

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY Kattankulathur, Chengalpattu District

Rock Paper Scissor Game

Using Socket Programming

## Submitted as a part of III year Course Requirement By

**Neetish Singh RA1911003010216**

**Shresth Kapoor RA1911003010220**

**Aryan Rai Ra1911003010221**

**NAME OF THE COURSE/COURSE CODE**

COMPUTER NETWORK/18CSC302J

Under the guidance of

Dr. P. Madhavan

(Associate Professor, Department of Computing Technologies)

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

(Under Section 3 of UGC Act, 1956)

**BONAFIDE CERTIFICATE**

Certified that this project report "Rock Paper Scissor Game using Socket Programming " is the bonafide work of Neetish Singh (RA1911003010216), Shresth Kapoor (RA1911003010220) and Aryan Rai (RA1911003010221) who carried out the project work under my supervision.

Signature Signature

Subject Staff Dr.E. Sasikala,

Designation Course Coordinator

Department Associate Professor,

Data Science and Business

SRM Institute of Science and Technology

System Potheri, SRM nagar, Kattankulathur,

Technology Tamil Nadu 603203

Kattankulathur

**ABSTRACT**

**Rock paper scissors** (also known by other orderings of the three items, with "rock" sometimes being called "stone", or as **Rochambeau**, **roshambo**, or **ro-sham-bo**) is a hand game, usually played between two people, in which each player simultaneously forms one of three shapes with an outstretched hand. These shapes are "rock" (a closed fist), "paper" (a flat hand), and "scissors" (a fist with the index finger and middle finger extended, forming a V). "Scissors" is identical to the two-fingered V-sign (also indicating "victory" or "peace") except that it is pointed horizontally instead of being held upright in the air.

A simultaneous, zero-sum game, it has only two possible outcomes: a draw, or a win for one player and a loss for the other. A player who decides to play rock will beat another player who has chosen scissors ("rock crushes scissors" or sometimes "blunts scissors"[[4]](https://en.wikipedia.org/wiki/Rock_paper_scissors#cite_note-4)), but will lose to one who has played paper ("paper covers rock"); a play of paper will lose to a play of scissors ("scissors cuts paper"). If both players choose the same shape, the game is tied and is usually immediately replayed to break the tie. The type of game originated in China and spread with increased contact with East Asia, while developing different variants in signs over time.

Rock Paper Scissors Online in Python which we have developed is a cross-platform game that should work on any modern desktop operating systems. This will be able to run on Windows and Mac OS X as well.

All the Python scripts in this project are written for Python 3.x. The client GUI requires Python PyGame module.

**Main Files**

**Server.py -:** All clients must interact with the socket server which is created in this file. The file manages all the game board and the queue match making logic, credential validations + encryption, authentication and client to client interactions.

Client.py -: This file manages everything that the client sees and interact with to communicate to the socket server. Also contains the in-built python PyGame module which we have used to improve the user interface to play the game.

**REQUIREMENT SPECIFICATION**

**Hardware Requirements:**

Processor : 2.4 GHz Clock Speed

RAM : 1 GB

Hard Disk : 500 MB(minimum free space)

**Software Requirement:**

Operating System : Windows (7/8/10)

Platform : PyCharm (IDE)

Backend : Python 3.x Socket Programming

Special Tool : PyGame Module (UI design)

Server & Client : Command line/Terminal on localhost

**ACKNOWLEDGEMENT**

We express our heartfelt thanks to our honourable Vice Chancellor Dr. C.

MUTHAMIZHCHELVAN, for being the beacon in all our endeavours.

We would like to express my warmth of gratitude to our Registrar Dr. S. Ponnusamy, for his encouragemWe express our profound gratitude to our Dean (College of Engineering and Technology) Dr. T. V. Gopal, for bringing out novelty in all executions.

We would like to express my heartfelt thanks to Chairperson, School of Computing Dr. Revathi Venkataraman, for imparting confidence to complete my course project.

We wish to express my sincere thanks to Course Audit Professor Dr.M. LAKSHMI, Professor and Head, Data Science and Business Systems and Course Coordinator Dr.E. Sasikala, Associate Professor, