Rock, Paper, Scissors(, Magic)

[NAME REDACTED]

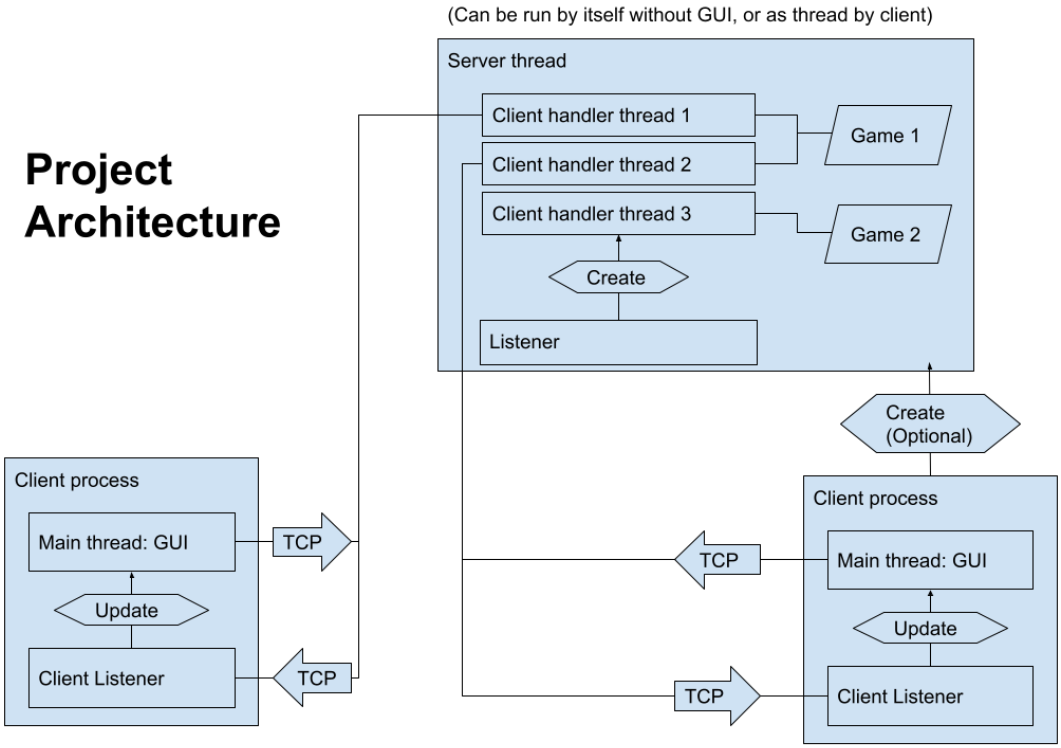
This is an overview of *Rock, Paper, Scissors(, Magic)*, written in Python for the CSCI 4300 Operating Systems final project.

The rules are simple:

- Two players choose ROCK, PAPER, or SCISSORS at random.
- ROCK beats SCISSORS.
- SCISSORS cuts PAPER.
- PAPER covers ROCK.
- Best 2 out of 3.

However, thanks to Revolutionary New Technology™, *Rock, Paper, Scissors(, Magic)* expands upon the classic *Rock, Paper, Scissors* formula by introducing an all-new fourth option: **MAGIC..?**. Its rules are as follows:

- MAGIC..? wins against nothing.
- MAGIC..? loses against everything.



The design of the project is separated into two sections: the server, and the client. Each can run independently of the other, but both can be run at once in the same process.

- The Server portion of the program can be run without any UI or client code using `python3 RPS.py server`, allowing it to act as a dedicated server on command-line-only machines.
- Likewise, the Client portion of the program runs independently of the server portion and will run by default.
- The client has a “Start Local Server” button that can be used to create a Server thread within the same process.
- Each client runs on a separate process.

The server manages the game. It contains the game logic and evaluates winners, and its say is the end-all-be-all. Moves must be valid. There is no client trust.

In fact, all communications between the client and server are simply the clients submitting an ID for their move (from 1 to 4), and the server telling each client what to display on their screen- what the flavor text should say, if the player can make a move right now, whether to show the Rematch button, and so on.

Each message is headed by “RPS” to divide them in the TCP stream.

The entire game state, as seen by the client, is contained in only two bytes (not counting strings):

0001	0011	1	0	00	10	01
Opponent’s move (Rock)	My move (Scissors)	Hide the moves menu?	Show rematch button?	Round 3: No point	Round 2: Opponent’s point	Round 1: My point

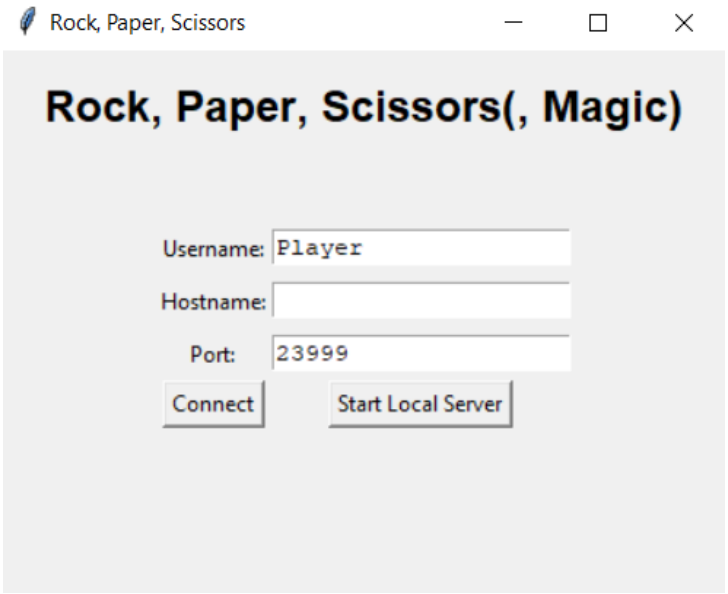
Each client has its own thread, and games are managed by a list of dictionaries. Each one tracks the state of the game, and who is participating in it. Each game state also has its own lock, so that two client threads cannot try to read or modify the same game at the same time.

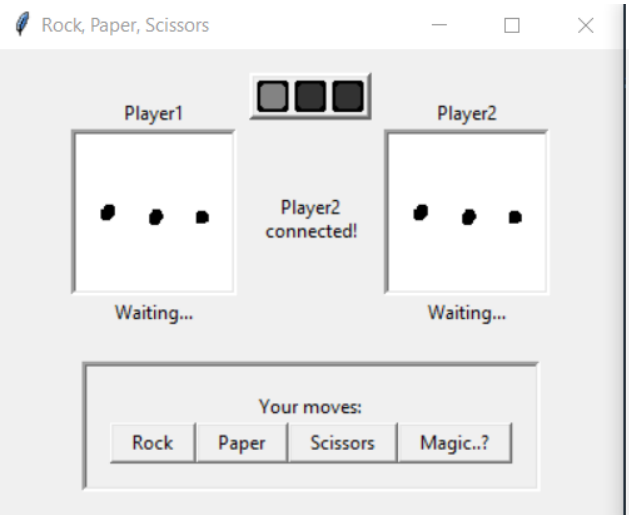
Functions provided include:

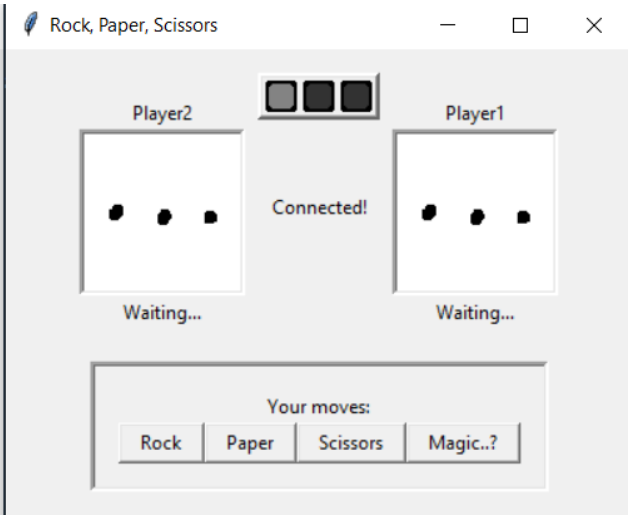
- packGamestate: This function is used by the server to condense the state of the game into two bytes to send over the network.
- parseGamestate: This function is used by the client to read the two-byte gamestate and update its UI accordingly.
- clSendMove: This function is used by the client to tell the server what move it has picked. Sending a move ID of 0 votes for a rematch.
- ListenServer: This function **is** the server portion of the program. If run as a dedicated server, the program will immediately run this function and ignore the remainder of the script; if used by the client, it will be instantiated as a thread.
- handleClient: This function is a thread spawned by the server for each client. It contains all of the game logic, and runs it when the server receives a message from one of the clients.

