

# **PROJECT REPORT**

# **COMPUTER NETWORKS**

***TITLE* : *Rock Paper Scissors Game Implementation Using Socket Programming***

***LECTURER* :** *Sir Muhammad Ali Naseer Shah*

***Project Members* :** *Shahryar(k21-3411)*

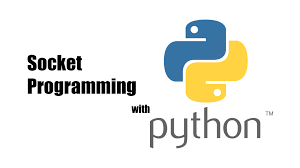
*Sajjad Ali(K21-4560)*

*Muhammad Yameen(k21-3452)*

***SECTION:*** *BS-CS-9A*

***Course Code:*** *CS-3001*

***TOOLS:***



****

***Introduction:***

*Rock Paper Scissors is a classic game that has been enjoyed by people of all ages for generations. In this project, we have implemented the game of Rock Paper Scissors using TCP sockets, allowing for multiple players to connect to a single server and compete against each other in real-time.*

*The project consists of three main components: the server code, and two client codes that represent the two players. The server code runs on a central machine and is responsible for coordinating the game and communicating with the clients. The clients, on the other hand, are the players themselves and are responsible for making their moves and receiving updates from the server.*

*To begin the game, both clients connect to the server using TCP sockets. Once connected, the server waits for both clients to send their moves (rock, paper, or scissors) and then compares them to determine the winner. The server then sends the results to both clients and the game continues until one player reaches a predetermined number of wins.*

*Overall, this project provides an interactive and engaging way to learn about TCP sockets and network programming, while all having fun with a classic game.*

***Functional features:***

1. *Game play*
2. *Score tracking*

***Proposed Work:***

*The project is about creating a rock paper scissors game over a TCP client server environment. There is one server and two clients, the two clients connect to the server and the game begins. There are three rounds played by the players and after this tournament the results are announced (whether either of the players has won or there is a tie). The GUI has been implemented using the tkinter library.*

***Methodology:***

1. *Purpose: The game is designed for educational purposes to shoe hoe TCP client server connection can be used for different purposes.*
2. *Requirements gathering: The requirements we got were mostly the simple rules of the game and some were given by our instructor (that the game should contain multiple rounds and in the GUI there should be buttons for the options (rock, paper or scissors)*
3. *Design: By looking at the requirements, we designed the structure of our code, that there should be one server and two clients (as two players are there for this game).*
4. *Implementation: The implementation was done in python language on VS code, it took around one week for implementing this project. The server code is the main code which performs the actual game logic and allows the players to come and connect to the game. The clients join the network over TCP connection and play the game.*
5. *Testing: once the code was completed we both tested by running and giving different inputs and checking whether correct results are presented.*

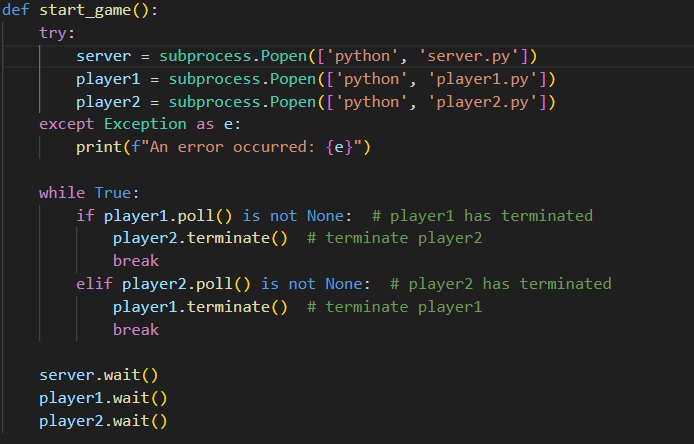
***Algorithms:***

1. *Socket Programming*

***Tools and technologies:***

* *Language: Python 3*
* *Libraries: socket, threading and tkinter*
* *OS: windows*

***Code Snippets:***

******

***A computer screen shot of a program code

Description automatically generated***

***A screen shot of a computer program

Description automatically generated***

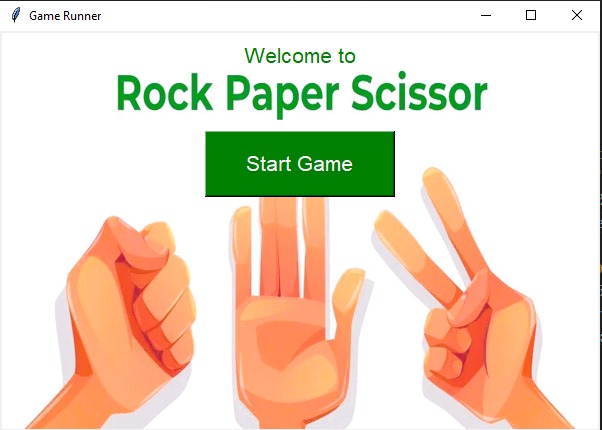
***A screen shot of a computer program

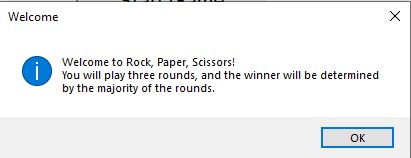
Description automatically generated***

***A computer screen with text

Description automatically generated***

***Game Snippets:***

****

****

****

****

****

***Important Functions Description:***

***1.*** *Start\_game (): This function is responsible for starting the Rock Paper Scissors game by launching the server and two player processes using subprocess.It waits for the player processes to terminate and then waits for the server process to finish.*

***2.*** *Create\_button (window, move\_name, image\_path): This function creates a button widget with an image of rock, paper, or scissors, depending on the move\_name parameter.It loads the image from the specified image\_path and binds it to a command that sends the corresponding move to the server when clicked.*

***3.*** *Send\_move (move): This function sends the selected move (rock, paper, or scissors) to the server through the established socket connection. It then receives the result of the round from the server and updates the result\_label with the round number and winner.*

***4.*** *Receive\_end\_message (): This function receives the end message from the server, indicating the final result of the game (e.g., whether it's a tie or which player wins). It updates the result\_label with the end message.*

*These functions collectively enable the communication between the client (player) and the server in the Rock Paper Scissors game, allowing players to make their moves and receive feedback on the game outcomes.*

***References:***

* *Python Socket Programming Tutorial*: <https://realpython.com>
* *Rock paper scissors game logic*: https://realpython.com/python-rock-paper-scissors/
* *TCP-Protocol*: <https://www.geeksforgeeks.org/tcp-server-client-implementation-in-python/>

***Conclusion:***

*In conclusion, the implementation of the Rock Paper Scissors game using socket programming in Python has been successfully achieved. This project aimed to demonstrate the practical application of socket programming concepts, particularly in the context of creating a multiplayer game environment over a network.*

*Moreover, the project showcased the effective use of Python's socket module to establish network communication between the server and clients. By leveraging TCP/IP sockets, reliable and bidirectional data exchange was facilitated, enabling seamless gameplay experiences for connected players.*

*Additionally, the project emphasized the importance of user experience and interface design. Simple yet intuitive command-line interfaces were developed for both the server and clients, enabling ease of use and clarity in navigating the game environment.*

*Overall, this project serves as a practical demonstration of how socket programming can be utilized to create engaging multiplayer games over a network. By combining fundamental networking concepts with Python's versatile programming capabilities, the Rock Paper Scissors game was successfully implemented, offering an enjoyable and interactive gaming experience for players across different networked devices.*