

```

1  // TicTacToe game in Java for two players.
2
3  import java.util.Scanner;
4
5  public class ticTacToe {
6      public static Scanner sc = new Scanner(System.in);
7      static char[][] board = {
8          {'1', '2', '3'},
9          {'4', '5', '6'},
10         {'7', '8', '9'},
11     };
12     static char player1choice, player2choice;
13     static int row, column;
14     static int player1count = 0, player2count = 0;
15
16     public static void main(String[] args) {
17         System.out.println("Welcome to the tictactoe game made by Kevin!");
18         System.out.println("Enter letter to represent player 1: ");
19         player1choice = sc.next().charAt(0);
20         System.out.println("Enter letter to represent player 2: ");
21         player2choice = sc.next().charAt(0);
22         printBoard();
23         // If no player has won, continue
24         while (!checkWinner(player1choice) && !checkWinner(player2choice)) {
25             player1Turn();
26             player2Turn();
27         }
28     }
29
30     // Print the current board
31     public static void printBoard() {
32         for (int i = 0; i < 3; i++) {
33             System.out.print("| ");
34             for (int j = 0; j < 3; j++) {
35                 System.out.print(board[i][j] + " | ");
36             }
37             System.out.println();
38         }
39     }
40
41     // Determine 2D (x,y) coordinates from 1D position
42     public static void determineRowAndColumn(int position) {
43         if (position == 1) {
44             row = 0;
45             column = 0;
46         }
47         if (position == 2) {
48             row = 0;
49             column = 1;
50         }
51         if (position == 3) {
52             row = 0;
53             column = 2;
54         }
55         if (position == 4) {
56             row = 1;
57             column = 0;
58         }
59         if (position == 5) {
60             row = 1;
61             column = 1;
62         }
63         if (position == 6) {
64             row = 1;
65             column = 2;
66         }
67         if (position == 7) {
68             row = 2;
69             column = 0;
70         }
71         if (position == 8) {
72             row = 2;
73             column = 1;
74         }
75         if (position == 9) {
76             row = 2;
77             column = 2;
78         }
79     }
80

```

```

81 // Make the move on the board for player 1
82 public static void player1move(int row, int column) {
83     board[row][column] = player1choice;
84 }
85
86 // Make the move on the board for player 2
87 public static void player2move(int row, int column) {
88     board[row][column] = player2choice;
89 }
90
91 // Check if a player won
92 public static boolean checkWinner(char playerChoice) {
93     // Check rows
94     if (board[0][0] == playerChoice && board[0][1] == playerChoice && board[0][2] == playerChoice)
95         return true;
96     if (board[1][0] == playerChoice && board[1][1] == playerChoice && board[1][2] == playerChoice)
97         return true;
98     if (board[2][0] == playerChoice && board[2][1] == playerChoice && board[2][2] == playerChoice)
99         return true;
100     // Check columns
101     if (board[0][0] == playerChoice && board[1][0] == playerChoice && board[2][0] == playerChoice)
102         return true;
103     if (board[0][1] == playerChoice && board[1][1] == playerChoice && board[2][1] == playerChoice)
104         return true;
105     if (board[0][2] == playerChoice && board[1][2] == playerChoice && board[2][2] == playerChoice)
106         return true;
107     // Check diagonals
108     if (board[0][0] == playerChoice && board[1][1] == playerChoice && board[2][2] == playerChoice)
109         return true;
110     return board[0][2] == playerChoice && board[1][1] == playerChoice && board[2][0] == playerChoice;
111 }
112
113 public static void checkTie() {
114     int count = player1count + player2count; // Number of turns taken by players
115     if (count == 9) {
116         System.out.println("It's a tie");
117         System.exit(0);
118     }
119 }
120
121 public static void player1Turn() {
122     int flag1 = 0;
123     while (flag1 == 0) {
124         System.out.println("Player " + player1choice + ", enter position {1..9}");
125         int player1position = sc.nextInt();
126         determineRowAndColumn(player1position);
127         if (board[row][column] != player2choice && board[row][column] != player1choice) {
128             player1count += 1;
129             checkTie();
130             player1move(row, column);
131             printBoard();
132             boolean b = checkWinner(player1choice);
133             if (b) {
134                 System.out.println("Player 1(" + player1choice + ") won!");
135                 System.exit(0);
136             }
137             flag1 = 1;
138         } else {
139             System.out.println("Already occupied! Try again");
140         }
141     }
142 }
143
144 public static void player2Turn() {
145     int flag2 = 0;
146     while (flag2 == 0) {
147         System.out.println("Player " + player2choice + ", enter position {1..9}");
148         int player2position = sc.nextInt();
149         determineRowAndColumn(player2position);
150         if (board[row][column] != player1choice && board[row][column] != player2choice) {
151             player2count += 1;
152             checkTie();
153             player2move(row, column);
154             printBoard();
155             boolean b = checkWinner(player2choice);
156             if (b) {
157                 System.out.println("Player 2 (" + player2choice + ") won!");
158                 System.exit(0);
159             }
160             flag2 = 1;
161         } else {
162             System.out.println("Already occupied! Try again");
163         }
164     }
165 }

```

```
164     }
165 }
166 }
167 /**
168  * Variable      Data      Table
169  * board         int[][]   Store board
170  * player1choice char      Store 1st users avatar
171  * player2choice char      Store 2nd users avatar
172  * player1position int      Where 1st user choose to move
173  * player2position int      Where 2nd user choose to move
174  * row, column   int        2d coordinates for move
175  * position      int        1d coordinates for move
176  * i, j          int        Used in calculation
177 */
```