TIC-TAC-TOE IN C

Pavitha N[1], Tanushri Bhuruk[2]

[1,2]*Department of Artificial Intelligence & Data Science (AI&DS)*

Abstract— — The square block (3 x 3) can be filled with a cross (X) or a circle (O) in the Tic-Tac-Toe game, which can be played by two players (O). By allowing each player to mark their move, the game will switch between the players. When one of the players makes a horizontal, vertical, or diagonal line with three identical markers, the program displays which player has won, whether X or O. The game is constructed such that two players can play tic-tac-toe by entering the position to place their marks on the board. The software will have a display function and a select function for placing the symbol as well as a toggle function for switching between the symbols, allowing each participant to take a turn playing the game. After each player makes a move, the computer will update and check for game circumstances as the game progresses.

KEYWORDS: - Tic-Tac-Toe, Virtual Game, Toggle Turns, Display Winner, etc.

I. INTRODUCTION

People began to show interest in using their computers as a source of entertainment in the 1970s, resulting in the emergence of a multibillion-dollar game business. The origin is uncertain, though ancient Egyptians claim to have discovered the Tic-Tac-Toe pattern etched on rocks around 3500 years ago. Later, they discovered how much fun it was to play a game using this pattern. The game became popular after that, and it could be played on a wooden board, a table, or even a piece of paper. Tic-Tac-Toe is a popular board game in which two players alternately place X's and O's on a 3x3 grid. The players must first pick who would go first and who will record his actions with an X. The game continues with the opponents placing their marks on any unoccupied cell. The goal of the game is for the first player to collect three marks in

a row, which might be vertical, horizontal, or diagonal. If all of the cells are filled, the game ends in a tie. Using the C programming language, we created a two-person tic tac toe game in which the player who forms a horizontal, vertical, or diagonal sequence of three marks wins. When a player is successful in forming a vertical, diagonal or horizontal sequence of either X or 0, the game declares the winner. The system has been designed in such a way that the 3X3 grid will be displayed first along with the positions on each square. The first player will then have to enter the position in which he/she wants to place the mark in. The turns are toggled until any one of the player is successful in making a sequence of 3 consecutive marks either vertically, horizontally or diagonally. And finally the winner is declared.

II. LITERATURE REVIEW

We have reviewed the tic-tac-toe research articles and rewrote the findings below.

Tic-Tac-Toe is a straightforward but fun board game. The Tic-Tac-Toe game is learned using a variety of methods. Fok and Ong [3] and Grim et al. [4] For example, they developed strategies based on a neural network of artificial intelligence to play them.

Citrenbaum [5] and Yakowitz [6] discuss games like Tic-Tac-Toe, such as Go-Moku, Hex, and Bridg-It.

Traditionally, the Tic-tac-toe game is a pencil and paper game played by two people taking turns placing their pieces on the 3rd grid with the intention of becoming the first player to complete a horizontal, vertical, or diagonal line. Row with their pieces

Many versions of the Tic Tac game software have been recorded, and have recently been made available on smart phones, including the Apple iPhone [7] and the Android environment [8].

The integrated digital circuit design of Stephen Mann and Matthew Netsch [9] to perform neural network (NN) calculations to explore the Tic-Tac-Toe area was presented. By literally mapping the routes between sensible gates on each chip, FPGAs can create modern digital designs.

Shahzeb Siddiqui et al [10] has used another NN application that extends the game by adding two new lines, two additional columns, and a third dimension. The paper lists the best place to use the concept of developing a neural network that combines the spread of backpropagation combined with parts of the genetic algorithm to improve the chances of finding the best solution and highlighting our methods of use.

Pinaki Chakraborty [11] officially created the Tic-Tac-Toe game and created its own creative techniques based on artificial intelligence. By allowing the high-pitched voice of the classic movement, Leaw and Cheong [12] made a modest imitation of the old tic-tac-toe game.

Edward [14] uses an optically subjugated gate array (OPGA) to demonstrate the efficiency of electrooptical circuits that incorporate human input, display, and sensible power into a single device using a simple Tic-tact-toe game. There are additional updates on route strategies, designs, and logical simulations.

Alauddin [15] introduces the hardware introduction of the smart Tic-Tac toy. Graphical Liquid crystal display (GLCD) touch screen and small controls are used for operation. The small controller accepts player movement on the GLCD (shown as X) and analyzes it using a sophisticated algorithm to determine the best calculation action. The next opposing movement is printed on the screen by a small circle (O).

Tic Tac Toe is a two-person game, as described by Agustia.M and Amri.P [16], in which one player represents the opposite (o) and the other player demonstrates the opposite (x). The participant draws nine grids on a piece of paper or anywhere else he or she likes. This is a game where two players draw crosses and circles in one compact grid compartment with nine spaces; the goal is to get a line of three crosses or three circles before your opponent.

III. METHODOLOGY

i. SYSTEM REQUIREMENTS:-

a. **CLion**- CLion is a fully Integrated Development Environment for coding in C and C++ on Windows, Linux, and macOS.

b. C programming language

c. draw.io for making the flowchart

ii. STEPS:-

1) Creating a 1D array to store the elements

2) Creating functions –

create_board(for creating the 3X3 board) marking_board(for marking X's and O's on the board)

check_for_win (checking and declaring the winner)

iii. PSEUDO CODE OF THE FUNCTIONS USED :-

i. create_board()

```
void create_board()
{
  clear_screen;
  printf("\n\n\tTic Tac Toe\n\n");
  printf("Player 1 (X) -- Player 2 (O)\n\n");
  printf("  |  |  \n");
  printf(" %c | %c | %c \n",box[1],box[2],box[3]);
  printf("___|___|___\n");
  printf("  |  |  \n");
  printf(" %c | %c | %c \n",box[4],box[5],box[6]);
  printf("___|___|___\n");
  printf("  |  |  \n");
  printf(" %c | %c | %c \n",box[7],box[8],box[9]);
  printf("___|___|___\n");
  printf("  |  |  \n");


}
```

ii. marking_board()

```
void marking_board( int choice, char mark)
{
  if(choice==1 && box[1]=='1')
    box[1]=mark;
  else if(choice==2 && box[2]=='2')
    box[2]=mark;
  else if(choice==3 && box[3]=='3')
    box[3]=mark;
  else if(choice==4 && box[4]=='4')
    box[4]=mark;
  else if (choice==5 && box[5]=='5')
    box[5]=mark;
  else if (choice==6 && box[6]=='6')
    box[6]=mark;
  else if (choice==7 && box[7]=='7')
    box[7]=mark;
  else if (choice==8 && box[8]=='8')
    box[8]=mark;
  else if (choice==9 && box[9]=='9')
    box[9]=mark;
  else
  {
    printf("Invalid move");
  }
}
```

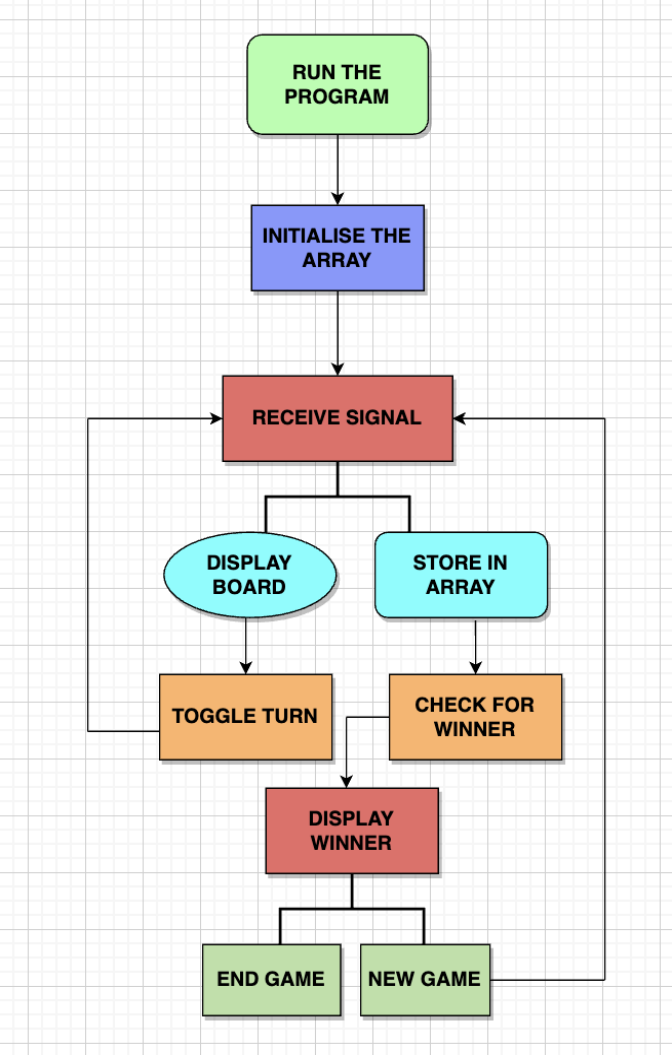
iii.check_for_win()

```
int check_for_win()
{
  if(box[1]==box[2] && box[2]==box[3])
```

```
    return 1;
  else if(box[4]==box[5] && box[5]==box[6])
    return 1; // horizontal match
  else if (box[7]==box[8] && box[8]==box[9])
    return 1;

  else if (box[1]==box[4] && box[4]==box[7])
    return 1;
  else if(box[2]==box[5] && box[5]==box[8]) //
vertical match
    return 1;
  else if(box[3]==box[6] && box[6]==box[9])
    return 1;


  else if(box[1]==box[5] && box[5]==box[9])
    return 1;
  else if(box[3]==box[5] && box[5]==box[7])
//diagonal match
    return 1;

  else if(box[1]!= '1' && box[2]!= '2' && box[3]!= '3' &&
box[4]!= '4'&& box[5]!= '5' && box[6]!= '6'&& box[7]!= '7'
&& box[8]!= '8' && box[9]!='9') //no match
    return 0;

  else
    return -1;


}
```

iv. Workflow

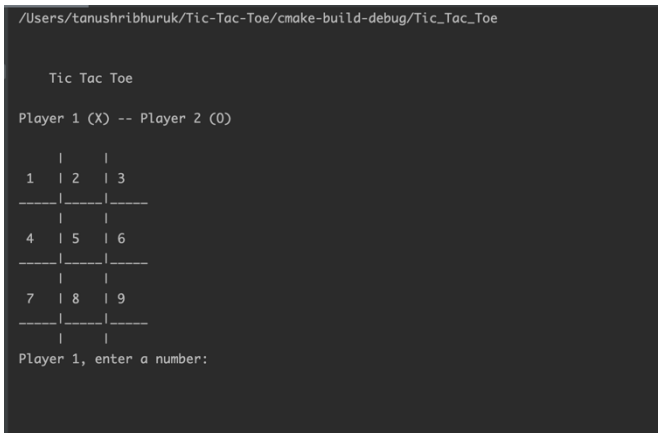
The workflow of the game has been demonstrated in the flowchart given below –



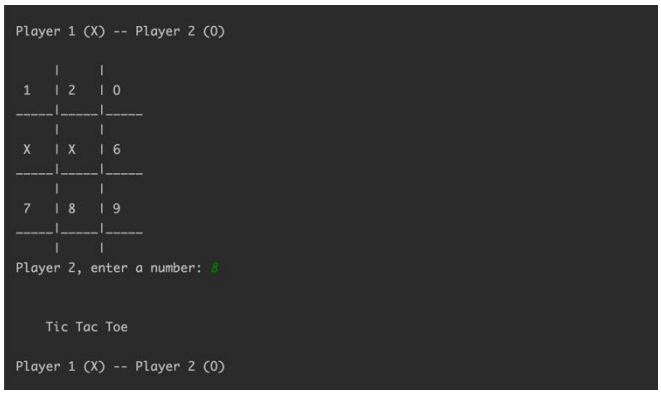
IV. RESULTS & DISCUSSION

We have successfully designed a 2-player Tic Tac Toe game in C.

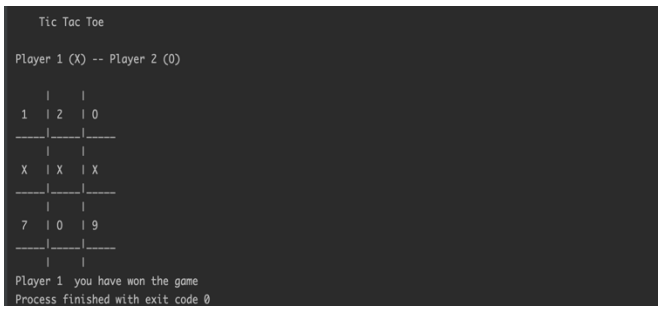
As we run the program, the board will be displayed in the following way where it will be shown that player 1 needs to use X and player 2 needs to use 0.



The 3X3 game board is displayed with the positions specified on each block.



In the next step, the player is allowed to enter their symbol which is either a X or 0 in their desired and vacant position.



If any of the players is successful in placing their symbol in either a vertical, horizontal or diagonal format, he/she will be declared as the winner.

V. LIMITATIONS

The game falls short of a better GUI for serene user interaction. It does not have a login system for storing the users' information and scores of previous games.

VI. FUTURE SCOPE

Instead of displaying the output on the console, an attractive GUI can be added to the game with a login and signup system for storing previous data. The game can also be equipped with different levels such as easy, medium and difficult.

VII. CONCLUSION

We have built a Tic-Tac-Toe using the C programming language. Tic Tac Toe is a game that is played by people of all ages. It has been developed using the C programming language. Any purpose-driven decision maker can have intelligence as a trait. A Tic Tac Toe algorithm has been presented in the system and tested, and it has proven to be effective. Chiefly, the system is free of errors.

VIII. REFERENCES

1) Al-Khateeb, B., Tic-tac-toe evolutionary actor, International Journal of Reasoningbased Intelligent System, Vol. 4, No. 4, pages 182-185, 2012.
2) Garg, R., Nayak, D., Tic-Tac-Toe Game: Simulation using Min-Max Algorithm,International Journal of Advanced Research in Computer Science, Vol. 8, No. 7, pp.1074-1077, 2017.
3) Hochmuth, G., On the Genetic Evolution of a Perfect Tic-Tac-Toe Strategy, Stanford University Bookstore, pages 75-82, 2003.
4) Ling, S., H., Lam, H., K., Play Tic-Tac-Toe using the Genetic Neural Network with Double Transfer Functions, Journal of Intelligent Learning Programs and Applications, Vol. 3, pages 37-44, 2011.
5) Mohammadi, H., Nigel, P., A., Venetsanopoulos, A., Santos, M., Evolving Tic-Tac-Toe. Playing algorithms using Co-Evolution, Interactive Fitness and Genetic Programming, International Journal of Computer Theory and Engineering, Vol. 5, No. 5, pages 797-801, 2013.
6) Pilgrim, A., R., Tic-Tac-Toe: Introductory Specialist Systems for Middle School Students, Summer Computer Science Workshop, pp.340-344, 1995.
7) Sharma, N., Tyagi, S., Atri, S., A Comparative Analysis of Min-Max and Max-Min Algorithms based on the Makespan Parameter, International Journal of Advanced Research in Computer Science, Vol. 8, No. 3, pp.1038-1041, 2017.
8) Von Neumann, J. (1956) The General and Logical Theory of Automata. In: New- man, J.R., Ed.,
9) The World of Mathematics, Vol. 4, Simon and Schuster, New York, 2070-2098.

10) Waters, D.P. (2012) Von Neumann's Theory of Self-Reproducing Automata: A Useful Framework for Biosemiotics? Biosemiotics, 5, 5-15.
11) Von Neumann, J. (1956) the General and Logical Theory of Automata. In: New-man, J.R., Ed., The World of Mathematics, Vol. 4, Simon and Schuster, New York, 2070-2098.
12) Waters, D.P. (2012) Von Neumann's Theory of Self-Reproducing Automata: A Useful Framework for Biosemiotics? Biosemiotics, 5, 5-15.