



School of Sciences and Engineering  
Computer Science and Engineering Department

Fall 2020  
October 17<sup>th</sup>, 2020  
Programming in Java

## **Assignment 2 – Tic Tac Toe Game**

Name: Chris Amgad  
ID: 900170819

Name: Haroon Tharwat  
ID: 900152625

Submitted to:  
Prof. Sherif Aly

## Inputs

The user will first have to choose the Game Mode, either 'M' or 'A'. Then, they will be asked to enter the desired width( $m$ ), height ( $n$ ) and number of symbols required to win ( $k$ ) of the Tic Tac Toe game. Validations are used to ensure that the user does not input any illogical or invalid value.

```

Enter Game Mode...Type 'A' to play vs PC or 'M' for manually 2 users to play vs each other
A
Enter n (the number of rows in the game)... n must be bigger than or equal 1
4
Enter m (the number of columns in the game)... m must be bigger than or equal 1
3
Enter k (the number of symbols required to win the game)... k must be smaller than or equal either n or m
3
n is 4
m is 3
k is 3
#####
| |
-----
| |
-----
| |
-----
| |
-----
#####
Its the Users turn
On what Row do you want to play...choose from 0 to 3

```

## Assumptions

- In Manual mode, first player uses 'X' Symbol, and second player uses 'O' symbol.
- In Automatic mode, User uses 'X' symbol and Computer uses 'O' symbol.
- If number of symbols of same type repeated for  $k$  times, either vertically, horizontally, diagonally, or anti-diagonally, player is declared as a winner and the other player, is the loser.
- If number of turns equals the board size, it is a draw between the 2 players.

## Algorithm used in Automatic mode

- The very first move of the computer is random.
- Starting from the 2<sup>nd</sup> computer move, the algorithm will check if there is a vertically available chance to win, i.e.: if one O is left for the computer to win the game vertically, it places an O in the needed cell to win the game.
- In case the algorithm wasn't able to find a scenario, where it is 1 symbol away from winning vertically, the algorithm will then check the array to analyze if the user enemy has 1 symbol left away from winning the game. If detected, the computer algorithm blocks the cell. This checking is done horizontally, vertically, diagonally, and anti-diagonally within all array cells so that it will be hard for user to complete  $k$  number of Xs. For example, if a user has 2 'X' symbols beside each other and 3 symbols are needed

to win, the algorithm will detect this and block it by inputting an 'O' in the column next or before these series of symbols in order to prevent the user from winning.

- If the computer was still not able to find a scenario, where the user is 1 symbol away from winning, it will play a move through an algorithm. This algorithm defines the move to be 2 cells below the row containing the Furthest away O. In case, this decided cell was not empty, we keep adding 1 cell below till we find an empty cell. In the case, there was no empty cells till the end of the game array, we start counting from the first index in the array and we move 1 cell to the right till we find an empty slot for the computer to play the move.

## Replay Feature

After one of the players wins, the game will allow the user to replay the game from one point in the game to another point, where the user will input the values of the 2 points.

```
#####
x |  | x |
-----
o |  | x |
-----
  | o | o |
-----
x |  | o |
#####
** o has Won the game... Congratulations! **
At what point in the game do you want to replay from in the game?...You can choose from 1st point up to 8th point
Please Enter an Integer from 1 to 8
|
```