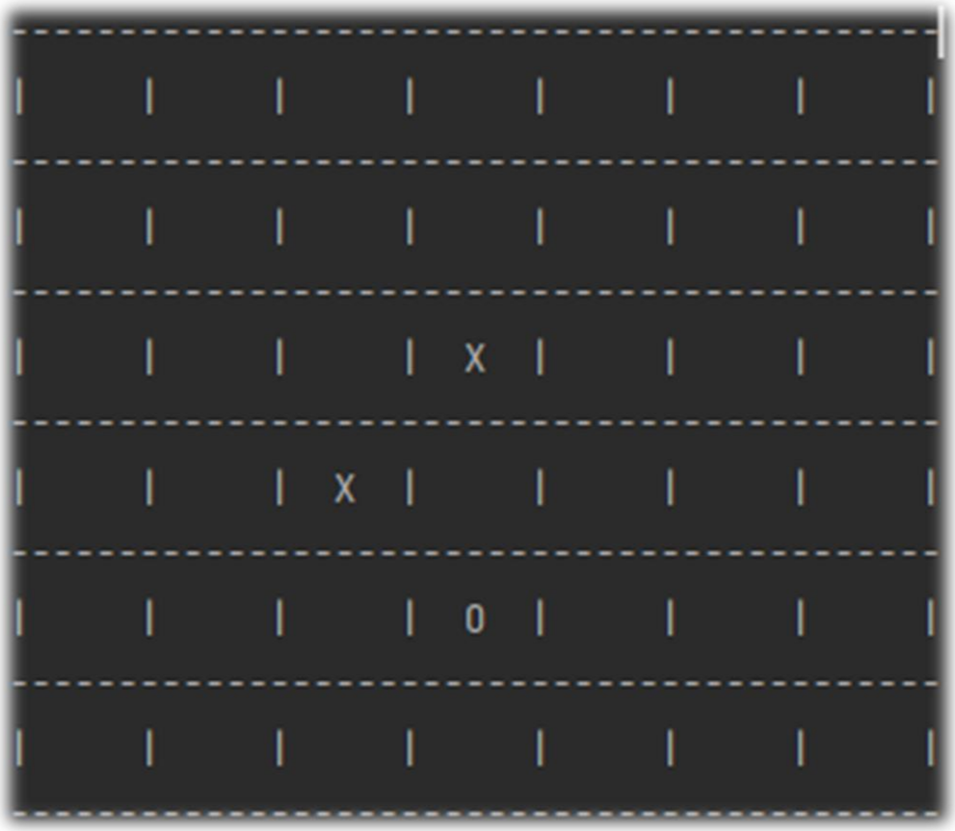
Tic Tac Toe Game



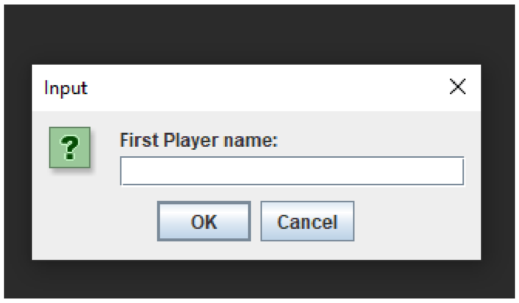
```
-------------------------------------------
|     |     |     |     |     |     |     |
-------------------------------------------
|     |     |     |     |     |     |     |
-------------------------------------------
|     |     |     |  X  |     |     |     |
-------------------------------------------
|     |     |  X  |     |     |     |     |
-------------------------------------------
|     |     |     |  O  |     |     |     |
-------------------------------------------
|     |     |     |     |     |     |     |
-------------------------------------------
```

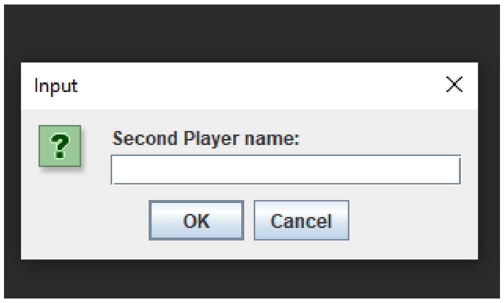
By: Mohamed Abdou Mabrouk

The Game

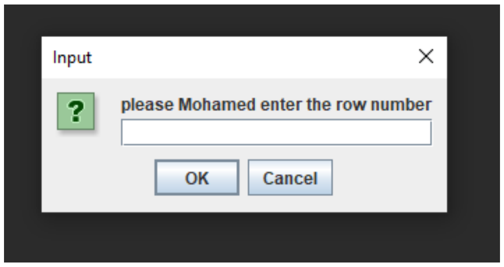
How to play:

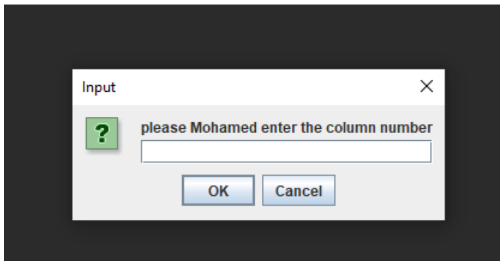
When we start the game, the console will ask for the first and second players names:



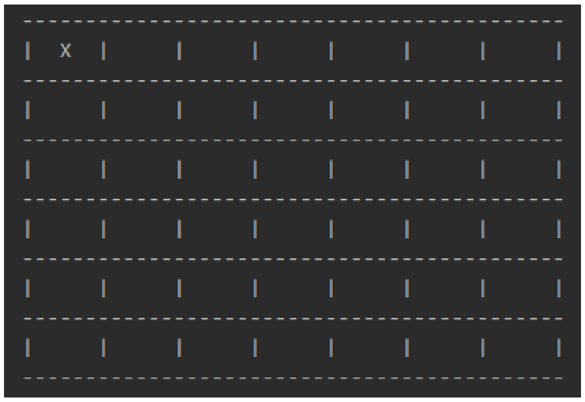


Then it will ask from the first player to choose the place to insert the letter:





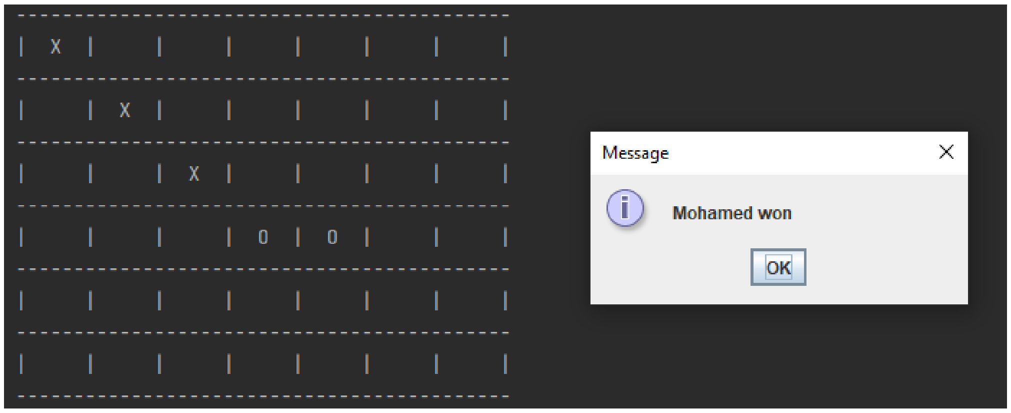
The letter will be inserted in the board:



```
-----------------------------------------------------
|  X  |     |     |     |     |     |     |
-----------------------------------------------------
|     |     |     |     |     |     |     |
-----------------------------------------------------
|     |     |     |     |     |     |     |
-----------------------------------------------------
|     |     |     |     |     |     |     |
-----------------------------------------------------
|     |     |     |     |     |     |     |
-----------------------------------------------------
|     |     |     |     |     |     |     |
-----------------------------------------------------
```

Rules:

- Who get three letters in row first win’s:

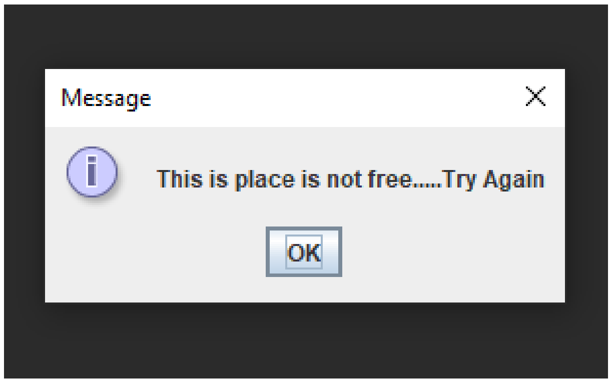


```
-------------------------------------------------
|  X  |     |     |     |     |     |     |
-------------------------------------------------
|     |  X  |     |     |     |     |     |
-------------------------------------------------
|     |     |  X  |     |     |     |     |
-------------------------------------------------
|     |     |     |  O  |  O  |     |     |
-------------------------------------------------
|     |     |     |     |     |     |     |
-------------------------------------------------
|     |     |     |     |     |     |     |
-------------------------------------------------
```

Message

Mohamed won

- You cannot play in the same place:



Message

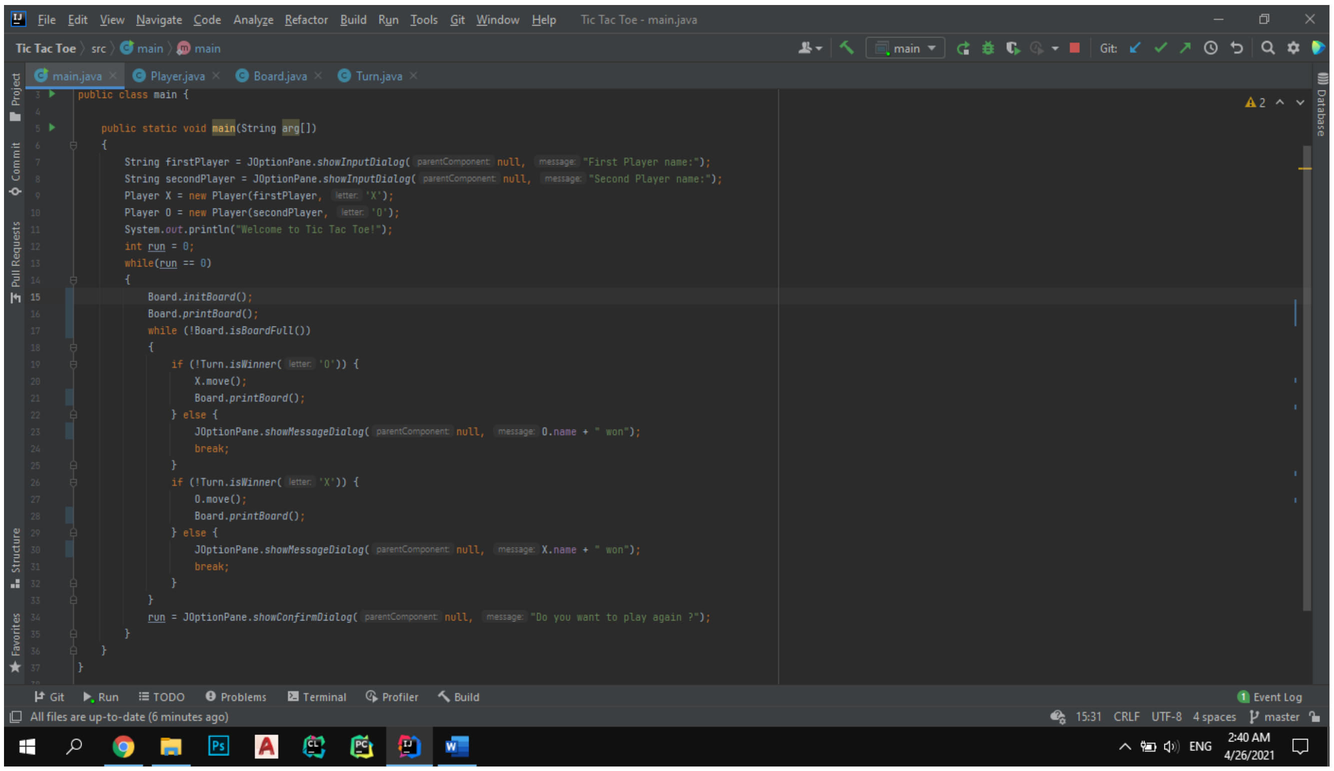
This is place is not free.....Try Again

Syntax

The program has 4 classes: -

- Main
- Player
- Board
- Turn

1- Main:



```java
public class main {

    public static void main(String arg[])
    {
        String firstPlayer = JOptionPane.showInputDialog(null, "First Player name:");
        String secondPlayer = JOptionPane.showInputDialog(null, "Second Player name:");
        Player X = new Player(firstPlayer, 'X');
        Player O = new Player(secondPlayer, 'O');
        System.out.println("Welcome to Tic Tac Toe!");
        int run = 0;
        while(run == 0)
        {
            Board.initBoard();
            Board.printBoard();
            while (!Board.isBoardFull())
            {
                if (!Turn.isWinner('O')) {
                    X.move();
                    Board.printBoard();
                } else {
                    JOptionPane.showMessageDialog(null, O.name + " won");
                    break;
                }
                if (!Turn.isWinner('X')) {
                    O.move();
                    Board.printBoard();
                } else {
                    JOptionPane.showMessageDialog(null, X.name + " won");
                    break;
                }
            }
            run = JOptionPane.showConfirmDialog(null, "Do you want to play again ?");
        }
    }
}
```

In this class we run the main and keep the game running by while loop until the board is full or any of players wins.

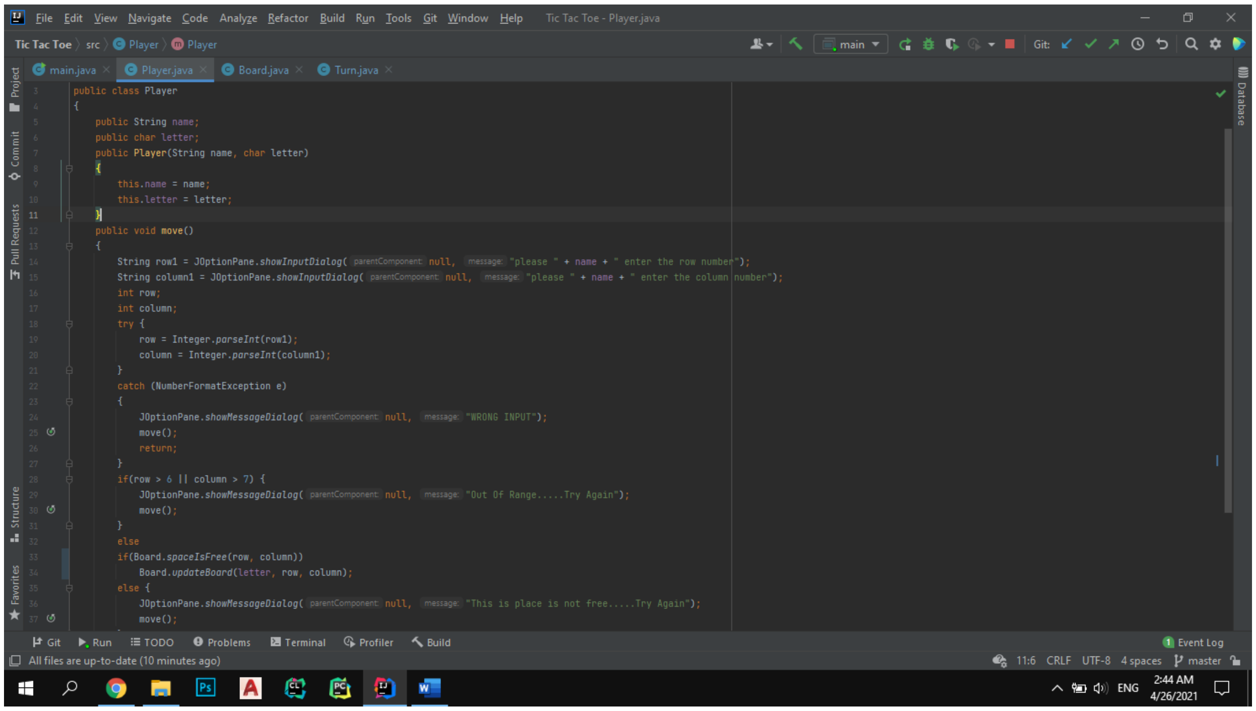
First, the program takes the players name from by the console.

Then create two objects from the player class.

Second, in the outer loop the program initiates the board and print it, in the inner loop condition the program checks if the board is full or not, if it's not full checks if the 'O' player won or not, if he/she did not win, the 'X' player plays after that checks if the 'X' player won, if he/she did not, the program will keep repeat the previous steps in the while loop.

Back to the outer loop the program will ask if the players want to play again.

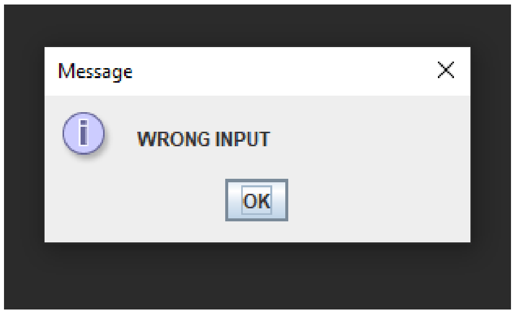
2- Player:



```java
public class Player
{
    public String name;
    public char letter;
    public Player(String name, char letter)
    {
        this.name = name;
        this.letter = letter;
    }
    public void move()
    {
        String row1 = JOptionPane.showInputDialog(null, "please " + name + " enter the row number");
        String column1 = JOptionPane.showInputDialog(null, "please " + name + " enter the column number");
        int row;
        int column;
        try {
            row = Integer.parseInt(row1);
            column = Integer.parseInt(column1);
        }
        catch (NumberFormatException e)
        {
            JOptionPane.showMessageDialog(null, "WRONG INPUT");
            move();
            return;
        }
        if(row > 6 || column > 7) {
            JOptionPane.showMessageDialog(null, "Out Of Range.....Try Again");
            move();
        }
        else
        if(Board.spaceIsFree(row, column))
            Board.updateBoard(letter, row, column);
        else {
            JOptionPane.showMessageDialog(null, "This is place is not free.....Try Again");
            move();
```

In this class constructor takes two parameters (name, letter) and set them.

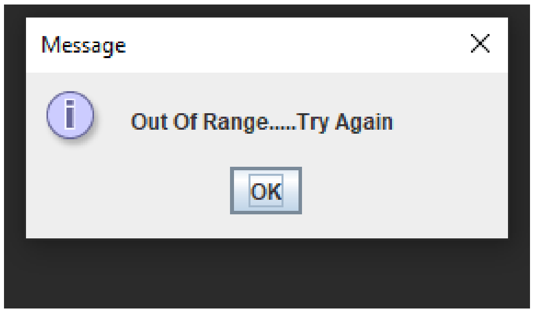
Move method takes rows and column and set them in integers, then by try and catch the program check that if the input is not out of the range, if it not set them in them in the board if the place is free.



Message

WRONG INPUT

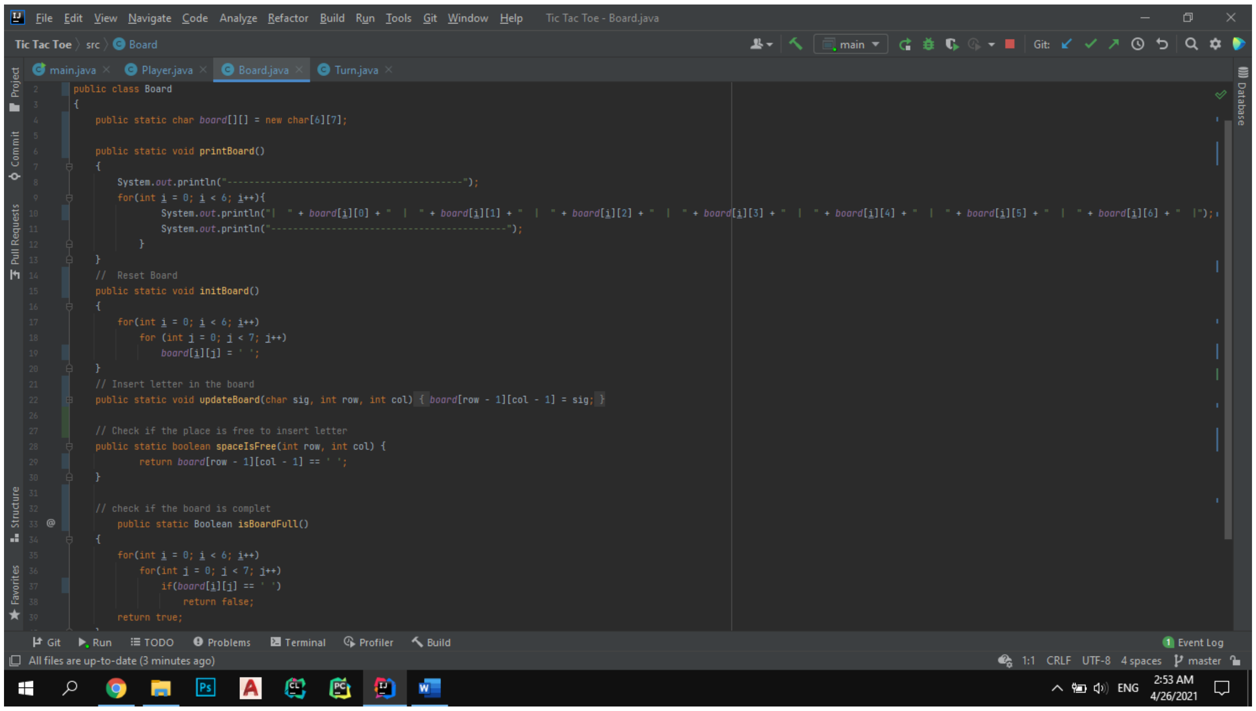
OR



Message

Out Of Range.....Try Again

3- Board:

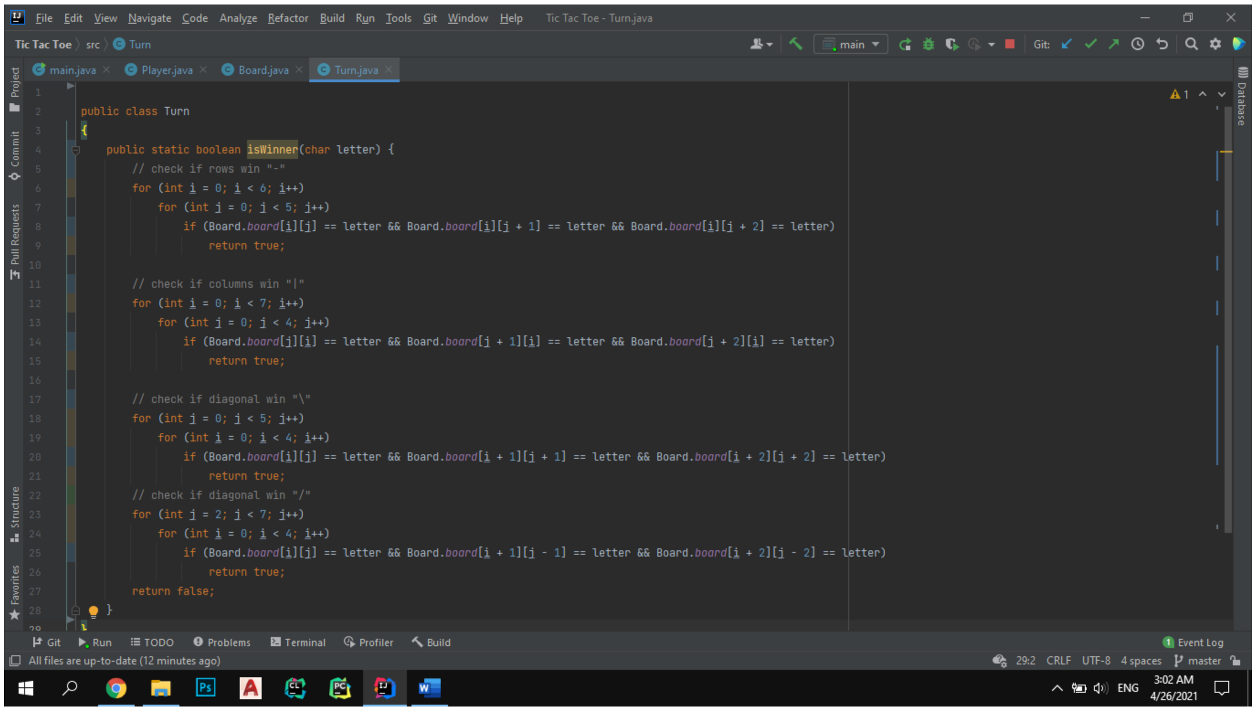

File Edit View Navigate Code Analyze Refactor Build Run Tools Git Window Help
Tic Tac Toe - Board.java
Tic Tac Toe 〉 src 〉 Board
main.java
Player.java
Board.java
Turn.java
Git Run TODO Problems Terminal Profiler Build
All files are up-to-date (3 minutes ago)

```java
public class Board
{
    public static char board[][] = new char[6][7];

    public static void printBoard()
    {
        System.out.println("----------------------------------------");
        for(int i = 0; i < 6; i++){
              System.out.println("|  " + board[i][0] + "  |  " + board[i][1] + "  |  " + board[i][2] + "  |  " + board[i][3] + "  |  " + board[i][4] + "  |  " + board[i][5] + "  |  " + board[i][6] + "  |");
              System.out.println("----------------------------------------");
          }
    }
//  Reset Board
    public static void initBoard()
    {
        for(int i = 0; i < 6; i++)
            for (int j = 0; j < 7; j++)
                board[i][j] = ' ';
    }
    // Insert letter in the board
    public static void updateBoard(char sig, int row, int col) { board[row - 1][col - 1] = sig; }

    // Check if the place is free to insert letter
    public static boolean spaceIsFree(int row, int col) {
            return board[row - 1][col - 1] == ' ';
    }

    // check if the board is complet
        public static Boolean isBoardFull()
    {
        for(int i = 0; i < 6; i++)
            for(int j = 0; j < 7; j++)
                if(board[i][j] == ' ')
                    return false;
        return true;
```

This class has the following methods:

1- Print the board.
2- Initiate the board and reset it by set all its elements by spaces.
3- Update the board by setting the letter in the element.
4- Check if space is free.
5- Check if the board is full.

4- Turn:

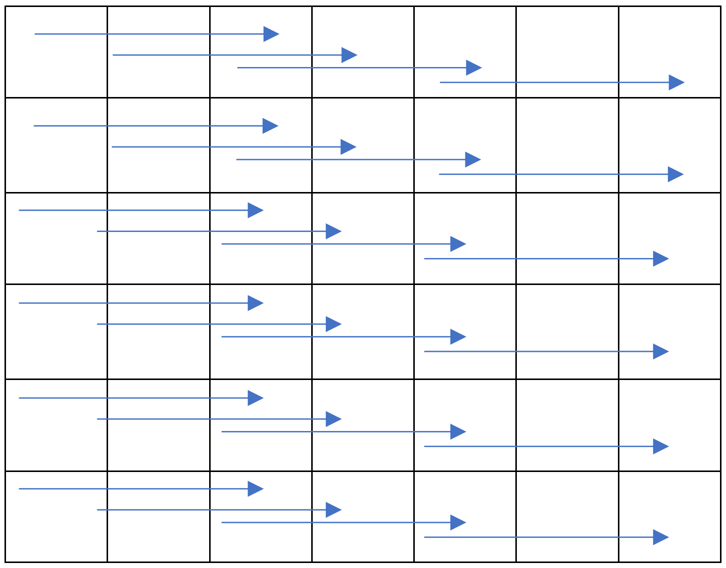


```java
public class Turn
{
    public static boolean isWinner(char letter) {
        // check if rows win "-"
        for (int i = 0; i < 6; i++)
            for (int j = 0; j < 5; j++)
                if (Board.board[i][j] == letter && Board.board[i][j + 1] == letter && Board.board[i][j + 2] == letter)
                    return true;

        // check if columns win "|"
        for (int i = 0; i < 7; i++)
            for (int j = 0; j < 4; j++)
                if (Board.board[j][i] == letter && Board.board[j + 1][i] == letter && Board.board[j + 2][i] == letter)
                    return true;

        // check if diagonal win "\"
        for (int j = 0; j < 5; j++)
            for (int i = 0; i < 4; i++)
                if (Board.board[i][j] == letter && Board.board[i + 1][j + 1] == letter && Board.board[i + 2][j + 2] == letter)
                    return true;
        // check if diagonal win "/"
        for (int j = 2; j < 7; j++)
            for (int i = 0; i < 4; i++)
                if (Board.board[i][j] == letter && Board.board[i + 1][j - 1] == letter && Board.board[i + 2][j - 2] == letter)
                    return true;
        return false;
    }
}
```

