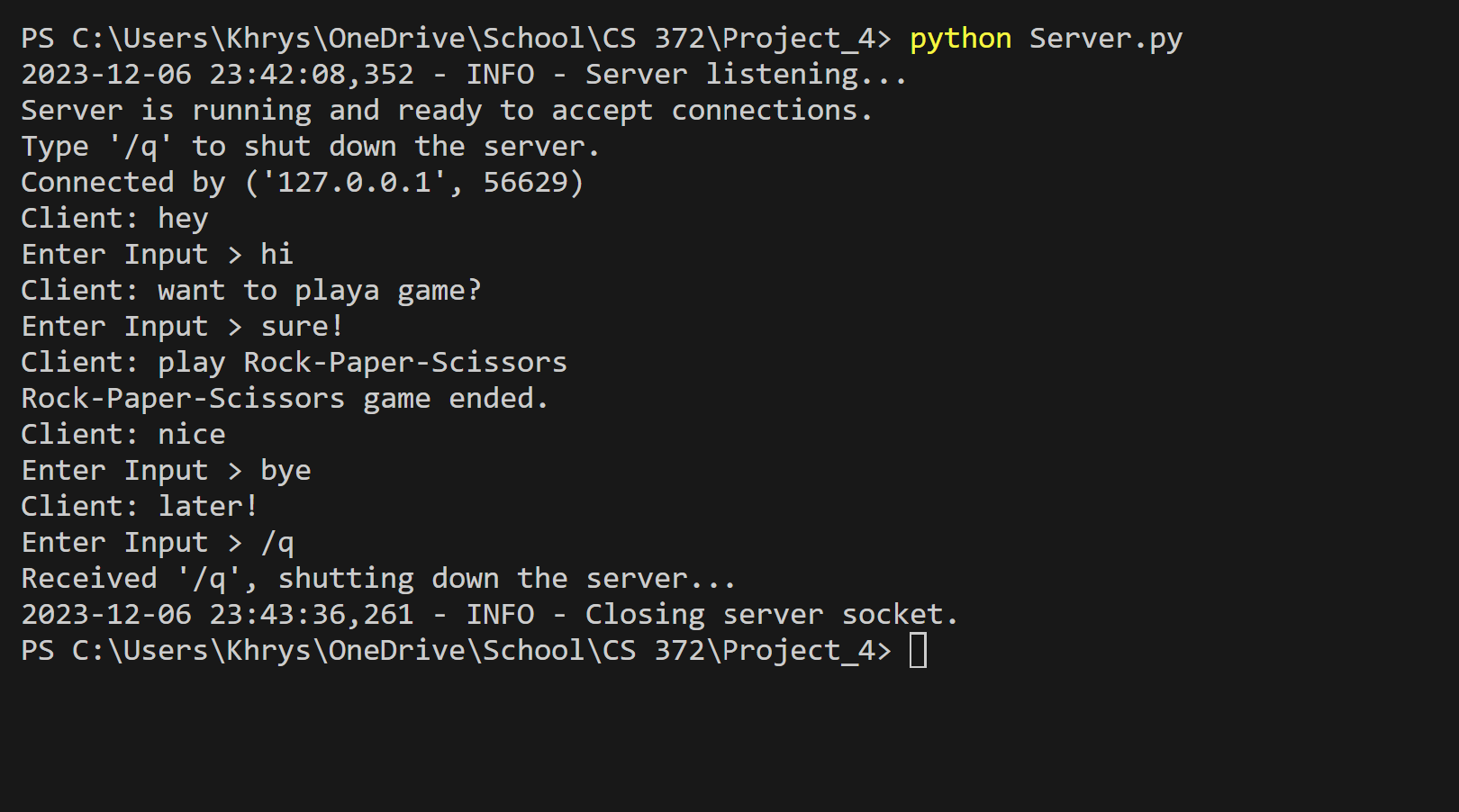
Khrystian Clark

12/06/2023

* Introduction
  + Purpose of the Assignment: Create a Server-Client Chat that can be used a game (rock-paper-scissors), utilizing socket logic properly as well as socket communication.
  + Bulk of the logic: The logic is shared pretty equally between the Server and Client code. The game logic as well is shared pretty equally, with the Client initializing it and the Server returning the final outcome.
  + Development Environment:
    1. Python3
    2. Written in python
    3. IDE’s used to create/test: VS Code, GitHub CodeSpace, PowerShell
* Instructions
  + Note: These are python3 instructions
    1. Save the Server.py and Client.py files locally
    2. In the terminal, map to the location of the newly saved files
    3. In a terminal, run the program “python3 Server.py” (this has to be the first one to run in order for this program to work)
    4. In a separate terminal, run the program “python3 Client.py”
    5. Review the output in the terminal response window.
    6. Have a conversation between the two, in whichever terminal window is prompted for input. (If this is done out of order, it will input the last thing entered when not prompted.)
    7. To exit, type “/q” in whichever response window is prompted.
  + To run the Rock Paper Scissors game:
    1. Follow steps 1-6.
    2. In the terminal prompted, type “play rock-paper-scissors”
    3. This will initialize a game of rock paper scissors, where the Server (computer) plays against the Client (user).
    4. In the client window, enter “rock”, “paper” or “scissors” (just one, and it has to be one of those three or else the program will re-prompt you for input)
    5. Review the results in the Client window to see if you have won.
    6. Repeat steps 2-5, until you feel like stopping and either returning to the conversation (just continue with conversational responses) or quitting (enter “/q”)
* comments / questions
  + I included logging in both client and server in hopes to ensure clarity and help with any debugging along the way. I found this idea in my many google searches on creating client server chatbots, and the logic resource is in the code.
  + All sources are cited in the code
  + Almost every line is commented for future use as a resource in school or work.
* Screenshots
  + Server instantiating the chat and ending it
  + Server instantiating chat and client ending it
  + Random conversation
  + A few instances of rock-paper scissors game

A computer screen shot of a program

Description automatically generated

Server.py

Client.py