Tic Tac Toe Manual

By:

Adham Mohamed - 6744

Abdelrahman El Attar - 6841

Intro

Welcome to our tic tac toe manual. This manual illustrates how our game works. The game is coded and designed by Java programming language.

The game consists of 4 classes:

- Player
- Menus
- Game
- Main

• Player Class

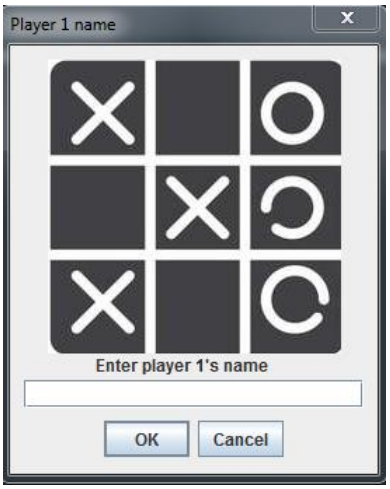
The Player class contains player name, player symbol and number of games won. When a new object is created. The method "public void setName(String name)" is initialized to take the username if the user hasn't entered anything, the name is set to "Player 1" and/or "Player 2" by default depending on which object name is

to be set. The method “public String getName()” is a getter used to obtain player name. The method “public void setPlayerSymbol(char symbol)” is a setter used to set the symbol ‘X’ or ‘O’ for player object and the method “public char getplayersymbol()” is the getter that returns the object symbol when needed.

“Boolean insertsymbol(Game game, int I, int j, char playersymbol)” this method is used to check whether the place where the user is inserting the Symbol whether ‘X’ or ‘O’ is empty. if empty, which is referred by us as ‘ ‘ the symbol is inserted. Else the user is asked again to enter new coordinates to fill.

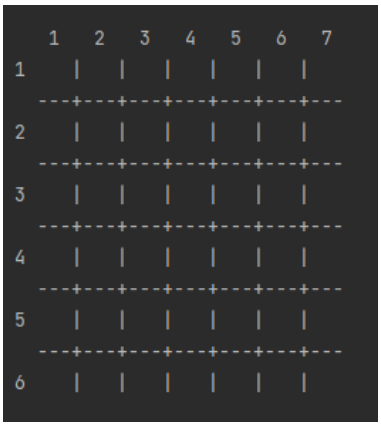
- # Menus Class

This class implements the main menu of the game. First it imports the icon of our tic tac toe game then “public static void welcomeMessage()” creates object JLabel logo and texts create a JPanel and set all objects in its position. It actually shows the welcoming message dialogue “public static void nameInputDialog(Player player1, Player player2)” this shows the message dialogue where users enter their names, and this dialogue appears twice once for player 1 and another for player 2.



- # Game Class

This class implements all the moves and turns of each player declares the winner or if it's a draw and all also draw the template of the game board. First of all, the class constructor, "public void initGameBoard()". This method fills the declared array "gameBoard" of type char with spaces (Gaps). So the initial Display of game board be like this when printed by method "public void printGameBoard()":



We customized our template by colour using some ANSI escape codes with escape sequence \033[.

Example for some used colours:

Red → \033[31m

Green → \033[32m

Yellow → \033[33m

And there are others.

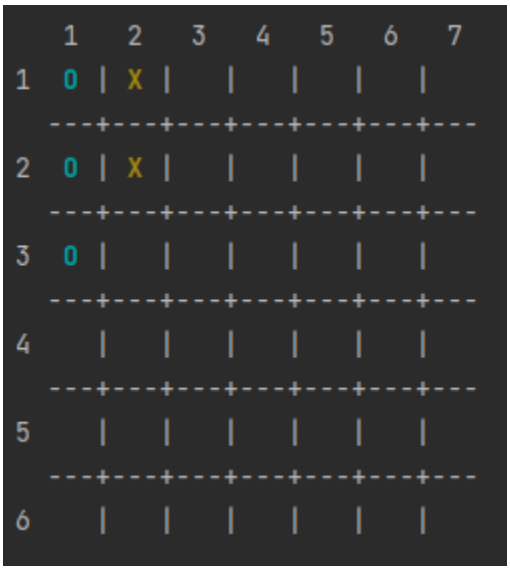
Then there a method "public static void randomisePlayerSymbols()" this method changes the symbol of each player every turn by importing java.util.Random and creating the object rand of class random which uses method nextInt(int) which choose a number between 0 and the number between braces for example:- 11.

