Tic-Tac-Toe using Socket Programming

Gandivalasa Keerthi Tej
AM.EN.U4AIE20027
Dept. of Computer Science(AI)
Amrita Vishwa Vidyapeetham
Amritapuri Campus

Ponnuru Sathwik Venkata Gowtham
AM.EN.U4AIE20064
Dept. of Computer Science(AI)
Amrita Vishwa Vidyapeetham
Amritapuri Campus

Kalla Yadhu Vamsi
AM.EN.U4AIE20039
Dept. of Computer Science(AI)
Amrita Vishwa Vidyapeetham
Amritapuri Campus

***Abstract*—** As part of a case study project employing Artificial Intelligent approaches, Tic-Tac-Toe is created on a computer-based flatform using Python. The primary goal of this project is to create a computer artificial intelligence based on a 3x3 Tic-Tac-Toe game that shows two players who will win and lose the game, accordingly, using the Socket Programming. It was adopted and modified as a subset of rules from best gameplay practices: (l) attempt to win, (2) endeavor to keep a misfortune, (3) make a key move, and (4) make a connection between client and server. Probabilities that the computer would locate a valuable move and disregard it is provided to make the game more entertaining and winnable at easier difficulty levels. Finally, the player uses artificial intelligence chooses its next move using a basic move and the gameplay is fast, balanced, and pleasant. Furthermore, using symbolic model checking, illustrate the winning strategy in a chess game, and show the winning strategy in a tic-tac-toe game using the symbolic model checking tool improved with the winning strategy verification method.

Through our project we intend to carry out the game using socket programming. We implemented a server code and client code. By running those codes in two different devices, the players can play the game in their respective terminal outputs.

Keywords - Tic -Tac -Toe, Design, Python, Socket programming.

I. INTRODUCTION

A. Tic-Tac-Toe

Tic-Tac-Toe is a two-player game where the square block (3 x 3) can be filled with either an X or a circle (O). The game will switch between players by enabling each player to mark their move. When one of the players makes a horizontal, vertical, or diagonal line with three identical markers, the program displays which player has won, X or O. The game is designed for two players to play tic-tac-toe with each other.

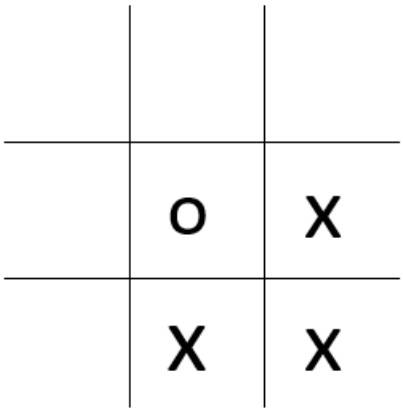

Fig.1. Tic-Tac-Toe

B. Socket Programming

Sockets allow two separate processes on the same or distinct machines to communicate with each other. Sockets were first introduced in 2.1BSD and developed into their current form in 4.2 BSD. The most common type of socket applications are client-server applications, where one side acts as the server and waits for connections from clients. This is the type of application that you'll be creating in this tutorial. It is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while the other socket reaches out to the other to form a connection. The server forms the listener socket while the client reaches out to the server.

C. Main Objective

This project's goal is to create a Tic-Tac-Toe game. The game is designed to have two parts: a single player game 'X' and a player game 'O' (both are group members or anyone eager to join a game) that compete against each other. The object of the diversion is to line up four 'X's' or 'O's' pieces vertically, evenly, or corner to corner in a row. If this is completed, the game ends in a win; otherwise, after no more moves are possible, the game ends in a tie.

II. BACKGROUND STUDY AND RELATED WORKS

The game of tic-tac-toe will be described in depth in this section. The fundamental rules of the game will be discussed first. Then there will be a review of existing Tic-Tac-Toe games, which will lead to a discussion of current Tic-Tac-Toe fashions and the recommended work model. Finally, this section will be completed with a science search focused on the Python programme. The first known form of tic-tac-toe is thought to have originated in the Roman Empire. It dates back to the first century BC (Crowley, 1993).

The first known variation of tic-tac-toe is thought to have originated under the Roman Empire. It dates back to the first century BC (Crowley, 1993). Terni Lapilli was the name of the sport at the time. Instead of having an unlimited number of pieces, each player was given only three. The game was played by moving them around to different unoccupied spaces. Tic-Tac Toe, on the other hand, is said to have originated in ancient Egypt, according to Claudia Zaslavsky's book (Zaslavsky, 1982). Around the first century BC, an early variant of tic-tac-toe was played in the Roman Empire.

Terni Lapilli was its previous name, and it was known for being averse to having any variety of pieces 5. Because each player only possessed three, they had to shuffle them about to free up spaces in order to continue playing. The framework markings for the game have been discovered chalked all over Rome. However, according to Claudia Zaslavsky's book Tic-Tac Toe: And Other Three-In-A-Row Games from Ancient Egypt to the Modern Computer, tic-tac-toe may have originated in ancient Egypt. In the year 1864: The main print reference to the British name "nougats and crosses." In 1884, the first print mention of the game "tick-tack-toe" appeared, however it was only referred to as "a kids' entertainment played on a slate, consisting in striving with the eyes closed to bring the pencil down on one of the quantities of a set, the number hit being scored." "Tic-tac-toe" may also derive from "tick-tack," the name of an archaic backgammon variant first depicted in 1558. In the twentieth century, the United States renamed nougats and crosses tic-tac-toe.

III. METHODOLOGY

A. Server

- The game server listens for and accepts clients connections. The server mainly acts as a forwarder by transmitting the data between the Players.
- The server doesn't do any logic — such as executing game rule, coordinating player’s turn and computing scores because we have handled those at the client application for improved communication efficiency.
- We are defining a function for taking input of the Player 1 and Player 2 respectively.
- We are defining another three functions to check the rows, columns and diagonals in the given matrix.
- We are also naming another function to know whether the player who won or the player got lost. Also Starting a server which gives us a connection to bind to port.

- We must make sure the players are connected through the Host IP Address and port number.
- We also created a function which starts the game and allows the players to give input alternatively.
- It displays the common messages to the players whether the game has been drawn, won or lost.

B. Client

- The Tic-Tac-Toe game client application provides an intuitive user interface for connecting to the server and playing the game.
- Initially a connect "frame" is opened where a player enters the name and connects to the server. After connecting, the connect frame is hidden and the main game window is displayed.
- We already defined the default colour for each individual player and an empty matrix. We are creating a thread function to run multiple thread at a time.
- Before defining Build Screen, we're defining display variables like width and heights in a window terminal.
- The Center Message function will decide the choice to select the block in the display grid and colour respectively. We also defined another two functions to print the current player's turn and their current matrix.
- Validate input function checks the input whether it is valid or not. Handle Mouse Event function allows the player to give input through the cursor.
- We define another function AcceptMsgs, in this function we are using encode and decode methods to process data between players and server.
- Start Game function takes the information from AcceptMsgs function and checks the status of pygame and starts the game.

C. Terminal

- By running both server and client codes, there will be a common terminal output where the 3x3 matrix grid of the game can be seen.
- The host organizes the game and runs both client and server code.
- Meanwhile client, the other player only runs the client code and gives the host IP Address to player with the player.

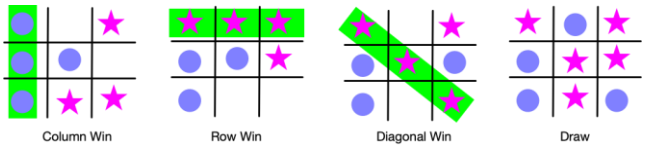

Fig.2. Tic Tac Toe Rules

IV. RESULTS AND DISCUSSIONS

The results are obtained from the game when the two-player had a round of tic-tac-toe as follows:

A. Game is Drawn

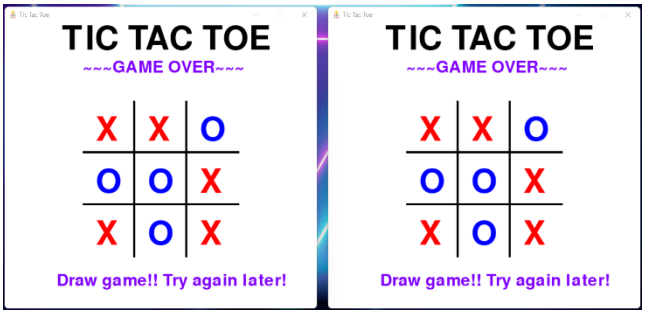

Fig.3. Draw game

This condition happens, when neither of them wins and there is no possibility of chances to either of the chances to win.

B. When Player 1 wins

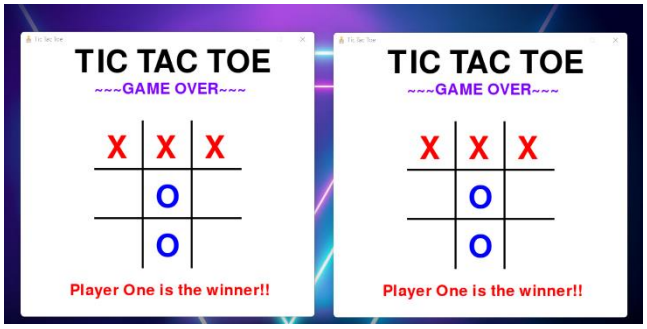

Fig.4. Player One is the winner

This particular case happens when the player one has higher winning possibility as he strikes either the row or the column or the diagonal first. Then he will be the winner.

C. When Player 2 wins

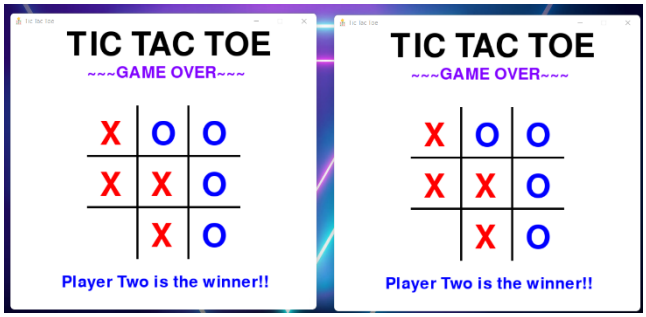

Fig.5. Player Two is the winner

This particular case happens when the player two has higher winning possibility as he strikes either the row or the column or the diagonal first. And he is ahead of player one. Then he will be the winner.

V. CONCLUSION

Our concept of work shows you how to create a turn-based network game in Python by implementing a network version of the Tic-tac-toe classic game. It covers important principles, concepts, and best practices for creating an effective network game. Our game Tic-Tac-Toe uses UDP. We have limited the board size to 3x3 in order to keep things simple. However, the tic-tac-toe game can also be played on larger board sizes. It would be useful to try to generate a generalized rule-based approach like this, creating rules through other methods and then utilizing them for playing games of this class but of higher dimensionality.

The Socket Programming used to communicate between the client and server. It establishes a connection between these players and enables to play the game. By using the host IP Address, we can connect and play the game on two different devices.

VI. REFERENCES

[1] D. B. Kiitsner and B. P. Clemens, "Electronic tictac-toe game having three function control," ed: Google Patents, 2003.

[2] P. Turner, "Combination tic-tac-toe and question and answer game," ed: Google Patents, 1987.

[3] E. Kaplan, "Interactive tic-tac-toe slot machine," ed: Google Patents, 1999.

[4] S. C. Raphael, A. S. Raphael, and R. R. King, "Three-dimensional tic-tac-toe game," ed: Google Patents, 1995.

[5] P. R. Anderson, J. D. Flint, J. J. Giobbi, S. P. Joshi, and E. A. Frohm, "Gaming machine with pattern-driven bonus array," ed: Google Patents, 2005.

[6] T. G. Daly, "Game and gaming machine having tic-tac-toe type feature," ed: Google Patents, 2016.

[7] J. Blake and J. Goodman, "Computer-based learning: games as an instructional strategy," ABNFJOURNAL, vol. 10, pp. 43-45, 1999.

[8] C. Schlieder, P. Kiefer, and S. Matyas, "Geogames: Designing location-based games from classic board games," IEEE Intelligent Systems, vol. 21, pp. 40-46, 2006.

[9] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," IEEE Internet of Things Journal, vol. 5, pp. 450-465, 2018.