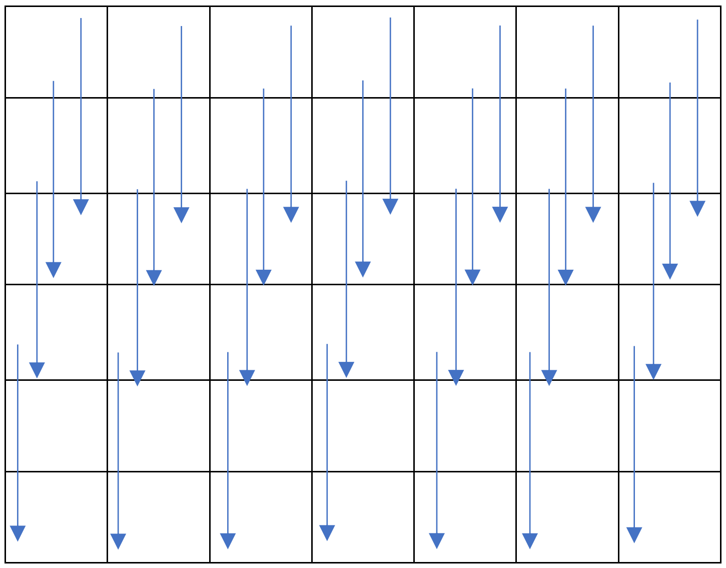
All files are up-to-date (12 minutes ago)

This class checks if the player had won or not by:

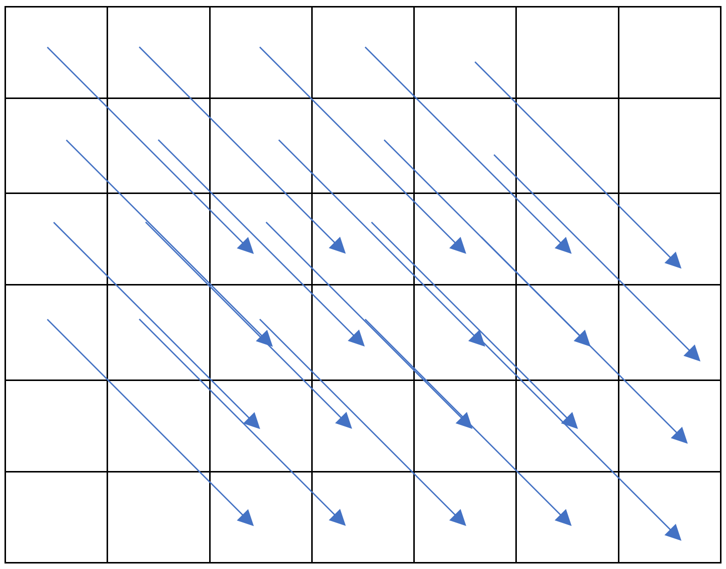
1- Checking Rows:



2- Checking Columns:



3- Checking First Diagonal:



4- Checking Second Diagonal: