

Artificial Intelligence Lab

Project Report

**Topic:**

Tic Tac Toe

Submitted to:

Aizaz Akmal

Submitted by:

Muhammad Abdullah(2020-CS-627)

Shaheer Arshad(2020-CS-639)

Muhammad Jafar(2020-CS-644)

Degree & Semester:

BSCS (5th Semester) A

**University of Engineering and Technology, New Campus,
Lahore**

Table of Content

| | |
|------------------------------------|---|
| 1. Introduction ----- | 3 |
| 2. Rules ----- | 3 |
| 3. How to Play ----- | 4 |
| 4. Implementation Algorithms ----- | 4 |
| 4.1 Minimax ----- | 4 |
| 5. Features ----- | 4 |
| 6. Implementation Details----- | 4 |
| 6.1 Libraries Used ----- | 4 |
| 6.2 Classes ----- | 4 |
| 6.3 Functions ----- | 4 |
| 6.4 Game Images ----- | 5 |
| 7. Summary----- | 7 |

1. Introduction:

Tic-Tac-Toe is one of the paper-and-pencil games. This game requires two players in 3x3 grid with Player 1 and 2 acts as their respective symbols. The objective of this game is to take place of three connecting grids in a horizontal, vertical, or diagonal way.

Tic-tac-toe is a casual strategy game for everyone. It is easy to teach, and the player can play this game almost everywhere. Because it only requires a writing tool and a writable surface. Vandalism is not recommended.

This game was first introduced at ancient time, however, there is no evidence who invented it and in which year. Some people think this game was invented at Ancient Egypt, and then Roman Empire called this game “Terni Lapilli”. The grid drawing for Egypt, and then Roman Empire called this game “Terni Lapilli”. The grid drawing for the game had been found chalked all over the ancient city’s ruins.

In 1952, Alexander S. Douglas for the EDSAC computer at University of Cambridge developed a computerized Tic-Tac-Toe game called “OXO”. This was the first video game of Tic-Tac-Toe and it has AI inside, therefore human could play against the computer opponent.

2. Rules:

The game is to be played between two people (in this program between HUMAN and COMPUTER).

The players choose their markers to mark respective cells.

The game starts with one of the players and the game ends when one of the players has one whole row/ column/ diagonal filled with his/her respective character (‘O’ or ‘X’).

A player can play perfect tic-tac-toe (win or draw) given they move according to the highest possible move from the following table.

1. Win: If the player has two in a row, play the third to get three in a row.

2. Block: If the opponent has two in a row, play the third to block them.

3. Fork: Create an opportunity where you can win in two ways.

4. Block opponent's fork:

a. Option 1: Create two in a row to force the opponent into defending, as long as it does not result in them creating a fork or winning. For example, if "X" has a corner, "O" has the center, and "X" has the opposite corner as well, "O" must not play a corner in order to win. (Playing a corner in this scenario creates a fork for "X" to win.)

b. Option 2: If there is a configuration where the opponent can fork, block that fork.

5. Center: Play the center.

6. Opposite corner: If the opponent is in the corner, play the opposite corner.

7. Empty corner: Play in a corner square.

8. Empty side: Play in a middle square on any of the 4 sides.

The first player, whom we shall designate "X", has 3 possible positions to mark during the first turn. Superficially, it might seem that there are 9 possible positions, corresponding to the 9 squares in the grid. However, by rotating the board, we will find that in the first turn, every corner mark is strategically equivalent to every other corner mark. The same is true of every edge mark. For strategy purposes, there are therefore only three possible first marks:

corner, edge, or center. Player X can win or force a draw from any of these starting marks; however, playing the corner gives the opponent the smallest choice of squares which must be played to avoid losing.

The second player, whom we shall designate "O", must respond to X's opening mark in such a way as to avoid the forced win. Player O must always respond to a corner opening with a center mark, and to a center opening with a corner mark. An edge opening must be answered with either a center mark, a corner mark next to the X, or an edge mark opposite the X. Any other responses will allow X to force the win. Once the opening is completed, O's task is to follow the above list of priorities in order to force the draw, or else to gain a win if X makes a weak play.

3. How to Play:

- ❖ Use any method to find who will start first.
- ❖ Then each player taking turn draw their symbol on a space from those nine possible spaces.
- ❖ The one who met the winning condition first win. If after nine possible spaces are used, but no one wins, that game is a draw.

4. Implementation Algorithms:

In this game, we use one algorithms. The detail of the algorithms is given below:

4.1 Minimax Algorithm:

Minimax is a kind of backtracking algorithm that is used in decision-making and game theory to find the optimal move for a player, assuming that your opponent also plays optimally. It is widely used in two player turn-based games such as Tic-Tac-Toe, Backgammon, Mancala, Chess, etc.

In Minimax the two players are called maximizer and minimizer. The maximizer tries to get the highest score possible while the minimizer tries to do the opposite and get the lowest score possible.

5. Features:

- a. Single Player (vs AI).
- b. Dual Player.
- c. Sounds.
- d. Themes.

6. Implementation Detail:

Implementation details of the project are as follow:

6.1 Libraries Used:

- **pygame:** Python Library to provide user interface for game graphics.
- **numpy:** Python Library to make arrays and use relative functions.

6.2 Classes:

- **Board:** make board, mark position and check winnings.
- **AI:** class to implement the minimax algorithm.
- **Game:** class to wrap AI and Board to make the game work on panel.

6.3 Functions:

- **Board:**
final_state: to check the board in final form for winning or draw.

mark_sqr: to mark player's mark on board.
 empty_sqr: to clear the mark.
 get_empty_sqr: to clean the board for restart.
 isfull: to check board is full.
 isempty: to check if board is empty.

- **AI:**

rnd: to generate random child.
 minimax: to generate child using algorithm.

- **Game:**

show_lines: to display grid on panel.
 draw_fig: to make marking graphics.
 make_move: to generate a special figure at special place.
 next_turn: to switch move between players.
 change_gamemode: to switch between P vs P or P vs AI.
 isover: to check if game is over.
 reset: to reinitialize the game.

win: to play music in case of game over.
 beep: to play music in case of click or key press.
 main_menu: to show main menu on the screen.
 draw_text: to show text on the screen.
 main: to wrap up all the classes and begin the fun.

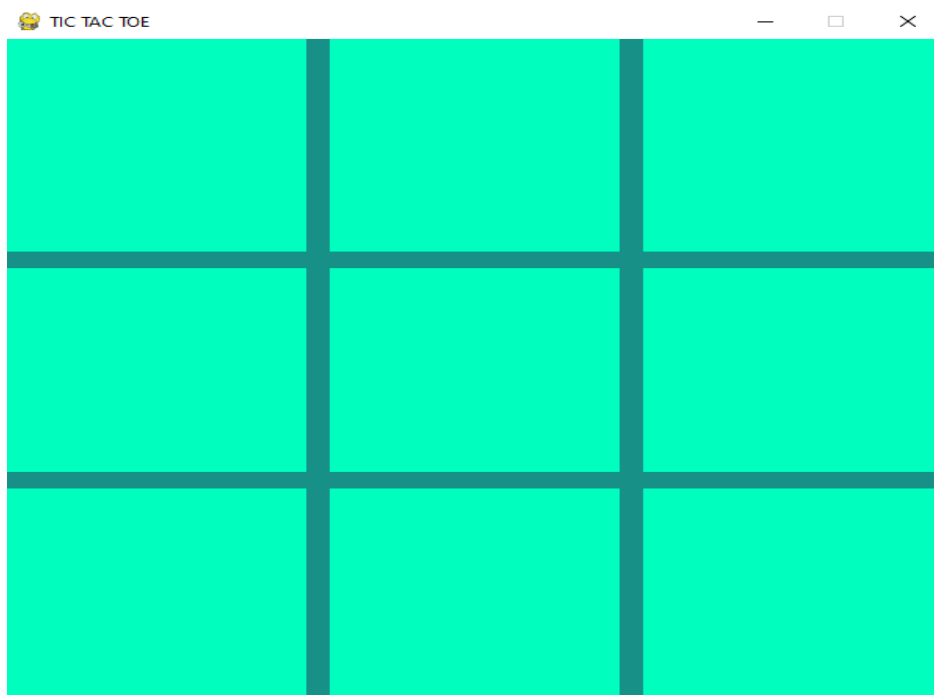
6.4 Game Images:

```
select figure for player 1
[1]Cross
[2]Square
[3]Ellipse
2
select figure for player 2
[1]Circle
[2]Rectangle
[3]Polygon
2
```