This program was made with Python ~3.9 and uses only standard library modules.

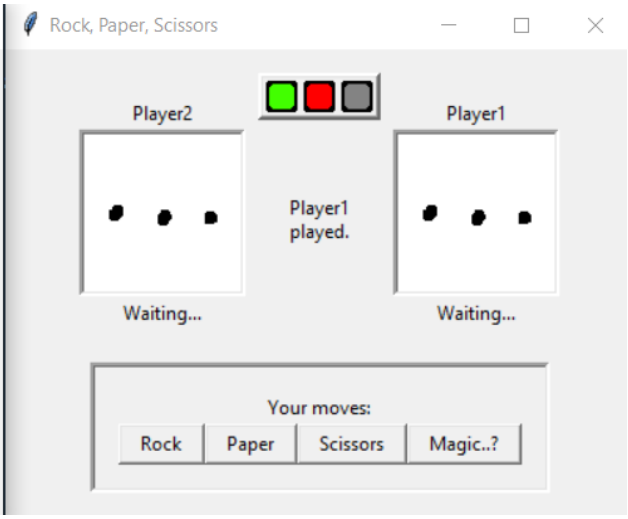
Here are both the GUI design and screenshots of the program running, since I never actually made any tentative mockups of the UI:

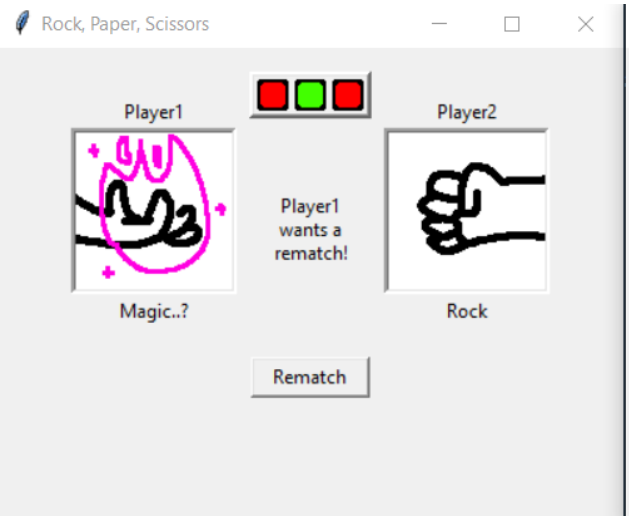


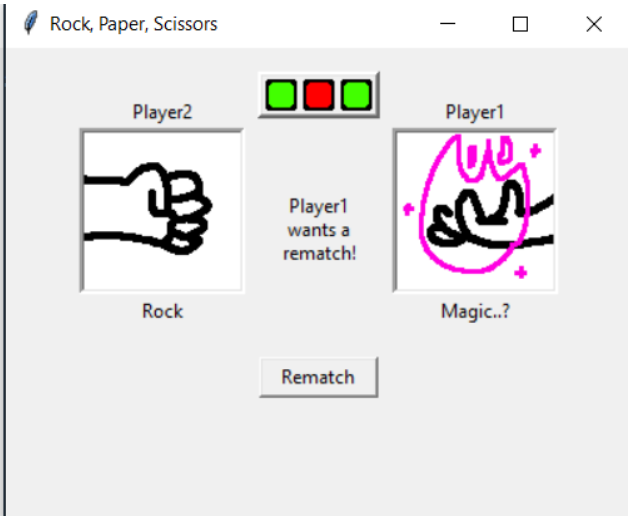

Rock, Paper, Scissors
Player1
Player2
Player2 connected!
Waiting...
Waiting...
Your moves:
Rock
Paper
Scissors
Magic..?


Rock, Paper, Scissors
Player2
Player1
Connected!
Waiting...
Waiting...
Your moves:
Rock
Paper
Scissors
Magic..?


Rock, Paper, Scissors
Player1
Player2
You played Magic..?
Magic..?
Waiting...


Rock, Paper, Scissors
Player2
Player1
Player1 played.
Waiting...
Waiting...
Your moves:
Rock
Paper
Scissors
Magic..?


Rock, Paper, Scissors
Player1
Player2
Player1 wants a rematch!
Magic..?
Rock
Rematch


Rock, Paper, Scissors
Player2
Player1
Player1 wants a rematch!
Rock
Magic..?
Rematch