Project report

Tuesday, July 6, 2021

Python ML Internship Pie Infocomm Pvt. Ltd.

Project Name: - RPS Game

RPS – ROCK PAPER & SCISSOR

Name: - Nikhil Mishra (nikhil.mishra_cs18@gla.ac.in) GLA University

Content of Report:

INTRODUCTION

I address the problem of how to implement a stateful multi-player game while minimizing trust in third-parties. By “stateful” we mean that there is some meaningful state that persists between separate interactions, as opposed to each interaction being self-contained, as in chess, tic-tac-toe, etc. A typical example of a stateful game is poker, since each hand affects the money balances held by each player, which persist between hands.

Most games that exist today prevent cheating either by centralization requiring that all interactions be routed through a trusted third-party solely responsible for maintaining the game state; or else by obfuscating the software running on each player’s device to prevent them from illegally altering the state.

In our project we seek a way to use cryptographic technology to alleviate these constraints. Our project implements a minimal game of this type as a proof-of-concept.

OBJECTIVE

For this project I have implemented a minimal example of a game with these properties, which we call “Rock-Paper-Scissors-with-state” (RPS). The rules of the game are as follows:

1. Any player may initiate an encounter with another player, who may or may not choose to accept. (The two players are known as the challenger and the defender, respectively.) No player may be in more than one encounter at the same time.

2. An encounter consists of a number of rounds. In each round, each player commits to their move (either Rock, Paper, or Scissors), and reveals the commitment to the other player. The winner of each round is found according to standard Rock-Paper-Scissors rules (Random Function).

3. The player to win two out three rounds win the encounter. When the encounter concludes, each player’s “skill rating” is adjusted according to a simple formula1 . Players’ skill ratings persist between encounters.

BACKGROUND

I created two individual components as part of our game prototype: voter, and client. Voters connect to a random function program. By keeping a list of connected voters, it can relay a message received from a voter. In the current implementation, a client does not directly join the voter network, but instead posts and receives messages from voters or with the help of random function.

The voter is a game-proof verifier that records and maintains the global game state.

