ONLINE

MULTIPLAYER GAME

BY SOMANSHU,YASH AND SUPRATIK

CONTENT

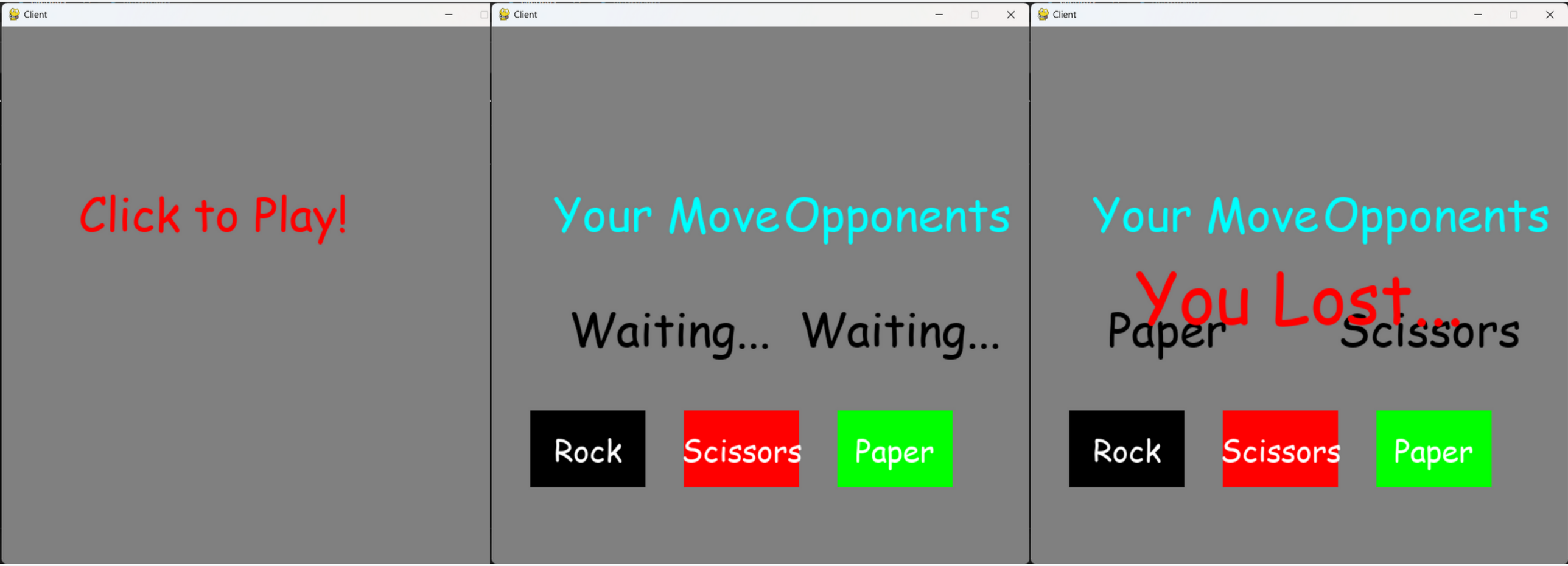
INTRODUCTION

Socket programming in Python involves creating a communication channel between two endpoints over a network. The endpoints can be a server and a client or two clients. Python provides a built-in module called "socket" that allows us to create and use sockets for network communication.

The socket module provides the socket class, which has methods for creating, binding, listening, accepting, connecting, and sending data through sockets. Here are the basic steps involved in socket programming in Python:

1. Creating a socket object: You can create a socket object using the socket() method provided by the socket module. The method takes two arguments: the first one is the address family, and the second one is the socket type.
2. Binding the socket: Once you have created a socket object, you can bind it to a specific IP address and port number using the bind() method. This step is only required for the server-side socket.
3. Listening: After binding the socket, you need to listen to incoming client connections using the listen() method.
4. Accepting connections: When a client tries to connect to the server, the server should accept the connection using the accept() method. This method returns a new socket object representing the client's connection.
5. Sending and receiving data: After establishing the connection, both the client and server can send and receive data through their respective socket objects using the send() and recv() methods.
6. Closing the connection: When the communication is complete, both the client and server should close their socket objects using the close() method.

SCREENSHOTS



CODE

There are 4 code files

1.network.py

2.server.py

3.game.py

4.client.py

REFERENCES

1.stackoverflow.com
2.youtube.com
3.geeksforgeeks.com
4.github.com