Report

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

Title: LAN Application Multiplayer Game

Mini Project Report

CSE3162 COMPUTER NETWORKS LAB

Student: Anurag Nayak (210905016)

Student: Moksha Kothari (210905017)

Student: Adruti Onam (210905190)

Course: CSE3162

10-Nov-2023

Abstract

The ”LAN Applications Game with Multiplayer Game Selection” project represents a foray into the integration of computer networking and game development, aiming to create a LAN-based gaming platform. This platform enables two players to engage in classic games like Connect 4, Tic Tac Toe, and Hangman. Central to this project is the implementation of a client-server architecture that allows real-time interaction and gameplay synchronization. Each player, acting as a client, connects to a central server that coordinates game sessions and maintains consistent game states across clients.

A notable aspect of this project is the development of a secure authentication system, ensuring that player access is verified before game sessions begin. This adds an essential layer of security and integrity to the gaming experience. The project's dual focus on technical network implementation and engaging game design allows players to enhance their strategic thinking and problem-solving skills in a competitive yet secure environment.

This abstract encapsulates the project's overarching goals, its technical implementation, and the expected outcomes. The report that follows delves deeper into the specifics of the network architecture, game logic, and user interface design, providing comprehensive insights into the challenges and successes encountered during development. Ultimately, the project aims not just to create a functional gaming application but also to explore the potential and implications of LAN-based systems in the realm of interactive multiplayer gaming.

Contents

1. Introduction

Beginning in the late 1990s and continuing today, a wide range of interesting devices are being connected to the Internet, leveraging their ability to send and receive digital data. Given the Internet's ubiquity, its well-defined (standardized) protocols, and the availability of Internet-ready commodity hardware, it's natural to use Internet technology to network these devices together and to Internet-connected servers. Many of these devices are based in the home—video game consoles (e.g., Microsoft's Xbox), Internet-ready televisions, digital picture frames that download and display digital pictures, washing machines, refrigerators, and even a toaster that downloads meteorological information and burns an image of the day's fore- cast (e.g., mixed clouds and sun) on your morning toast [BBC 2001].[1]

When writing programs that communicate across a computer network, one must first invent a protocol, an agreement on how those programs will communicate. Before delving into the design details of a protocol, high-level decisions must be made about which program is expected to initiate communication and when responses are expected. For example, a Web server is typically thought of as a long-running program (or daemon) that sends network messages only in response to requests coming in from the network. The other side of the protocol is a Web client, such as a browser, which always initiates communication with the server. This organization into client and server is used by most network-aware applications.[2]

This assignment centers on the design and development of a LAN-based gaming application, an endeavor that marries the intricacies of computer networking with the engaging world of multiplayer gaming. I chose this assignment due to its multifaceted nature, encompassing challenges in game development, network communication, and user interface design. The project's primary aim is to implement three classic games - Connect 4, Tic Tac Toe, and Hangman - within a networked environment, allowing two players to interact in real time via a central server.

The motivation behind selecting this project lies in addressing several analytical problems and questions. Firstly, there is the challenge of developing robust game logic that ensures fair play and seamless gameplay. Secondly, the project demands efficient client-server communication over a Local Area Network (LAN), a network limited to a small geographic area, such as a building or campus. The utilization of LAN is crucial for its capability to facilitate fast and reliable data transmission, a key component for real-time multiplayer gaming experiences. Lastly, crafting an intuitive and user-friendly interface presents its own set of challenges, requiring a design that is both appealing and easy to navigate for players.

In this report, we will delve into the key themes of game development, LAN networking, and user interface design. Each of these aspects will be explored in detail, providing the necessary background and explaining the core concepts involved.

2. Methodology

2.1 Central Server Setup

- Establish and manage client connections and games.
- Maintain a list of active clients and their statuses.

2.2 Client Connections Initialization

- Develop client-side application for server connection.
- Handle connection requests, disconnections, and reconnections.

2.3 Authentication Implementation

- Develop a login system.
- Implement security measures using password and maintain a user database.

2.4 Game Logic Implementation

- Develop rules, win conditions, and progression for Connect 4, Tic Tac Toe, and Hangman.
- Synchronize game state between clients and server.

2.5 Client-Server Communication

- Implement protocols for data exchange between clients and server.
- Ensure real-time synchronization of game state.

2.6 System Testing

- Conduct thorough testing to identify and fix bugs or vulnerabilities.
- Implement debugging tools and gather user feedback.

2.7 Documentation and User Instructions

- Document the code, prepare user documentation, and develop a user guide.
- Maintain version control for updates.

2.8 Player Choice for Gameplay

- Allow players to choose opponents.
- Ensure game starts only when both players are ready.

3. Implementation

3.1 Server and Client Initialization

- Set up the server to handle multiple client connections.
- Initialize client connections and manage disconnections.

3.2 Game Logic Implementation

- Implement Tic Tac Toe game logic (Moksha).
- Implement Connect 4 game logic (Adruti).
- Implement Hangman game logic (Anurag).

3.3 Game State and Conditions

- Handle game state and win/lose conditions for Tic Tac Toe (Moksha).
- Handle game state and win/lose conditions for Connect 4 (Adruti).
- Handle game state and win/lose conditions for Hangman (Anurag).

3.4 Client-Server Communication)

- Handle the authentication between clients and server for real-time gameplay (Adruti).
- Handle online Status of players (Anurag).

3.5 Testing and Documentation

- Test the system for any bugs or glitches and fix them (Moksha).
- Document the code and write user instructions (Adruti).

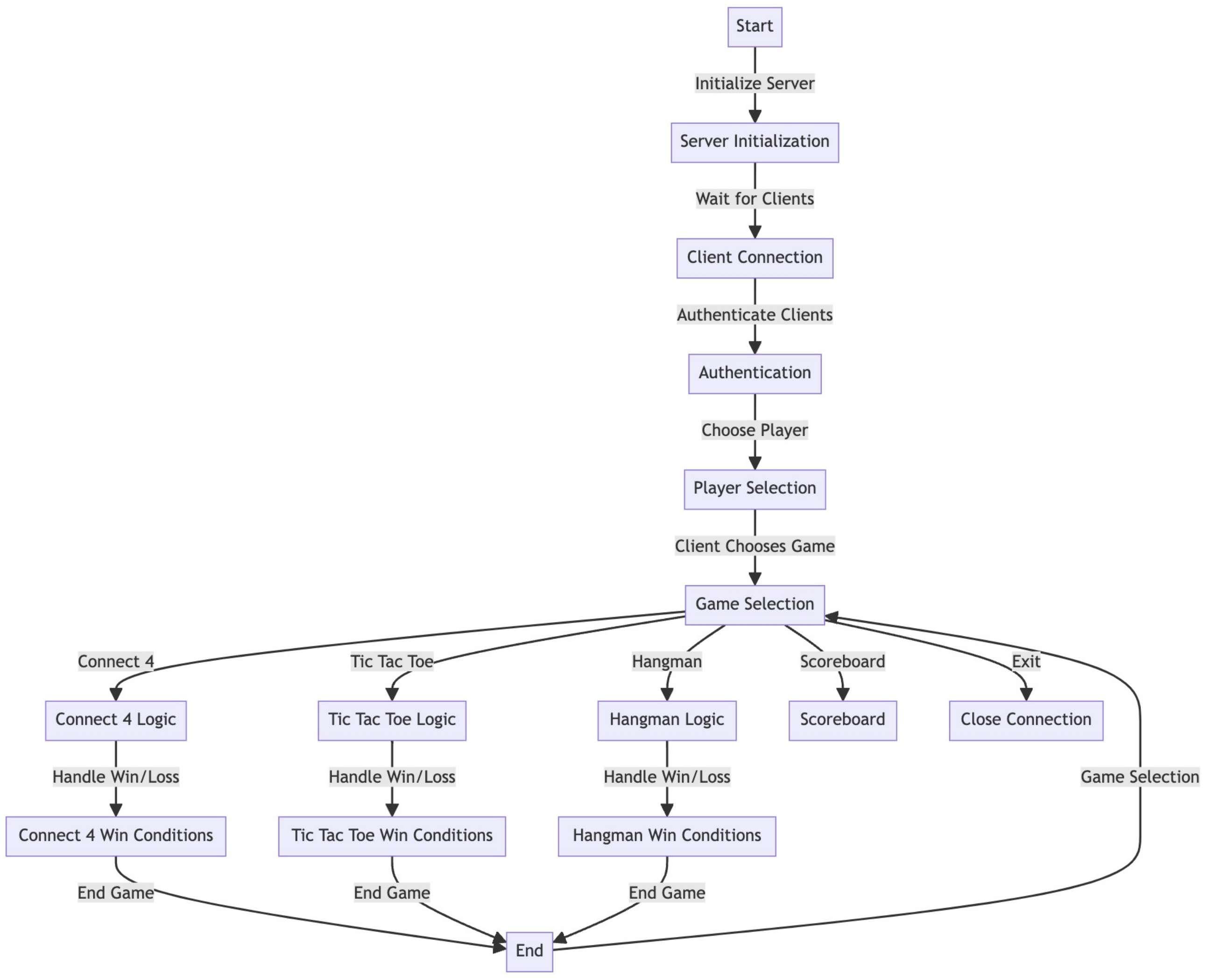
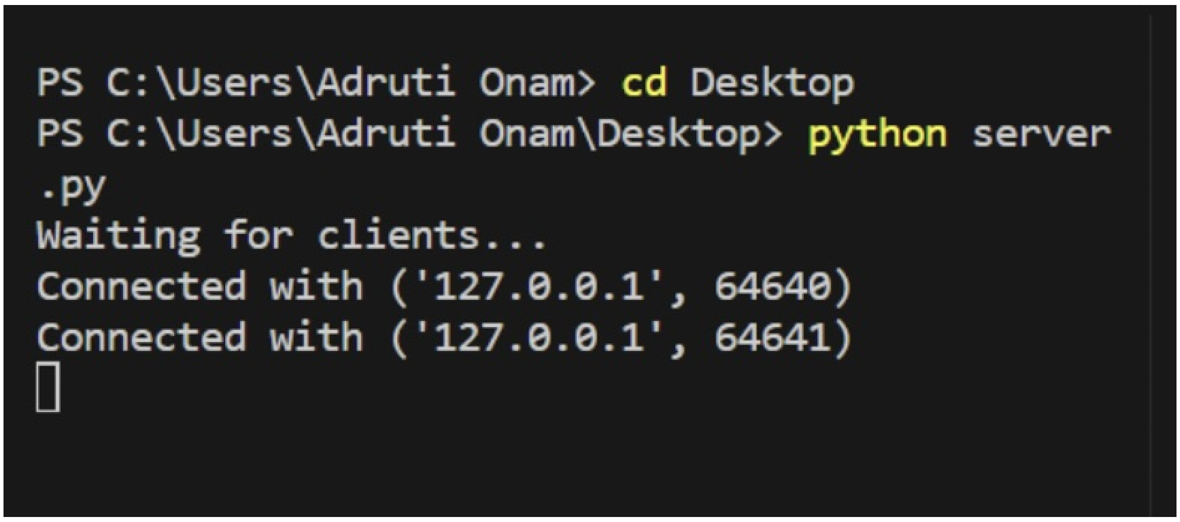


Figure 1: Implementation of Client-Server Code

4. Results and Analysis/Snapshots

4.1 Snapshots

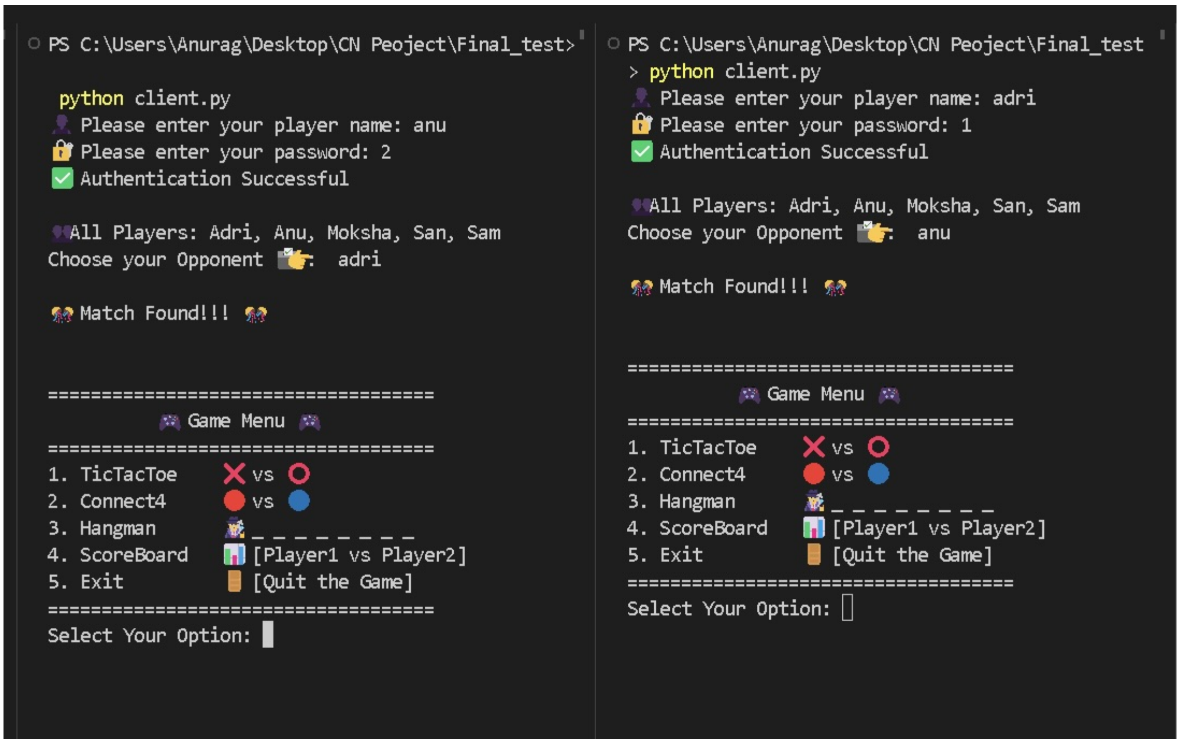
- Start of client, server programs where the client joins.



```
PS C:\Users\Adruti Onam> cd Desktop
PS C:\Users\Adruti Onam\Desktop> python server
.py
Waiting for clients...
Connected with ('127.0.0.1', 64640)
Connected with ('127.0.0.1', 64641)
```

Figure 2: Client-Server Initialization

- Authentication of client and choice of players along with game menu/exit.

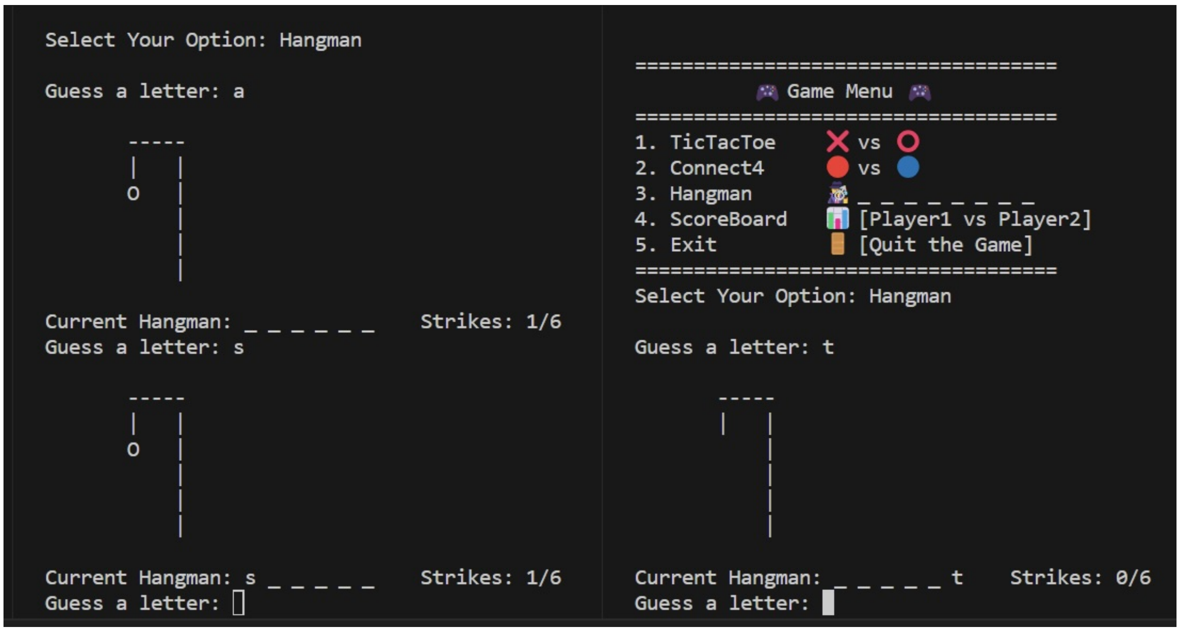


```
PS C:\Users\Anurag\Desktop\CN Peoject\Final_test>

 python client.py
👤 Please enter your player name: anu
🔐 Please enter your password: 2
✅ Authentication Successful

👥All Players: Adri, Anu, Moksha, San, Sam
Choose your Opponent 👉:  adri

🎊 Match Found!!! 🎊


====================================
          🎮 Game Menu 🎮
====================================
1. TicTacToe      ❌ vs ⭕
2. Connect4       🔴 vs 🔵
3. Hangman        🧑 _ _ _ _ _ _ _ _
4. ScoreBoard     📊 [Player1 vs Player2]
5. Exit           🚪 [Quit the Game]
====================================
Select Your Option:
```

```
PS C:\Users\Anurag\Desktop\CN Peoject\Final_test
> python client.py
👤 Please enter your player name: adri
🔐 Please enter your password: 1
✅ Authentication Successful

👥All Players: Adri, Anu, Moksha, San, Sam
Choose your Opponent 👉:  anu

🎊 Match Found!!! 🎊


====================================
          🎮 Game Menu 🎮
====================================
1. TicTacToe      ❌ vs ⭕
2. Connect4       🔴 vs 🔵
3. Hangman        🧑 _ _ _ _ _ _ _ _
4. ScoreBoard     📊 [Player1 vs Player2]
5. Exit           🚪 [Quit the Game]
====================================
Select Your Option:
```

Figure 3: Authentication, Choice of Players, Game Menu

- Gameplay and Win/Lose statement to keep track of who won in the game.

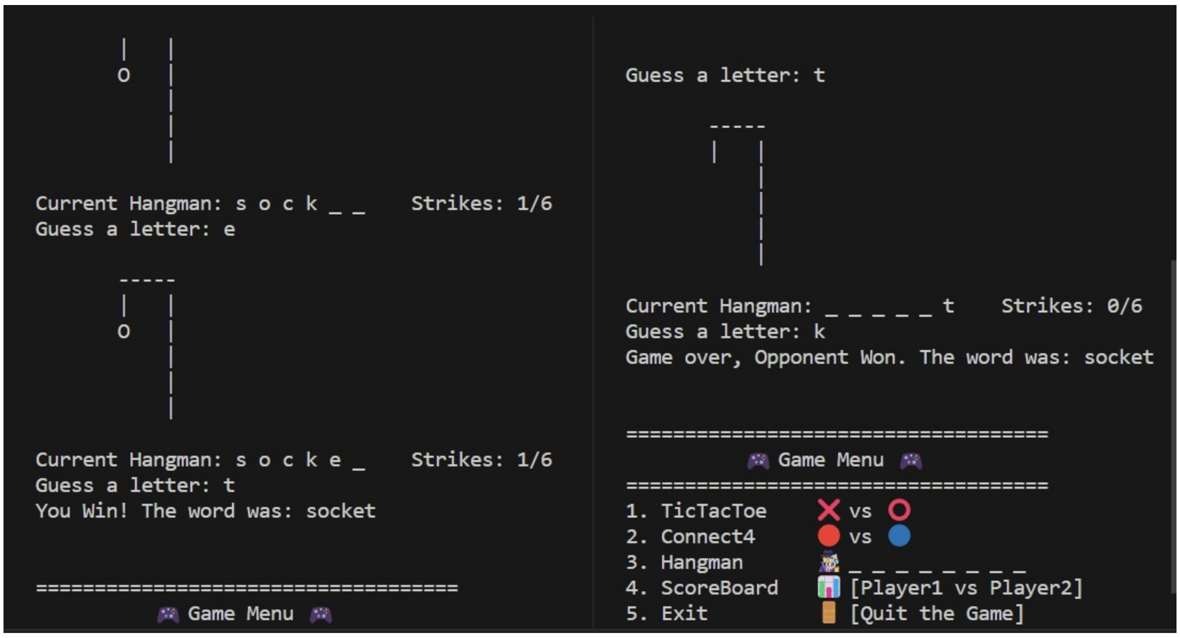


```
Select Your Option: Hangman

Guess a letter: a

       -----
       |   |
       O   |
           |
           |
           |

Current Hangman: _ _ _ _ _ _     Strikes: 1/6
Guess a letter: s

       -----
       |   |
       O   |
           |
           |
           |

Current Hangman: s _ _ _ _ _     Strikes: 1/6
Guess a letter:
```

```
====================================
          🎮 Game Menu 🎮
====================================
1. TicTacToe       ❌ vs ⭕
2. Connect4        🔴 vs 🔵
3. Hangman         🧙 _ _ _ _ _ _ _ _
4. ScoreBoard      📊 [Player1 vs Player2]
5. Exit            🚪 [Quit the Game]
====================================
Select Your Option: Hangman

Guess a letter: t

       -----
       |   |
           |
           |
           |
           |

Current Hangman: _ _ _ _ _ t     Strikes: 0/6
Guess a letter:
```

Figure 4: Gameplay

- (Here game: Hangman)

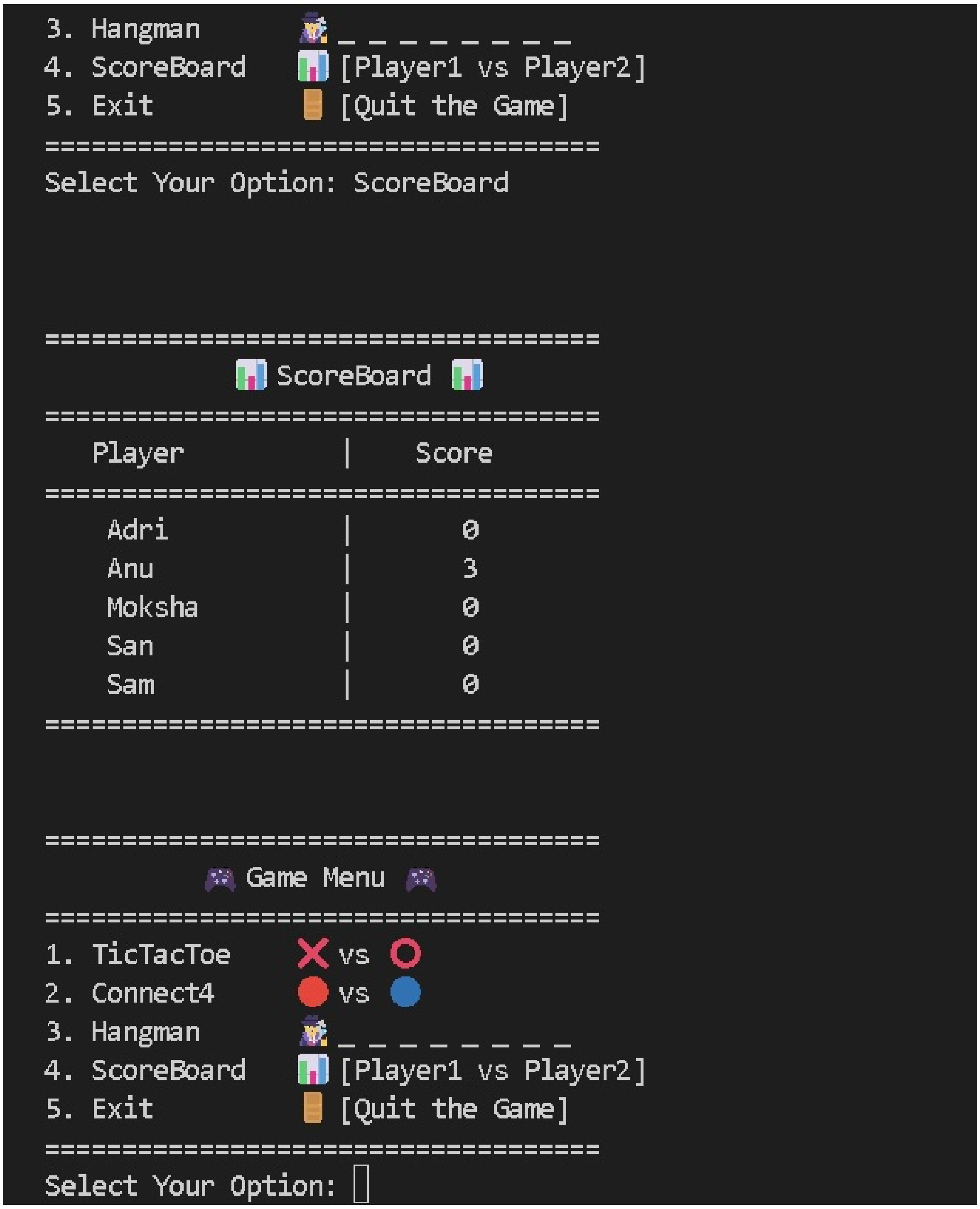


```
       |   |
       O   |
           |
           |
           |

Current Hangman: s o c k _ _     Strikes: 1/6
Guess a letter: e

       -----
       |   |
       O   |
           |
           |
           |

Current Hangman: s o c k e _     Strikes: 1/6
Guess a letter: t
You Win! The word was: socket


====================================
          🎮 Game Menu 🎮
```

```
Guess a letter: t

       -----
       |   |
           |
           |
           |
           |

Current Hangman: _ _ _ _ _ t     Strikes: 0/6
Guess a letter: k
Game over, Opponent Won. The word was: socket


====================================
          🎮 Game Menu 🎮
====================================
1. TicTacToe       ❌ vs ⭕
2. Connect4        🔴 vs 🔵
3. Hangman         🧙 _ _ _ _ _ _ _ _
4. ScoreBoard      📊 [Player1 vs Player2]
5. Exit            🚪 [Quit the Game]
```

Figure 5: Hangman Game

- Scoreboard to keep track of Scores of the two Players.



```
3. Hangman        🕵 _ _ _ _ _ _ _ _
4. ScoreBoard     📊 [Player1 vs Player2]
5. Exit           🚪 [Quit the Game]
====================================
Select Your Option: ScoreBoard



====================================
            📊 ScoreBoard 📊
====================================
   Player          |     Score
====================================
    Adri           |       0
    Anu            |       3
    Moksha         |       0
    San            |       0
    Sam            |       0
====================================


====================================
          🎮 Game Menu 🎮
====================================
1. TicTacToe      ❌ vs ⭕
2. Connect4       🔴 vs 🔵
3. Hangman        🕵 _ _ _ _ _ _ _ _
4. ScoreBoard     📊 [Player1 vs Player2]
5. Exit           🚪 [Quit the Game]
====================================
Select Your Option: ▯
```

Figure 6: Scoreboard

4.2 Outcomes

- Successfully implemented three games: Connect 4, Tic Tac Toe, and Hangman, each with their unique game logic and user interfaces.
- Established effective communication protocols between the client and server, en-

suring smooth data transmission and game synchronization.

4.3 Network Performance and Data Transmission Rates

- The LAN-based system facilitated low-latency communication, resulting in a responsive multiplayer gaming experience.
- Data transmission rates were found to be consistent and reliable, with minimal packet loss or delay.

4.4 Challenges

- Synchronization of game state across clients.
- Managing multiplayer interactions and player connections.
- Ensuring continuous gameplay and stable socket connections.
- Implementation of game logic for Hangman, Connect 4, and Tic Tac Toe.
- Rigorous testing and debugging to identify and fix system issues.
- Complexity in implementing scoring systems and leaderboards for multiplayer games.

4.5 Bugs

- Connection issues, including socket connection failures.
- Tracking and updating scores and leaderboards for various games.
- Handling authentication errors, such as incorrect usernames or passwords.
- Addressing input validation issues that lead to invalid moves or data format errors.
- Resolving data interference issues when handling multiple clients concurrently.

5. Discussion

5.1 Interpretation of Results and Implications

- Demonstrated effective implementation of game logic and client-server communication for Connect 4, Tic Tac Toe, and Hangman.
- Highlighted the positive implications of applying theoretical knowledge to create a real-world multiplayer gaming experience over LAN.

5.2 Addressing Challenges

- Overcame network synchronization challenges through the implementation of game state management protocols.
- Developed a robust system to manage multiplayer interactions and player connections.
- Addressed the complexity of scoring and leaderboards by developing dedicated modules for score tracking and management.

5.3 Resolution of Bugs

- Ensured stable connections through implemented error handling mechanisms.
- Conducted extensive testing for accurate score tracking.
- Enhanced security by implementing validation checks.
- Achieved smooth gameplay across multiple clients.
- Addressed data interference to maintain game consistency.

6. Conclusion

Most action games played on the net today are modified client / server games. Games such as Half-Life, including its mods such as Counter-Strike and Team Fortress Classic,operate on such a system, as do games based on the Quake3 engine and the Unreal Tournament engine. In these games, there is a single, authoritative server that is responsible for running the main game logic. To this are connected one or more "dumb" clients. These clients, initially, were nothing more than a way for the user input to be sampled and forwarded to the server for execution. The server would execute the input commands, move around other objects, and then send back to the client a list of objects to render. Of course, the real world system has more components to it, but the simplified breakdown is useful for thinking about prediction and lag compensation. [3]

In conclusion, the LAN Application Multiplayer Game project successfully fulfilled its goal of providing a seamless and engaging multiplayer gaming experience for players over a Local Area Network.The implementation of classic games such as

Connect 4, Tic Tac Toe, and Hangman, along with the incorporation of client-server architecture, has resulted in a system that is not only interactive but also enhances players' strategic and problem-solving skills. The collaborative efforts of the team members in handling different aspects of the project, from server setup and client communication to game logic implementation and testing,have been integral to the success of this project.

7. Future Scope

The project has laid a solid foundation for further improvements and additions. In the future, the following enhancements can be considered:

7.1 Expansion of Game Library

- Incorporate more games into the application, providing a wider range of options for players to choose from.

7.2 Multiplayer Support

- Extend the system to support more than two players in a single game, allowing for team-based gameplay.

7.3 Enhanced User Interface

- Improve the user interface to make the application more visually appealing and user-friendly.

7.4 In-Game Chat Feature

- Incorporate a chat feature that allows players to communicate with each other during the game.

8. References

Bibliography

[1] J.F. Kurose and K.W. Ross, *Computer Networking: A Top-Down Approach.*

[2] W. Richard Stevens, *UNIX Network Programming*, Addison-Wesley.

[3] Yahn W. Bernier, *Latency Compensating Methods in Client/Server In-game Protocol Design and Optimization.*

Bibliography