HARDWARE AND SOFTWARE REQUIREMENT

Hardware: Laptop with updated Windows 10

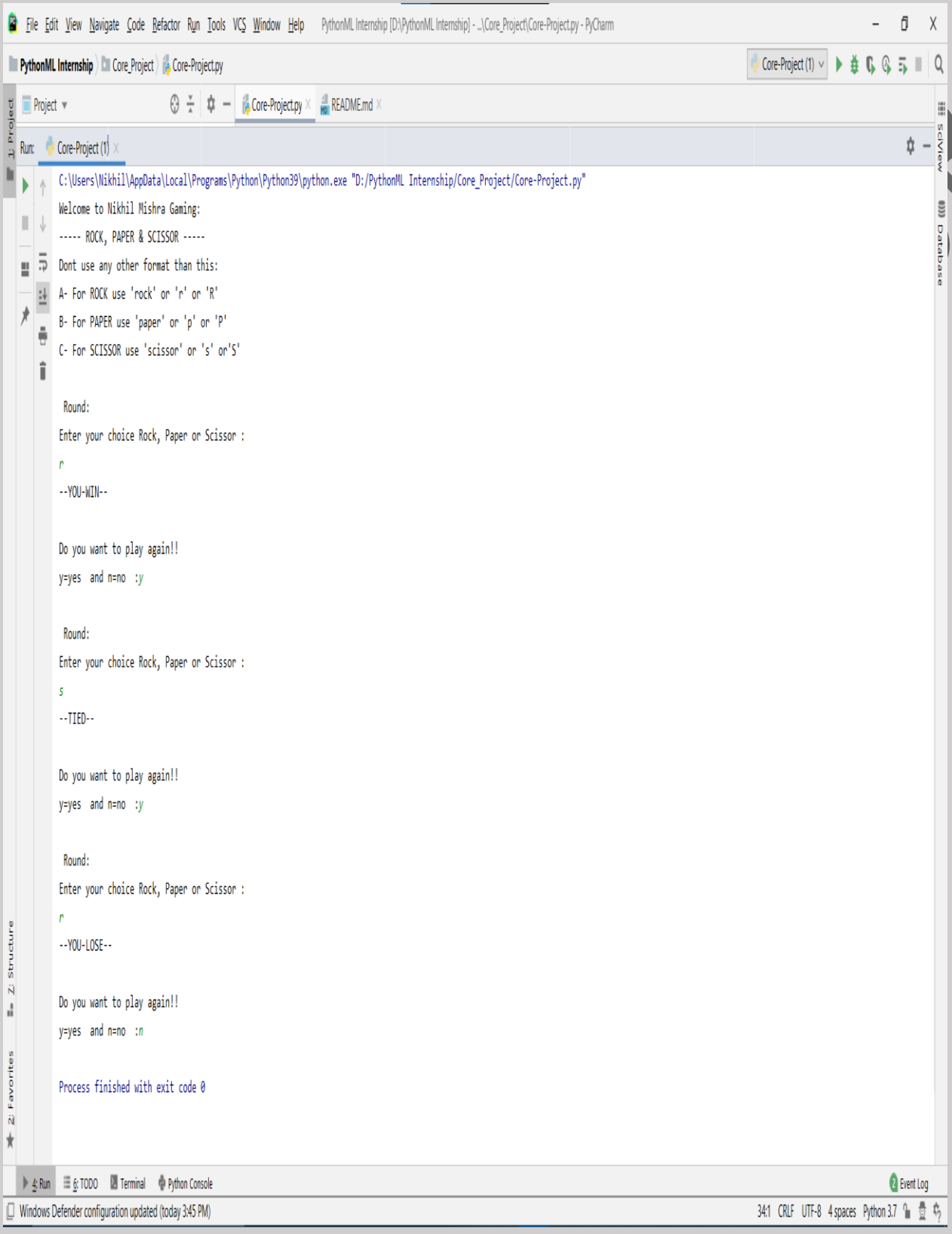
Software : Pycharm

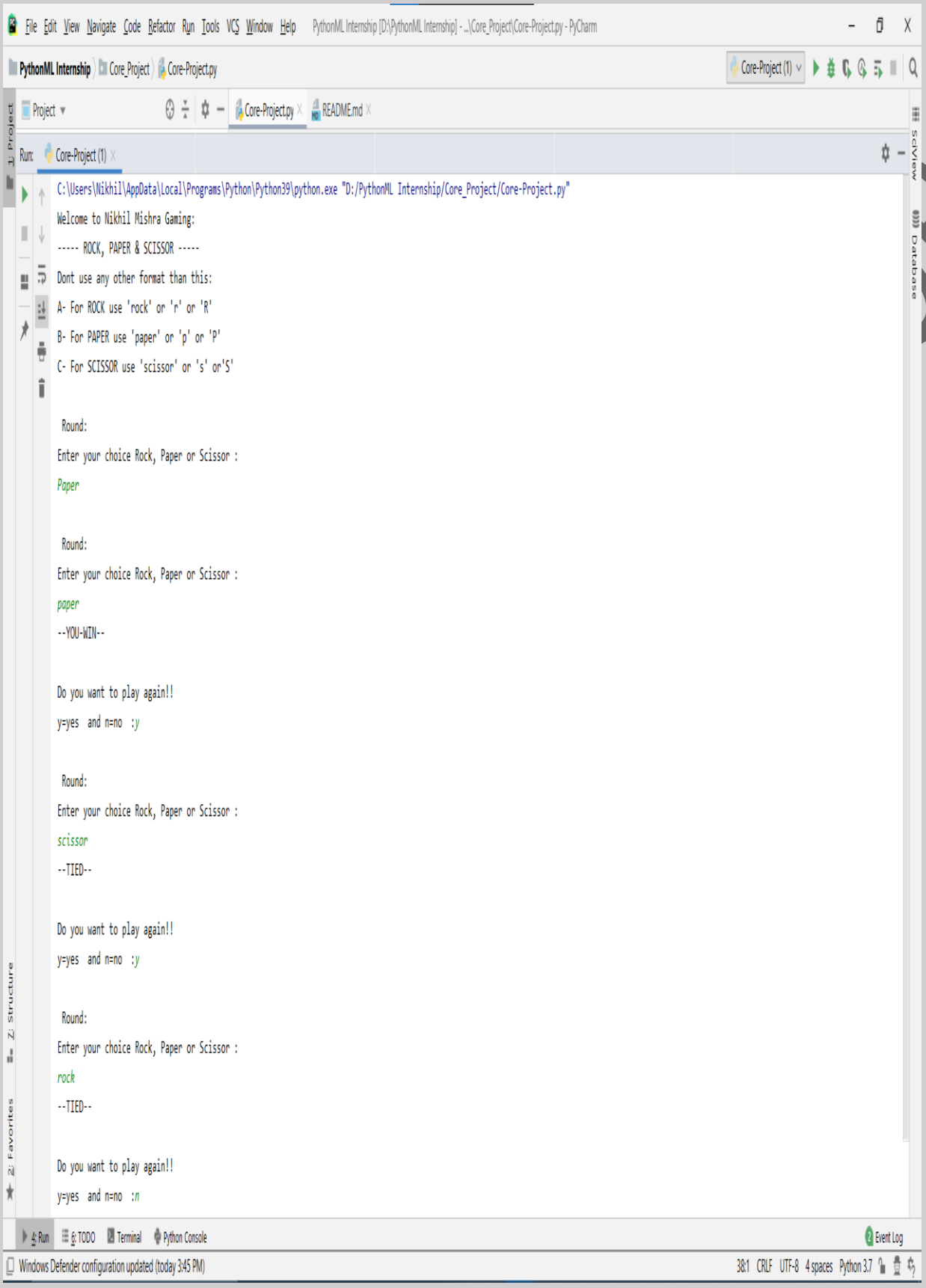
CODING

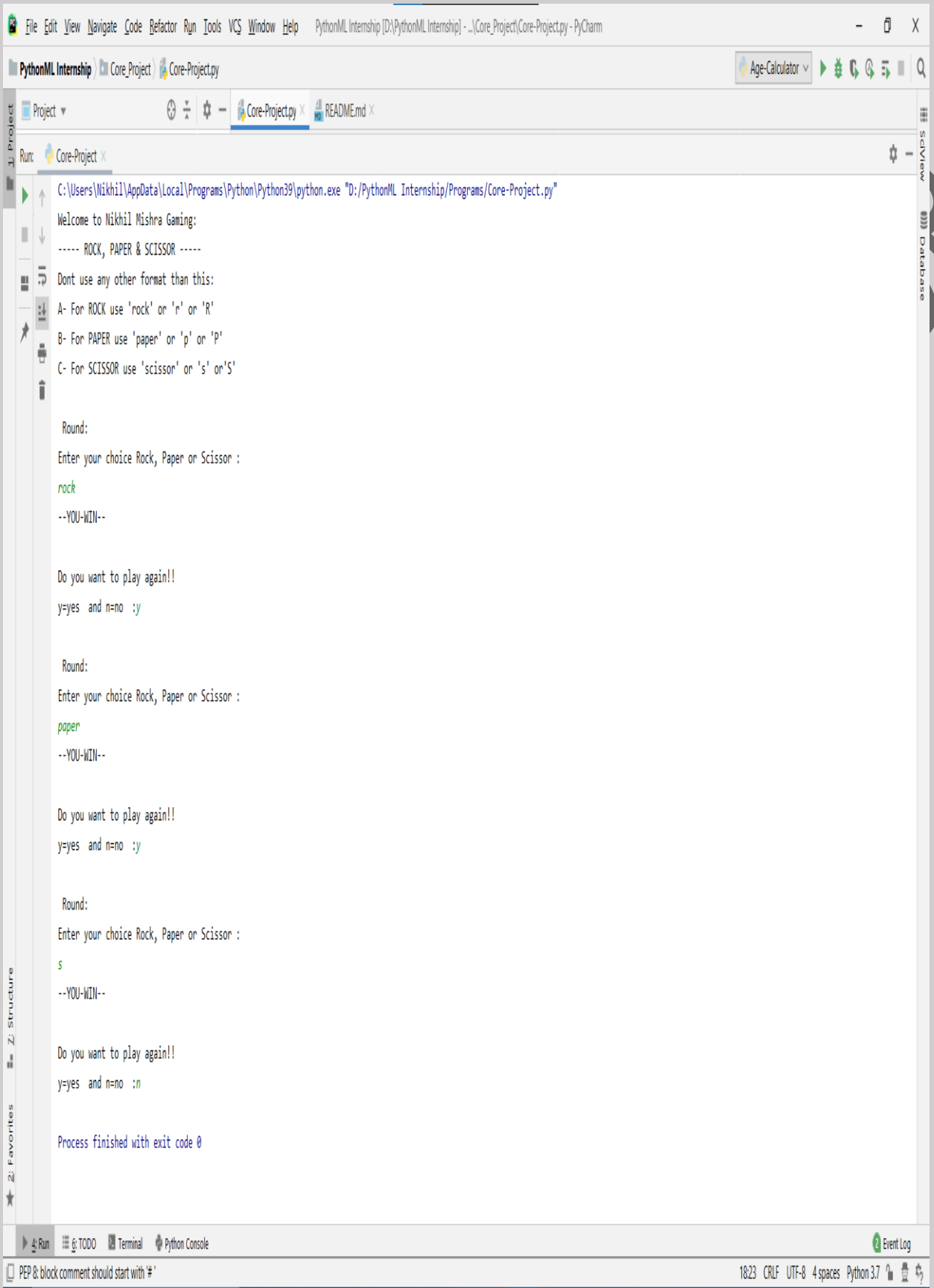
```
import time
import random
#Instructions
print("Welcome to Nikhil Mishra Gaming:")
print("----- ROCK, PAPER & SCISSOR -----")
time.sleep(1)
print("Dont use any other format than this:")
time.sleep(1.5)
print("A- For ROCK use 'rock' or 'r' or 'R' ")
time.sleep(1.5)
print("B- For PAPER use 'paper' or 'p' or 'P' ")
time.sleep(1.5)
print("C- For SCISSOR use 'scissor' or 's' or'S'")
time.sleep(1.5)


def statement():
    #Taking User Input
    print("Enter your choice Rock, Paper or Scissor : ")
    choice = input()
    choose = ["R", "P", "S"]
    #Making the Computer Choose
    comp = random.choice(choose)
    return choice, comp


def game():
    i = 1
    comw = 0
    youw = 0
    while i <= 100:
        print()
        print(" Round:")
        user, com = statement()
