

public class TicTacToe {
	private Player player1, player2;
	private Board board;

	public static void main(String args[]){
		TicTacToe t = new TicTacToe();
		t.startGame();
	}

	public void startGame(){
		Scanner s = new Scanner(System.in);
		// Players input
		player1 = takePlayerInput(1);
		player2 = takePlayerInput(2);
		while(player1.getSymbol() == player2.getSymbol()){
			System.out.println("Symbol Already taken !! Pick another
                                                        symbol !!");
			char symbol = s.next().charAt(0);
			player2.setSymbol(symbol);
		}
		// Create Board
		board = new Board(player1.getSymbol(), player2.getSymbol());
		// Conduct the Game
		boolean player1Turn = true;
		int status = Board.INCOMPLETE;
		while(status == Board.INCOMPLETE || status == Board.INVALID){
			if(player1Turn){
				System.out.println("Player 1 - " +
                                        player1.getName() + "'s turn");
				System.out.println("Enter x: ");
				int x = s.nextInt();
				System.out.println("Enter y: ");
				int y = s.nextInt();
				 status =  board.move(player1.getSymbol(), x, y);
				if(status != Board.INVALID){
					player1Turn = false;
					board.print();
				}else{
					System.out.println("Invalid Move!Try Again");
```

```
                        }
                  }else{
                        System.out.println("Player 2 - " +
                                          player2.getName() + "'s turn");
                        System.out.println("Enter x: ");
                        int x = s.nextInt();
                        System.out.println("Enter y: ");
                        int y = s.nextInt();
                        status = board.move(player2.getSymbol(), x, y);
                        if(status != Board.INVALID){
                              player1Turn = true;
                              board.print();
                        }else{
                             System.out.println("Invalid Move! Try Again");
                        }
                  }
            }

            if(status == Board.PLAYER_1_WINS){
                  System.out.println("Player 1 - " + player1.getName() +"
                                                                  wins !!");
            }else if(status == Board.PLAYER_2_WINS){
                  System.out.println("Player 2 - " + player2.getName() +"
                                                                  wins !!");
            }else{
                  System.out.println("Draw !!");
            }
      }

      private Player takePlayerInput(int num){
            Scanner s = new Scanner(System.in);
            System.out.println("Enter Player " + num + " name: ");
            String name = s.nextLine();
            System.out.println("Enter Player " + num + " symbol: ");
            char symbol = s.next().charAt(0);
            Player p = new Player(name, symbol);
            return p;
      }
}
```

Othello

Refer to course videos for better understanding of the game.
Othello is a board game and you are expected to implement the move function for this game.

Approach: We have eight different directions to explore before we make a move (before we make changes to the board) to ensure which all boxes will toggle. We will move in every direction one by one and check for the player's value who just made his turn. We will then toggle the desired boxes, which the current player secured by playing in that position. To explore all eight directions we can create arrays for different directions and looping in the board we can increment or decrement in respective value to move in the particular direction, like we explore ways in backtracking lecture for rat in a maze game.

Now this code can be written on your own. Refer to the solution tab for the solution.