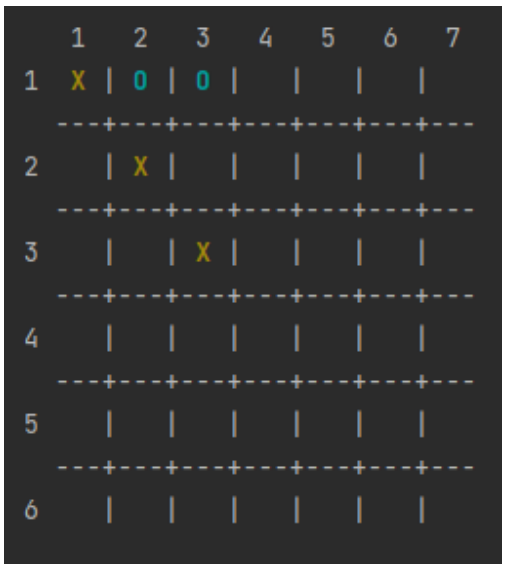
So it chooses 0,1,2,3,4,…,10 and then sees if the number (%2 != 0) if true player1 will be 'X' and player2 will be 'O' if not then vice versa. The main method of our

game is “public int checkForWinner(Player player1, Player player2)” this method checks the win every round either one of our 4 conditions is true (vertical alignment, Horizontal alignment, left diagonal and right diagonal alignment)

This method by the game board every turn checking if any of our winning conditions is true for example if we are talking about vertical alignment, it checks whether symbol in [i][j] == symbol in [i+1][j] == symbol in [i+2][j] and so on. If player1 is the winner, it will return 1 else if player2 it will return 2 else it will return 0 which no one has win this turn and not Draw.

This is the 4 methods of winning:





Else it will be draw if the board it totally filled which we check by limiting number of turns to 42, which is equal to the number of squares in the game board.

The last method in this class is "public void newTurn()" this method prints newline and print the board again after the symbol has been . In other words, it is used to refresh the game state.

