Tic-Tac-Toe Implementation using Socket Programming

by Aryan Debray

Submission date: 11-Nov-2023 11:36AM (UTC+0530)
Submission ID: 2224595331
File name: TicTacToe_Report.pdf (192.94K)
Word count: 1335
Character count: 6628

Tic-Tac-Toe Implementation using Socket Programming

Aryan Debray (2105020)

Roudrak Saha (2105058)

1. Abstract

This paper provides the details involved in developing a client server-based multiplayer tic-tac-toe game. The server takes care of the game state, consisting of the board, player turns, and the decision on who wins the game. The clients show the board to the players and send their moves to the server.

Using Python socket module the project is realized. Both the server and client codes are two different independent programs. The first action taken involves power on the server. Then, the clients can connect to the server.

2. Introduction

A common and basic example of a game is tic-tac-toe, which is sometimes referred as noughts and crosses or Xs and Os that involves playing on a three-by-three grid area. The main thing to do in this game is the first one to get three similar marks in a row. These marks make up horizontal, vertical, and diagonal designs within the grid.

This game is played in a three by three square grid and each player alternatively marks the empty cells with their preferred symbols usually represented as X and O. The game will go on until a player successfully makes a run of three
of their marks in a line or the grid is completely filled up without a clear winner thus a draw.

Seemingly simple, but tic-tac-toe has educative value and often used in pedagogy. As such, it is a good means of inculcating basic strategies, encouraging development of critical thinking ability and facilitating pattern recognition especially among beginners in education.

The simple, informative game is still a source of knowledge used in educational circles to develop key cognitive skills and as such it should be studied in research. The research paper talks about the history, play-dynamics, strategy, and educational implications of tic tac toe and how it is still relevant to early childhood education.

The application of a reliable communication between the server and the clients is performed using TCP (Transmission Control Protocol). The server receives the data from the client on an assigned port. On connection of a client, a server creates a new thread to manage their connection. Messages are used to convey the moves by the client to the server. The server validates the moves and advances the game state correspondingly. Then, the server also sends a message to the clients in order to update the game board.

The game goes on till a player wins or it becomes a draw. The server chooses the winner by identifying if there are three in a line (either horizontally, vertically, or diagonally). If there is none, then the outcome is a draw.

3. Working Model

The methodology followed for implementing Tic-Tac-Toe Game with Socket Programming is as follows-

1. Server Setup:
   - The server initializes a socket, binds it to the localhost (127.0.0.1) on port 12345, and begins listening for two player connections.

2. Player Connection:
   - The server awaits two players to connect and accepts their connections, distinguishing them as Player 1 and Player 2 based on their sockets and addresses.

3. Game Initialization:
   - The server initializes the game state using a 'board' list to track moves and the current player, with 'X' starting first.

4. Game State Sending Functions:
   - The server defines two functions, `send_state1()` and `send_state2()`, responsible for transmitting the current game state to Player 1 and Player 2, respectively.

5. Game Logic:
   - The server defines functions to assess the game's outcome, covering win conditions, both row and column wins, diagonal wins, and draws.

6. Main Game Loop:
   - The primary game loop on the server's side continues until there is a winner or a draw.
   - It sends the current game state to Player 1, receives their move, updates the board, and checks for a win or draw.
   - When the game concludes, it transmits the result to both players and terminates the loop.
   - The loop then forwards the game state to Player 2, obtains their move, updates the board, and verifies for a win or draw.
   - If the game ends, it conveys the outcome to both players and exits the loop.

7. Client 1 and Client 2:
   - Both clients establish connections to the server via sockets and define a function to display the game board.
   - They enter a loop to accept and display the current game state.
   - If the game concludes, they display the result and discontinue the loop.
   - Alternatively, they enable the player to input their move and transmit it to the server.

8. Game Conclusion:
   - Once a victor is identified or a draw is achieved, the server and both clients reveal the game's outcome, and the sockets are subsequently closed.

9. Game Flow:
   - Player 1 and Player 2 take turns submitting their moves via their respective clients. The server manages the game state, evaluates wins and draws, and communicates the game's status to the players.

4. Algorithm

4.1. Server Code

i. Initialize the server socket and bind it to a port.
ii. Accept two player connections.
iii. Initialize the game state.
iv. Send the current game state to both players.
v. While the game is not over:
    i. Send the current game state to the current player.
    ii. Receive the current player's move.
    iii. Update the game state with the current player's move.
    iv. Check for a win or draw.
        a) If there is a winner, send a message to both players announcing the winner and the end of the game.
        b) If there is a draw, send a message to both players announcing the draw and the end of the game.
    v. Switch to the next player.
vi. Close the sockets and clean up

4.2. Client Code

i. Initialize the client socket and connect to the server.
ii. Repeatedly:
    i. Receive the current game state from the server.
    ii. Display the game state to the player.
    iii. Allow the player to make a move and send it to the server.
    iv. Check if the game is over. If it is, exit the loop.
iii. Close the client socket.

5. Result

i. The server code waits for the two players to connect. Client 1 is player 1, hence has the gameboard displayed and is asking to make it's move. Client 2 is player 2, hence nothing is displayed and is waiting for client 1 to place turn.

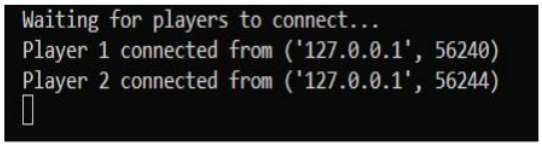

Figure 1. Server

Figure 2. Client 1

ii. The board is updated in the server. Client 1 has played their turn and Client 2 is going to play their turn.

Figure 3. Server

Figure 4. Client 1

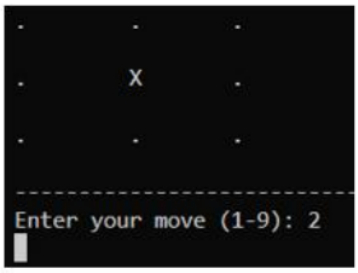

Figure 5. Client 2

iii. Its Client 1's turn to play again after Client 2 played it's turn. Board updated in server as well as in Client 1's screen.

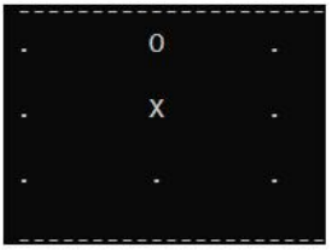

Figure 6. Server

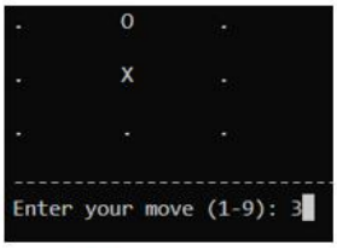

Figure 7. Client 1

iv. The game is played out. Client 1 wins. Client 2 loses. Winner is displayed in server.

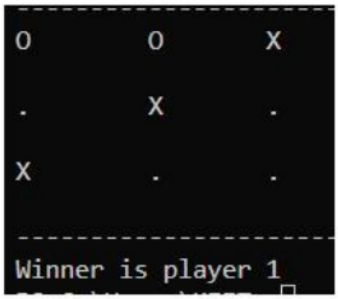

Figure 8. Server

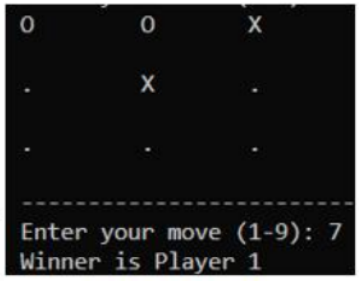

Figure 9. Client 1

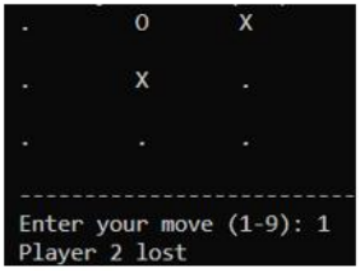

Figure 10. Client 2

6. Conclusion

In conclusion, a Tic-Tac-Toe game has been successfully developed using socket programming in Python. This game enables two participants to connect to a central server and engage in gameplay over a network. The server's role is to maintain the game state, which is communicated to both players after each move. Subsequently, the players are responsible for making their moves and transmitting them back to the server. The server's functionality also encompasses verifying the game outcome, whether it is a win or a draw, and updating the game state accordingly.

This implementation demonstrates efficiency as the data exchanged between server and client sockets is minimal, primarily involving the game state and the player's chosen move. This design makes the game suitable for network play even in scenarios with limited bandwidth.

The overall assessment of this Tic-Tac-Toe implementation using socket programming in Python is positive. The game offers ease of play and comprehension, coupled with reasonable efficiency. It is expected that this game will provide an enjoyable experience for players.

7. References

1. https://www.geeksforgeeks.org/socket-programming-python/

2. https://levelup.gitconnected.com/program-a-networked-tic-tac-toe-game-in-python-30f8826e591d?gi=471aac2f72af

3. https://www.freecodecamp.org/news/socket-programming-in-python/

4. https://youtu.be/3QiPPX-KeSc?feature=shared

Tic-Tac-Toe Implementation using Socket Programming

ORIGINALITY REPORT

2%	0%	1%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Francesco Di Giacomo, Mohamed Abbadi, Agostino Cortesi, Pieter Spronck, Giulia Costantini, Giuseppe Maggiore. "High performance encapsulation and networking in Casanova 2", Entertainment Computing, 2017 Publication	1%
2	Submitted to University of Wales Swansea Student Paper	1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		