Programming

2020 / 2021

Java programming

PROJECT

Tic-Tac-Toe

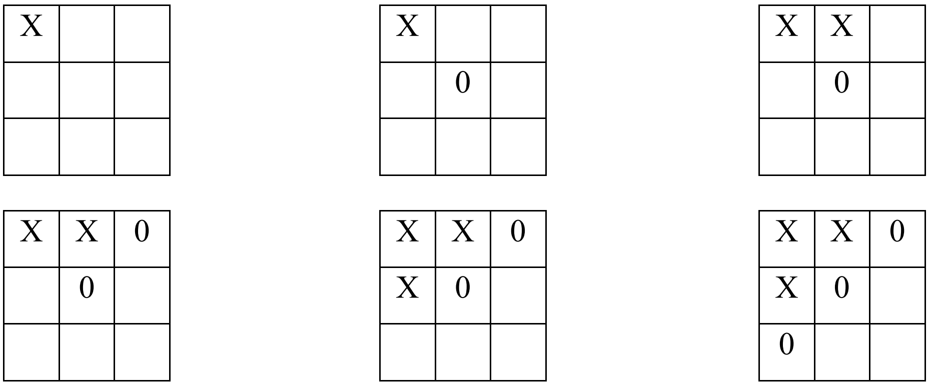
Table of contents

Description

The goal of the project is to implement an application for the Tic-Tac-Toe game, known as *Tres en línea* or *Tres en Raya* in Spain.

The game has a 3x3 board on which crosses (‘X’) and zeros (‘0’) can be placed. The player who succeeds in placing three marks of the same kind in a horizontal, vertical, or diagonal row wins the game.

All the pieces (i.e. symbols) fit in a 3x3 square. The representation of the different pieces will depend on the symbol to draw, cross, or zero. Figure 1 shows an example of possible plays.



X		

X		
	0	

X	X	
	0	

X	X	0
	0	

X	X	0
X	0	

X	X	0
X	0	
0		

Figure 1. Play sequence for which the player with Zeros wins the game

When there are no empty squares on the board, the game finishes. There is a draw if no player has won yet at that moment.

The project has two parts: basic options and advanced options. Implementing the **basic** options allows the student to obtain up to 60% of the maximum grade. Implementing the **advanced** options will allow the student to obtain the 80% of the maximum grade. And implementing the **cpu** option will allow the student to obtain the 100% of the maximum grade.

Basic options

The basic behaviour of the application will correspond to the simulation of a Tic-Tac-Toe game in which two players participate. The players must select the position (square) where they want to place the symbol with which they are playing: Cross or Zero. The players must decide who will use crosses and who will use zeros. The player with the cross will start the game.

The board will be empty initially and the application will ask each player the position where a symbol must be placed. When a game finishes, the application will show the number of games won by each player (‘cross’ player and ‘zero’ player) and will ask the players whether they want to play again.

In summary, the board state information and the application functionality are as follows:

1. The board is empty initially.
2. The application will show the number of pieces used until a given moment.
3. The ‘cross’ player will always start a game.
4. The application will ask a player the position where a piece should be placed, first the row, and next to the column, e.g. (1,1). In this case, the piece must be placed in the square that corresponds to the first row and the first column of the board.
5. When a game finishes, the application will indicate if the game has been a draw or one of the players has won, specifying the winner (‘cross’ or ‘zero’ player).
6. The application will ask the players whether they want to play another game.
7. The application will store the data about the games won by each player. At the beginning of each game, the application will show the number of games won by each player and the number of draws.

A simulation example is shown below:

A Tic-Tac-Toe game is going to start. Cross (‘X’) starts and next Zero (‘0’) until the game finishes. You can play as many games as you wish.

Player X, please specify the position where you want to place a piece, taking into account that the board is 3x3, e.g. (2,3).

Placed pieces: 0

(1,1)

X		

Placed pieces: 1

Player 0, please specify your play; remember that the board is 3x3:

(2,1)

X		
	0	

Placed pieces: 2

Player X, please specify your play; remember that the board is 3x3:

(1,2)

X	X	
	0	

Placed pieces: 3

Player 0, please specify your play; remember that the board is 3x3:

(1,3)

X	X	0
	0	

Placed pieces: 4

Player X, please specify your play; remember that the board is 3x3:

(2,1)

X	X	0
X	0	

Placed pieces: 5

Player 0, please specify your play; remember that the board is 3x3:

(3,1)

<table>
<tr><td><table><tr><td>X</td><td>X</td><td>0</td></tr><tr><td>X</td><td>0</td><td></td></tr><tr><td>0</td><td></td><td></td></tr></table>Placed pieces: 6</td></tr>
<tr><td>Player 0 has won the game.
Do you want to play another game?</td></tr>
<tr><td>Yes</td></tr>
<tr><td>So far, Player X has won 0 game(s), Player 0 has won 1 game(s), and there has been 0 draw(s).
Another Tic-Tac-Toe game is going to start. Cross ('X') starts and next Zero ('0') until the game finishes. You can play as many games as you wish.
Player X, please specify the position where you want to place a piece, taking into account that the board is 3x3, e.g. (2,3).<table><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>Placed pieces: 0</td></tr>
<tr><td>(1,1)</td></tr>
<tr><td><table><tr><td>X</td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>Placed pieces: 1</td></tr>
<tr><td>Player 0, please specify your play; remember that the board is 3x3:</td></tr>
<tr><td>....</td></tr>
</table>

Advanced options

The advanced version of the game includes the following options when the application starts:

1. The players will be allowed to specify their name so that they can be referred to differently to Player X and Player 0; e.g. "Pedro" for Player X and "Ana" for Player 0.
2. The players will be allowed to select the symbol that they want to use in the game from this set: X,0,A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U, V,W,Y,Z,+,*,-,@,1.
   It will not be allowed to change the symbol during a game series; the application should be re-started.

CPU option

1. It will be possible to select 1-player or 2-player mode at the beginning of a play. In 1-player mode, the computer will be the second player and will place the pieces randomly. The computer will always use zero (‘0’).

Finally, the user(s) will select the game configuration, with the “basic”, “advanced” and “cpu” words.

Submission rules

1. The project can be developed by one, or two students.

2. The project will be presented on the 21st of December 2020, and will be submitted via Aula Global, no later than 23:55 CET.
   One of the members of the group will upload a single compressed file (.zip) with the name: **final_project_id1_id2**
   Where **id1,** and **id2,** are the student IDs of the students.

3. The compressed file will contain:
   a. A text or word or PDF document with (1) the name and Student IDs of the students and (2) a list of the options implemented. Also, any relevant explanation or assumptions made.
   b. **src** folder. Source files (**.java**).
   c. **bin** folder. Compiled files (**.class**).

4. The teacher will evaluate the project code only if it can be executed, that is**, the compiled files (.class) had been created and submitted**, **there are no compilation errors in the code**. Otherwise, the grade of the project will be 0.