

Zeid Solh
205323715
EC ENGR 180DA
Lab 3

Screenshot of my single player rock paper scissors game in the command line.

A screenshot of a Mac OS X desktop environment. In the center is a terminal window titled "rps.py - Lab 3". The code in the terminal is:

```
15     print("Paper covers rock! You lose.")
16 elif user_action == "paper":
17     if computer_action == "rock":
18         print("Paper covers rock! You win!")
19     else:
20         print("Scissors cuts paper! You lose.")
21 elif user_action == "scissors":
22     if computer_action == "paper":
23         print("Scissors cuts paper! You win!")
24     else:
25         print("Rock smashes scissors! You lose.")
26 play_again = input("Play again? (y/n): ")
27 if play_again.lower() != "y":
28     break
```

The terminal output shows the game in progress:

```
You chose rock, computer chose scissors.  
Rock smashes scissors! You win!  
Play again? (y/n): y  
Enter a choice (rock, paper, scissors): paper  
You chose paper, computer chose paper.  
Both players selected paper. It's a tie!  
Play again? (y/n): n
```

The desktop background is a blue ocean scene. The dock at the bottom contains various application icons.

Screenshot of my multiplayer rock paper scissors game in the command line. (with Warren)

A screenshot of a Mac OS X desktop environment. In the center is a terminal window titled "rps.py - Lab 3". The code in the terminal is:

```
1 import paho.mqtt.client as mqtt
2 import numpy as np
3
4 def rps(rival):
5     user_gesture = int(input('Input 0 for Rock, 1 for Paper, 2 for Scissors: '))
6     comp_gesture = rival
7     mapping = {0 : 'Rock', 1 : 'Paper', 2 : 'Scissors'}
8
9     if user_gesture == (comp_gesture-1):
10        print("Rival chooses (mapping[comp_gesture]); Rival wins!")
11    elif comp_gesture == (user_gesture-1):
12        print("Rival chooses (mapping[comp_gesture]); You win!")
13    else:
14        print("Rival chooses (mapping[comp_gesture]); Tie!")
15    return user_gesture
16
17 # 0. define callbacks - functions that run when events happen.
18 # The callback for when the client receives a CONNACK response from the server.
19 def on_connect(client, userdata, flags, rc):
20     print("Connection returned result: " + str(rc))
21     # Subscribing in on_connect() means that if we lose the connection and
22     # reconnect then subscriptions will be renewed.
23     client.subscribe("ec180d/warren/rps", qos=1)
24     # The callback of the client when it disconnects.
25 def on_disconnect(client, userdata, rc):
26     if rc != 0:
27         print("Unexpected Disconnect")
28     else:
29         print("Expected Disconnect")
30     # The default message callback.
31     # (you can create separate callbacks per subscribed topic)
32     def on_message(client, userdata, message):
33         #print("Message topic: " + message.topic)
34         #print("Message payload: " + str(message.payload))
```

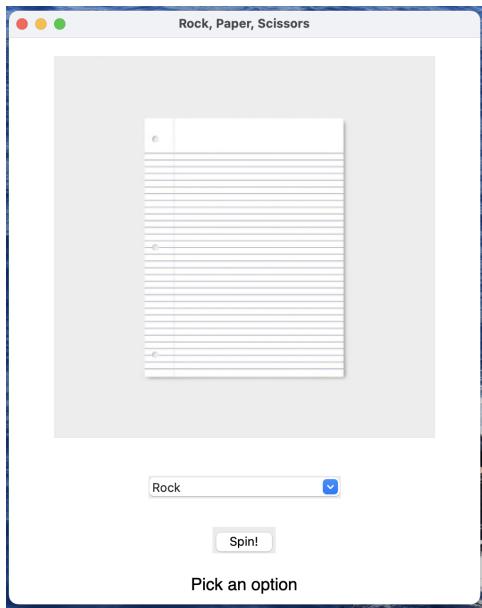
The terminal output shows the game in progress:

```
Connection returned result: 0
Received message on topic ec180d/warren/rps with QoS 1
Input 0 for Rock, 1 for Paper, 2 for Scissors: 2
Rival chooses Rock; Computer wins!
Received message on topic ec180d/warren/rps with QoS 1
Input 0 for Rock, 1 for Paper, 2 for Scissors: 0
Rival chooses Scissors; Computer wins!
^CTraceback (most recent call last):
File "/Users/zeidsolh/Library/Mobile Documents/com~apple~CloudDocs/Studies/UCLA/UCLA Quarter 11/EC ENGR 180DA/180DA-WarmUp/Lab 3/rps-multiplayer.py", line 54, in <module>
```

