Tic Tac Toe Game Project Description

Project Overview

Create a console-based Tic Tac Toe game where two players can play against each other. The game will follow the standard rules of Tic Tac Toe, where the objective is to get three of your marks (X or O) in a row, column, or diagonal. The game ends when one player wins or the board is full, resulting in a tie.

Objectives

- Develop an understanding of arrays, loops, and conditionals in Java.
- Practice implementing a game loop and handling user input.
- Learn how to check for win conditions and draw the game board in a console application.

Requirements

1. **Game Setup:**
    - The game should use a 3x3 board.
    - Players take turns to place their marks (X or O) on the board.
2. **Gameplay:**
    - Prompt players to enter their move.
    - Validate the move to ensure it is within the bounds of the board and the cell is not already occupied.
    - Alternate turns between the two players.
3. **Winning Condition:**
    - Check for a winning condition after each move (three marks in a row, column, or diagonal).
    - The game ends when one player wins or the board is full, resulting in a tie.

Step-by-Step Guide

Step 1: Define Constants and Variables

Define constants and variables to represent the game board and other necessary information.

Step 1.1: Define Constants

- **Description:** Define constants for the size of the board and the marks used by the players.

Step 1.2: Define Variables

- **Description:** Define variables to represent the game board and the current player.

Step 2: Initialize the Game

Create methods to initialize the game board and set the starting player.

Method 2.1: `initializeBoard`

- **Description:** Initialize the game board with empty spaces.

Method 2.2: `initializeGame`

- **Description:** Initialize the game by setting up the board and starting player.

Step 3: Display the Board

Create methods to display the current state of the game board.

Method 3.1: `displayBoard`

- **Description:** Display the current state of the game board in the console.

Step 4: Player Move

Create methods to handle a player's move, including input validation.

Method 4.1: `makeMove`

- **Description:** Handle a player's move by prompting for input and updating the board.

Step 5: Check for Win or Tie

Create methods to check for a winning condition or a tie after each move.

Method 5.1: `checkWin`

- **Description:** Check if the current player has won the game.

Method 5.2: `checkTie`

- **Description:** Check if the game has ended in a tie.

Step 6: Switch Player

Create a method to switch the current player after each move.

Method 6.1: `switchPlayer`

- **Description:** Switch the current player from X to O or O to X.

Step 7: Main Game Loop

Create the main game loop to handle turns and check for win or tie conditions.

Method 7.1: `playGame`

- **Description:** The main game loop that controls the flow of the game from start to finish.

Main Method

The main method to start the game.

```
public class TicTacToeGame {

        public static void main(String[] args) {

        TicTacToe game = new TicTacToe();

        game.playGame();

        }

}
```