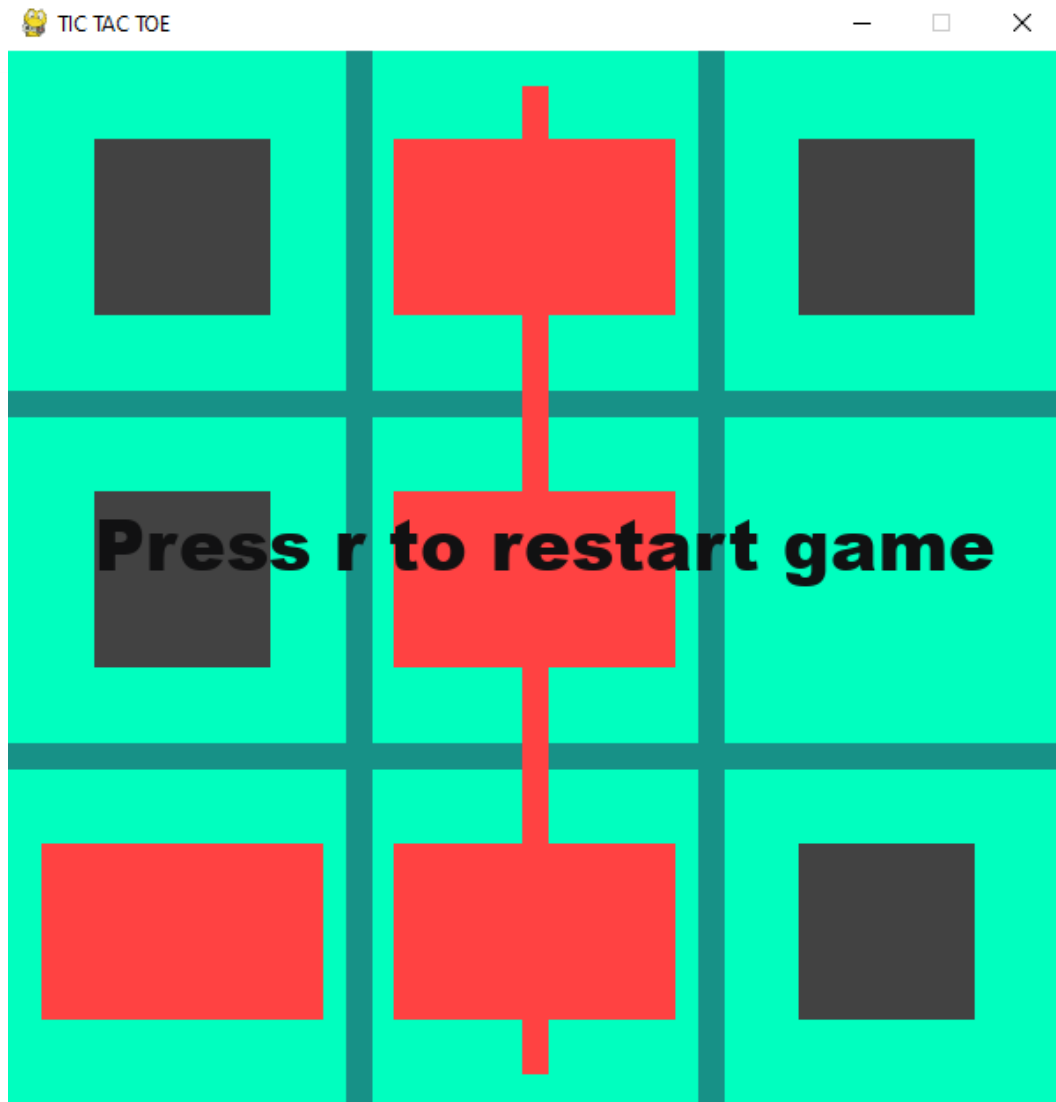
Choosing Player Graphics



Main Menu



Board



Gameplay

7. Summary:

Tic-Tac-Toe game is a traditional game that is still played until present day. All algorithm that used in this game have the same purpose, to block the opponent's way. Each developer has his/her own style of algorithm. However, in order to be more effective, the basic of algorithm should include all aspects based on Newell and Simon's 1972 Tic-Tac-Toe.