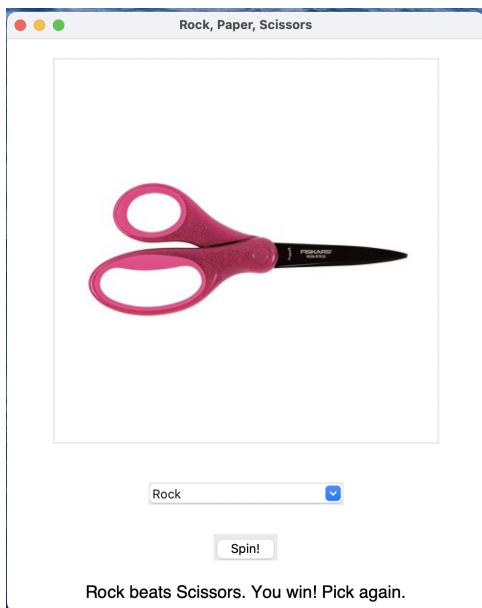
The desktop background is a blue ocean scene. The dock at the bottom contains various application icons.

(In the example, it should have been “Rival chooses Rock; Rival wins!” instead of “Rival chooses Rock; Computer wins!” because we are now playing against a rival online instead of the computer)

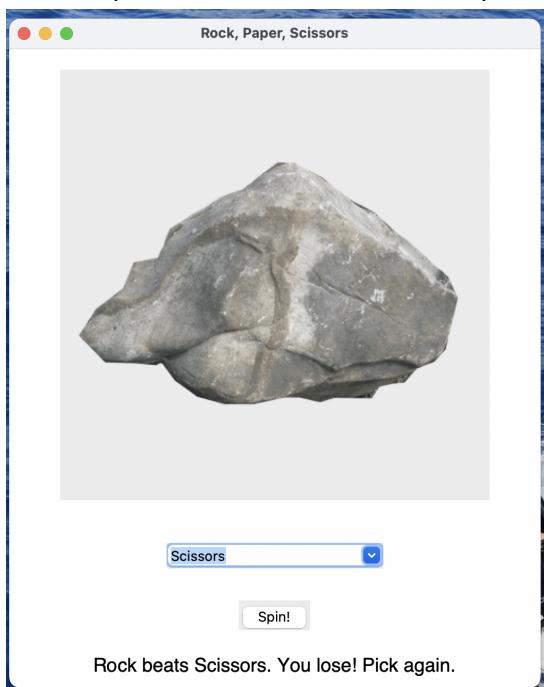
Screenshot of my single player GUI rock paper scissors game, computer is image, player is drop down box. This is the greeting page.



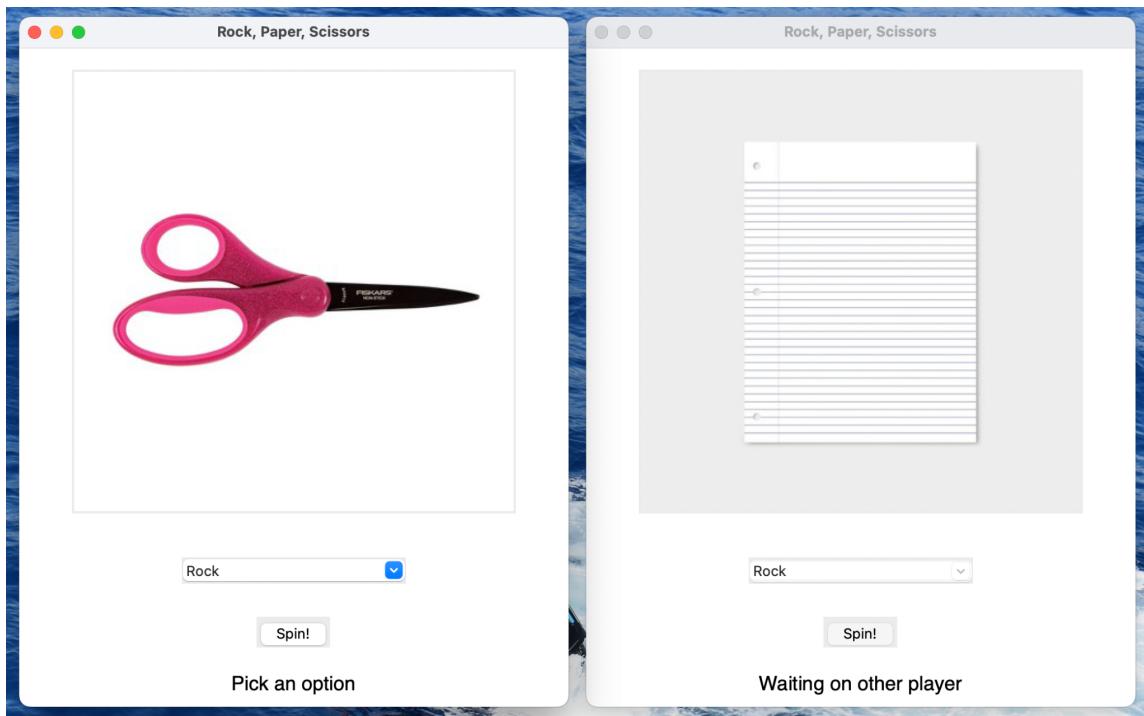
Below, I picked rock and the computer picked scissors. So I won.



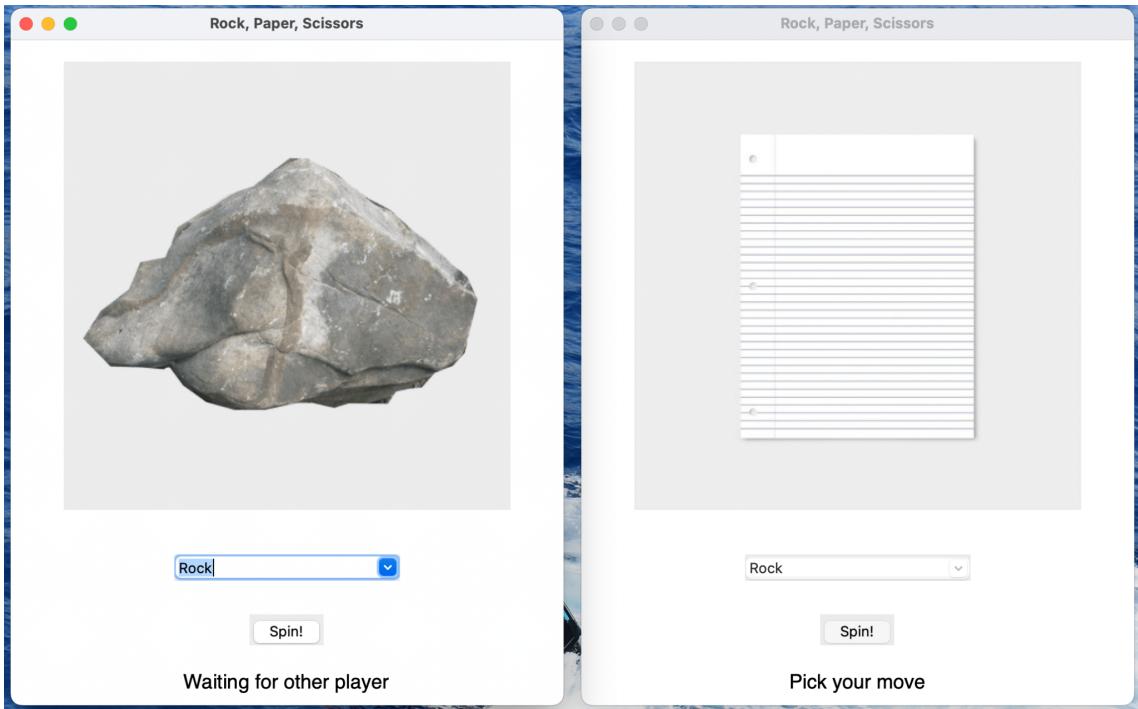
Below, I picked scissors and the computer picked rock. So I lost.



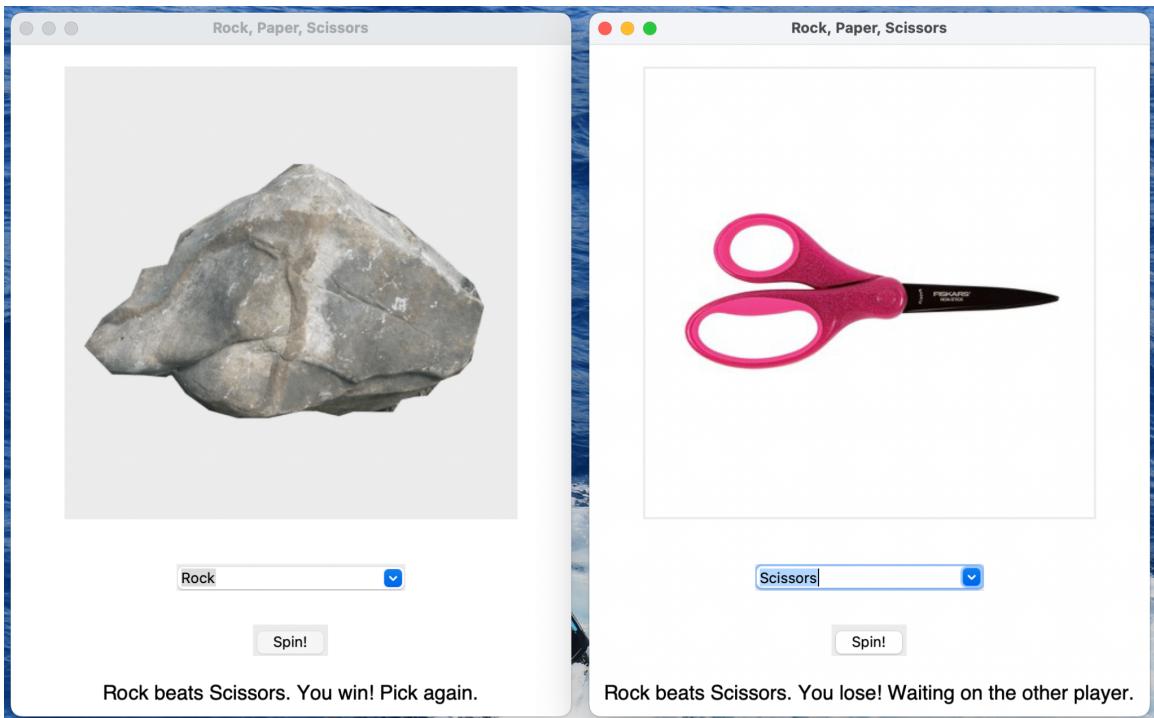
Screenshot of my multiplayer GUI rock paper scissors game, left window is player 1, right window is player. This is the greeting page. Player 1 always picks first, player 2 always picks second. I implemented both players on my end (didn't use a teammate).



Below, player 1 chose rock. So player 1 can see a picture of his rock. Player 2 still hasn't picked anything.



Below, player 2 chose scissors. So player 2 can see a picture of his scissors. Straight after selecting his option, he gets the message that he lost. Likewise, player 1 gets the option that he won since rock beats scissors.



Now player 1 can pick again and the game goes on and on.

We would like to use graphics in our project game because we want to create our own playable version of beat saber. So we will need graphics to show the blocks coming towards the player in order for the player to slice the blocks.