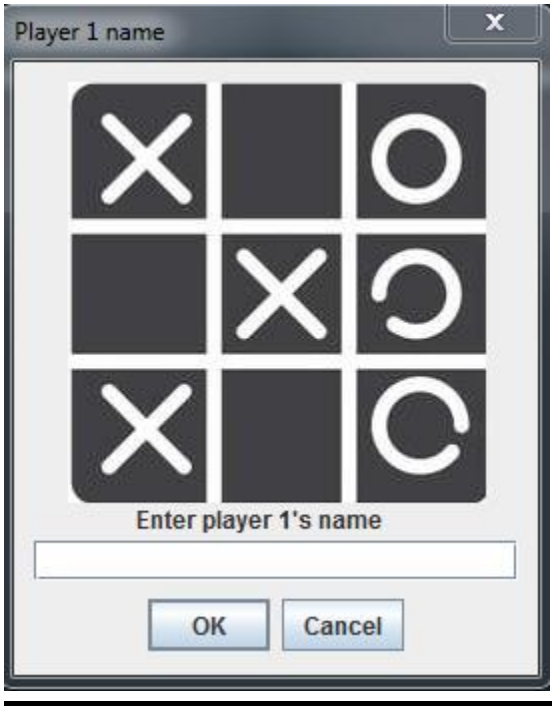
- # Main Class

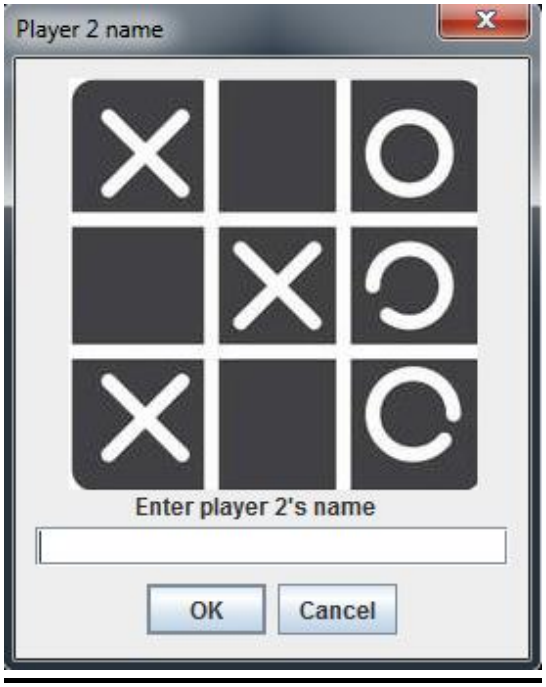
This class is structure of our code which implements all other classes such as Player, Game and Menus. It starts by creating 2 objects from Player class (player1 and player2) then using menus class for the message dialogue and name input for every player then it create an object from class then use the methods symbol for giving every player a symbol but randomly and then using "public static void gameSession(Game game, Player player1, Player player2)" method initialize the Game and take inputs every time by user to insert symbol and check validity and also end the game if either the one of the checkForWinner() method is equal to 1 or 2 or the number of turns ended which is equal to 42 and declaring DRAW it also counts number of wins for each player.

***Sample run in the following pages.**

Sample Run


6x7 Tic Tac Toe!
Welcome to 6x7 Tic Tac Toe !
Press OK to continue
OK


Player 1 name
Enter player 1's name
OK
Cancel

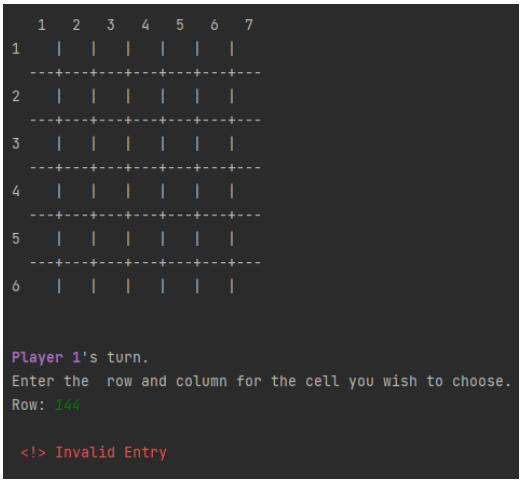

Player 2 name
Enter player 2's name
OK
Cancel

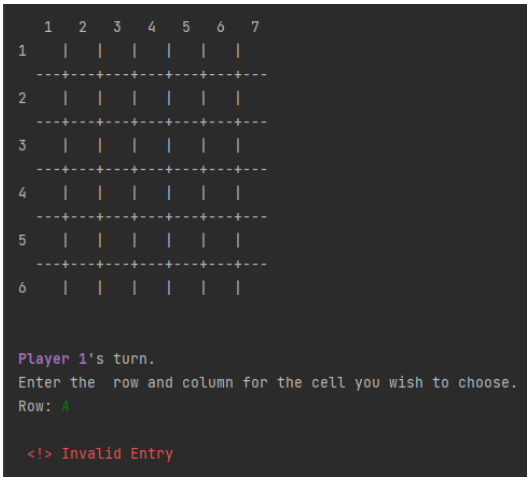

1 2 3 4 5 6 7
1
2
3
4
5
6

```
   1   2   3   4   5   6   7
1    |   |   |   |   |   |
  ---+---+---+---+---+---+---
2    |   |   |   |   |   |
  ---+---+---+---+---+---+---
3    |   |   |   |   |   |
  ---+---+---+---+---+---+---
4    |   |   |   |   |   |
  ---+---+---+---+---+---+---
5    |   |   |   |   |   |
  ---+---+---+---+---+---+---
6    |   |   |   |   |   |


Player 1's turn.
Enter the  row and column for the cell you wish to choose.
Row: 1
Column: 1
```

```
   1   2   3   4   5   6   7
1  X |   |   |   |   |   |
  ---+---+---+---+---+---+---
2    |   |   |   |   |   |
  ---+---+---+---+---+---+---
3    |   |   |   |   |   |
  ---+---+---+---+---+---+---
4    |   |   |   |   |   |
  ---+---+---+---+---+---+---
5    |   |   |   |   |   |
  ---+---+---+---+---+---+---
6    |   |   |   |   |   |


Player 2's turn.
Enter the  row and column for the cell you wish to choose.
Row: 2
Column: 1
```

Invalid Inputs





Team members

Adham Mohamed - 6744

Abdelrahman El Attar - 6841