        #Determine a Winner, Loser or a Tie
        if (user == "rock" or user == "r") or user == "R":
            if com == "R":
                print("--TIED--")
                print()
                break
            elif com == "P":
                print("--YOU-LOSE--")
                comw = comw + 1
                print()
                break
            elif com == "S":
                print("--YOU-WIN--")
                youw = youw + 1
```

```
                print()
                break
            else:
                print("Sorry you used illegal format!!")
                print()
                break
        if (user == "paper" or user == "p") or user == "P":
            if com == "P":
                print("--TIED--")
                print()
                break
            elif com == "S":
                print("--YOU-LOSE--")
                print()
                comw = comw + 1
                break
            elif com == "R":
                print("--YOU-WIN--")
                youw = youw + 1
                print()
                break
            else:
                print("Sorry you used illegal format!!")
                print()
                break
        if (user == "scissor" or user == "s") or user == "S":
            if com == "S":
                print("--TIED--")
                print()
                break
            elif com == "R":
                print("--YOU-LOSE--")
                comw = comw + 1
                print()
                break
            elif com == "P":
                print("--YOU-WIN--")
                youw = youw + 1
                print()
                break
            else:
                print("Sorry you used illegal format!!")
                print()
                break
    #Play Again
    print("Do you want to play again!!")
    time.sleep(0.6)
    stop = input("y=yes  and n=no  :")
    if stop == "y":
        game()
    else:
        exit()
game()
```

