



# Cover page

**Embedded Systems Intake 40**

**Tic Tac Toc Project**

**Submitted to:**

**Dr: Eman Hisham**

**Done by:**

**Moustafa Ghareeb Muhammad**

**Ahmed Refaat Rashad Ali**

**Mostafa Nader Ibrahim Eltorky**

**Mostafa Sayed Abdel-Hamid Abdel-Latif**

**Mohamed Ibrahem Saad Mahmoud**

# Table of Contents

[Cover page 1](#_Toc26663988)

[Table of Contents 2](#_Toc26663989)

[Table of figures: 3](#_Toc26663990)

[Objectives: 4](#_Toc26663991)

[Description: 5](#_Toc26663992)

[Overview: 6](#_Toc26663993)

[Players: 6](#_Toc26663994)

[Theory of Game: 6](#_Toc26663995)

[Core Logic: 7](#_Toc26663996)

[First move: 7](#_Toc26663997)

[Second move: 8](#_Toc26663998)

[Case 2: 8](#_Toc26663999)

[Case 3: 9](#_Toc26664000)

[Case 4: 9](#_Toc26664001)

[Third and fourth move: 9](#_Toc26664002)

[Core Logic - Humans: 10](#_Toc26664003)

[Classes: 10](#_Toc26664004)

[Class Main: 10](#_Toc26664005)

[Class NewGame: 10](#_Toc26664006)

[Future plan: 10](#_Toc26664007)

[Conclusion: 11](#_Toc26664008)

[Reference: 12](#_Toc26664009)

# Table of figures:

[Figure 1 6](#_Toc26663925)

[Figure 2 7](#_Toc26663926)

[Figure 3 7](#_Toc26663927)

[Figure 4 7](#_Toc26663928)

[Figure 5 8](#_Toc26663929)

[Figure 6 8](#_Toc26663930)

[Figure 7 8](#_Toc26663931)

[Figure 8 8](#_Toc26663932)

[Figure 9 9](#_Toc26663933)

[Figure 10 9](#_Toc26663934)

[Figure 11 9](#_Toc26663935)

[Figure 12 10](#_Toc26663936)

[Figure 13 10](#_Toc26663937)

# Objectives:

Our project name is Tic-Tac-Toe game. This game is very popular and is fairly simple by itself. It is actually a two players game. In this game, there is a board with *n* x *n* squares. In our game, it is 3 x 3 squares. The goal of Tic-Tac-Toe is to be one of the players to get three same symbols in a row - horizontally, vertically or diagonally - on a 3 x 3 grid.

# Description:

Tic Tac toe (also known as X and O) is a game for two players, X and O, who takes turn marking a 3×3 grid. The player who can place three of their marks in a horizontal, vertical or diagonal is the winner.

The project has been implemented using java as programming language, keyboard, and joystick as hardware (input methods). We use java fx to provide the game with an interactive GUI (Graphical user interface) and sounds and we use an external library called jinput to interface with the joystick. The user can play using keyboard or the joystick. The user to choose the level of the hardness of the game easy, medium and hard mode, to choose the preferable theme, or to switch on/off the background sound and to pause and continue the game.

We provide the game with an interactive (GUI) Graphical user interface and sounds.

The game is provided with two options for interfacing the first one is the keyboard and the second is the joystick.

The game is consisted of three modes easy, medium and hard.

In the easy option, the computer put X or O in a random way, in the medium mode the computer try to block the user from winning and in the hard mode the computer try to win and block the user from winning.

There are options in the game the first is play vs the computer and the other one is two player.

The user can change the theme and switch off/on the background sound.

The user can pause and continue the game whenever he wants.

# Overview:

This game can be played in a 3x3 grid (shown in the figure 2.1). The game  
can be played by two players. There are two options for players:

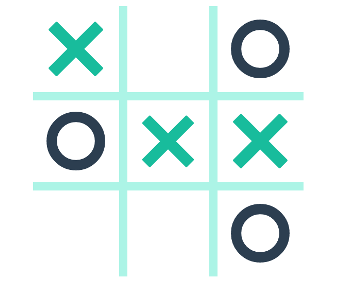


Figure 1

## Players:

For the option human, both the players are human and for the option computer, the first player is human and the second player is computer.

## Theory of Game:

A player can choose between two symbols with his opponent, usual games use “X” and “O”. If first player chooses “X” then the second player have to play with “O” and vice versa.

A player marks any of the 3x3 squares with his symbol (may be “X” or “O”) and his aim is to create a straight line horizontally or vertically or diagonally with two intensions:  
a) Create a straight line before his opponent to win the game.  
b) Restrict his opponent from creating a straight line first.  
In case logically no one can create a straight line with his own symbol, the  
game results a tie. Hence there are only three possible results – a player wins, his opponent (human or computer) wins or it’s a tie.

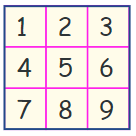


Figure 2

If any player is able to draw three Xs or three Os in the following combinations then that player wins. The combinations are:

|  |  |
| --- | --- |
| a) 1, 2, 3 | b) 4, 5, 6 |
| c) 7, 8, 9 | d) 1, 4, 7 |
| e) 2, 5, 8 | f) 3, 6, 9 |
| h) 1, 5, 9 | i) 3, 5, 7 |

# Core Logic:

There are two core logics in this game – when both players are human, and when one is computer. Suppose the player use X and the computer use O. The logic used for the AI is as follows:

First move:  
a) If the center is free, get the center. (Figure: 3)  
b) Otherwise, get any of the corners. (Figure: 4)

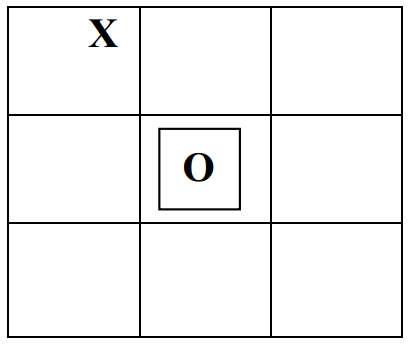


Figure 3

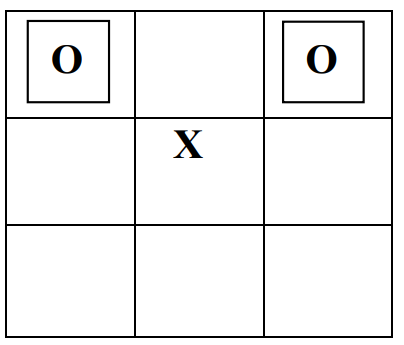


Figure 4

## Second move:

a) Block user from winning. (Figure: 5).  
 b) Option for winning by applying the following logic - If the center is occupied by user, get any of the corners. (Figure: 6)

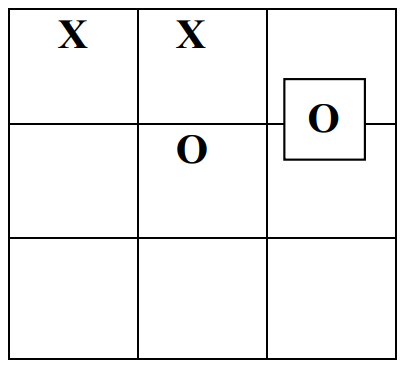


Figure 5

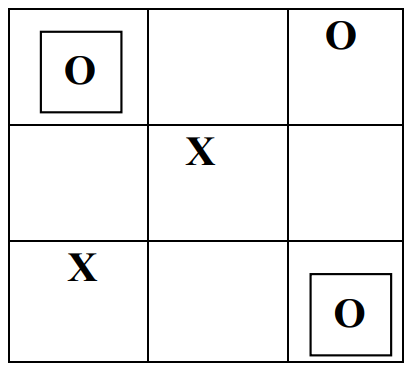


Figure 6

Otherwise, the following cases happen:  
Case 1:

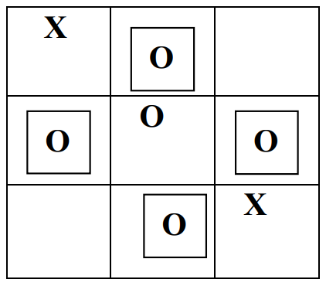


Figure 7

If any situation arises like the figure 7 then the computer sets its symbol any one of the positions among 2, 4, 6 and 8.

### Case 2:

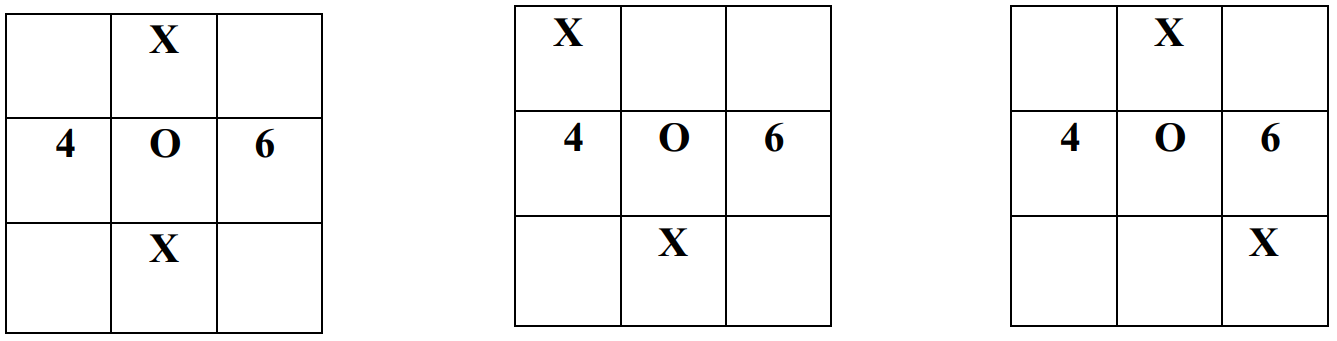


Figure 8

If any situation arises like the figure 3.6 or figure 8 then the computer sets its symbol at any position among 4 and 6.

Case 3:

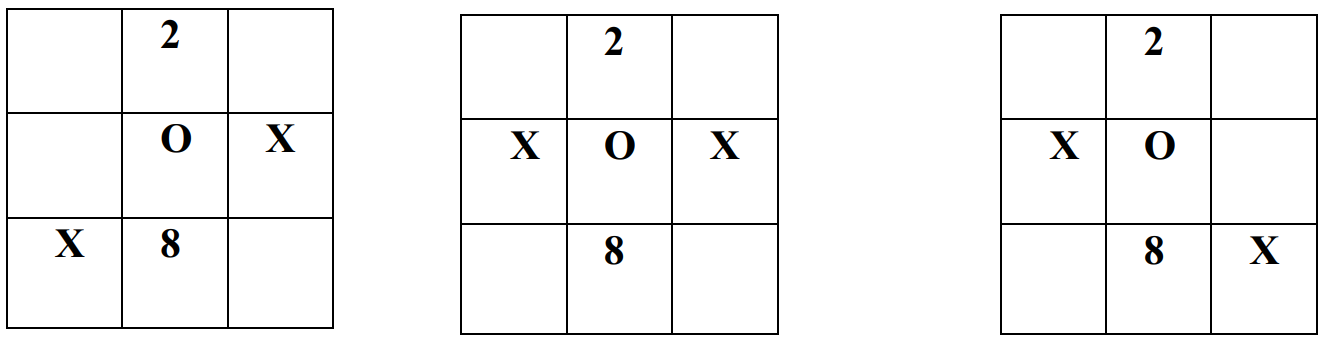


Figure 9

If any situation arises like the figure 9 then the computer sets its symbol at any position among 2 and 8.

Case 4:

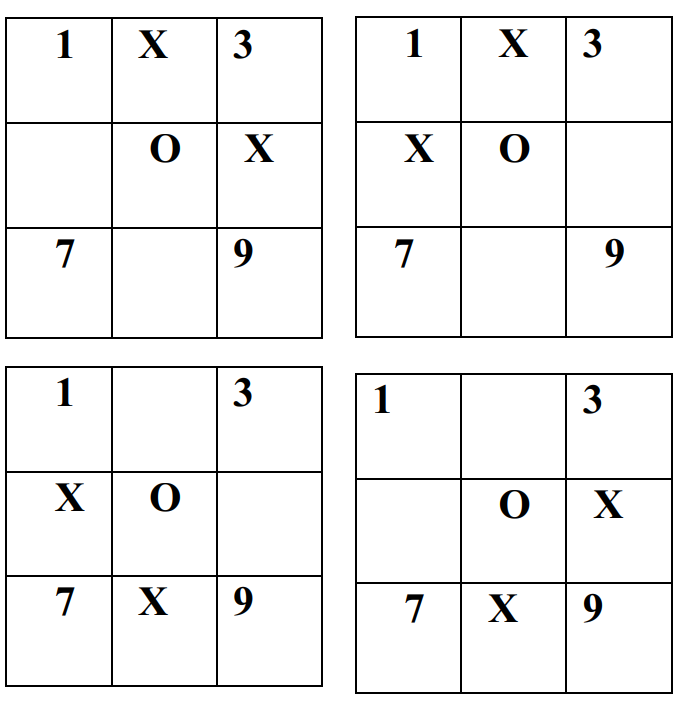
****

Figure 10

If any situation arises like the figure 10 or 3.15 then the computer sets its symbol at any position among 1, 3, 7 and 9.

Third and fourth move:  
a) Option for winning. (Figure: 11)  
b) Block user from winning. (Figure: 12)  
c) Randomly play a move. (Figure: 13)

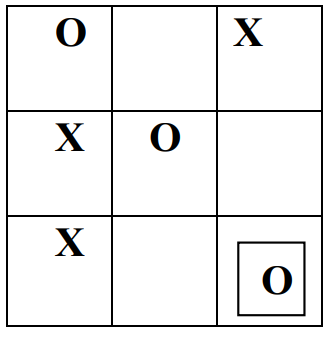


Figure 11

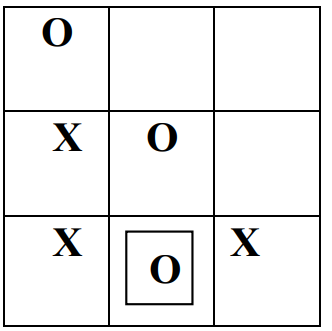
****

Figure 12

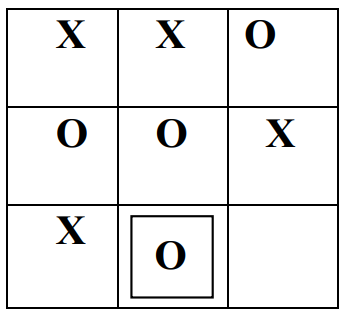
****

Figure 13

# Core Logic - Humans:

For each move, check whether any 3 combination is occupied by any player and display the winner accordingly.

Classes:  
There are two classes in our program. first class is Main.java and another is NewGame.java.

Class Main:  
In this class the main task of the game is done.

**Picture of the game**

Class NewGame:  
This class is used to show the dialog for choosing options.

**Picture of the game**

Future plan:  
1. Mouse functions will be added.  
2. We want to design more complex boards for the game in future and make it portable to smaller boards such as raspberry pi.

# Conclusion:

We finally implement a Tic tac toe game (X O) using java application with many features like 3 level of hardness (easy medium hard),2 ways of interfacing (keyboard and joystick),2 ways for playing the first one is user vs computer and the second one is 2 player mode. The user can pause, continue or changing the theme and the sound.

Reference:  
Books:  
H.M.Deitel and P.J.Deital, Java How to program: Sixth Edition Herbert Schildt, The Complete Reference: Fifth edition

**Aim:**

This project consists of developing and implementing a computer program that plays tic-tac-toe against another player.

**Abstraction:**

Tic-tac-toe is a two players game (one of them being your computer program). The two players take turns putting marks on a 3x3 board. The player who first gets 3 of his/her marks in a row (vertically, horizontally, or diagonally) wins the game, and the other loses the game.

A game will consist of a sequence of the following actions:

* Initially, program should ask the user which marks the user prefers ("X" or "O"). The player that gets to play with the "X" marks will play first (we call him/her player 1) and the player that gets to play with the "O" marks will play second (we call him/her player 2).
* Player 1 and 2 take turns making moves. A move (mark row column) should satisfy the following constraints:
* Move preconditions
* mark is "X" if it is player 1's turn and "O" if it is player 2's turn.
* The position (row, column) on the board is empty.
* Move postconditions
* After the move, the position (row, column) on the board will be occupied by a mark.
* It will be the other player's turn.
* The game ends when either:
* one of the players wins the game, i.e. this player gets three of his/her marks in a row (vertically, horizontally, or diagonally).
* all the positions on the board are occupied. In this case, the game ends in a draw.