OUTPUT SCREENSHOOTS




File Edit View Navigate Code Refactor Run Tools VCS Window Help
PythonML Internship [D:\PythonML Internship] - ...\Core_Project\Core-Project.py - PyCharm
PythonML Internship
Core_Project
Core-Project.py
Core-Project (1)
Project
Core-Project.py
README.md
Run:
Core-Project (1)
C:\Users\Nikhil\AppData\Local\Programs\Python\Python39\python.exe "D:/PythonML Internship/Core_Project/Core-Project.py"
Welcome to Nikhil Mishra Gaming:
----- ROCK, PAPER & SCISSOR -----
Dont use any other format than this:
A- For ROCK use 'rock' or 'r' or 'R'
B- For PAPER use 'paper' or 'p' or 'P'
C- For SCISSOR use 'scissor' or 's' or'S'
Round:
Enter your choice Rock, Paper or Scissor :
Paper
Round:
Enter your choice Rock, Paper or Scissor :
paper
--YOU-WIN--
Do you want to play again!!
y=yes and n=no :y
Round:
Enter your choice Rock, Paper or Scissor :
scissor
--TIED--
Do you want to play again!!
y=yes and n=no :y
Round:
Enter your choice Rock, Paper or Scissor :
rock
--TIED--
Do you want to play again!!
y=yes and n=no :n
4: Run
6: TODO
Terminal
Python Console
Event Log
Windows Defender configuration updated (today 3:45 PM)
38:1 CRLF UTF-8 4 spaces Python 3.7
SciView
Database
Structure
Favorites


File Edit View Navigate Code Refactor Run Tools VCS Window Help
PythonML Internship [D:\PythonML Internship] - ...\Core_Project\Core-Project.py - PyCharm
PythonML Internship
Core_Project
Core-Project.py
Age-Calculator
Project
Core-Project.py
README.md
Run:
Core-Project
C:\Users\Nikhil\AppData\Local\Programs\Python\Python39\python.exe "D:/PythonML Internship/Programs/Core-Project.py"
Welcome to Nikhil Mishra Gaming:
----- ROCK, PAPER & SCISSOR -----
Dont use any other format than this:
A- For ROCK use 'rock' or 'r' or 'R'
B- For PAPER use 'paper' or 'p' or 'P'
C- For SCISSOR use 'scissor' or 's' or'S'
Round:
Enter your choice Rock, Paper or Scissor :
rock
--YOU-WIN--
Do you want to play again!!
y=yes and n=no :y
Round:
Enter your choice Rock, Paper or Scissor :
paper
--YOU-WIN--
Do you want to play again!!
y=yes and n=no :y
Round:
Enter your choice Rock, Paper or Scissor :
s
--YOU-WIN--
Do you want to play again!!
y=yes and n=no :n
Process finished with exit code 0
4: Run
6: TODO
Terminal
Python Console
Event Log
PEP 8: block comment should start with '# '
18:23 CRLF UTF-8 4 spaces Python 3.7

FUTURE SCOPE

There are several ways a player can detect malicious activity by their opponent: they try and reveal a move they did not commit to, their signature does not verify, or they never issued the Initiate Encounter transaction.

In this case the player has no formal way to penalizing this behavior beyond choosing not to play another game with that player. In fact, if they choose to discontinue the game upon detecting malicious behavior, they themselves may be penalized for aborting the game prematurely. As a result, for future work, we propose extending functionality of player to-player interaction to handle player-side censure of malicious behavior. There are several open problems in this case, as this faces issues like those raised in the fairness or attribution of abort — it may be difficult for the voters to determine which party is being malicious.

A players is trying to play multiple games at once, in the case when the player is trying to play another game while he is in the middle of playing a game or has committed to playing one.

CONCLUSION

I have designed and implemented Rock-Paper-Scissor-withState to provide a verifiable, decentralized, game platform. Our voter (Random Function) network leverages existing consensus methods to allow for a decentralized global state and our players provide verifiable game encounters. It is our hope that this general platform with its security guarantees and protocol flexibility can provide a base for more complex game protocols that include verifiable randomness, increased number of parties, and more efficient proof verification.

REFERENCES AND BIBLIOGRAPHY

1:- https://www.google.com/

2:- https://www.youtube.com/

3:- David Shwartz, Noah Youngs, and Arthur Britto. The ripple protocol consensus algorithm. 2014.

THANK YOU !!

ROCK PAPER SCISSOR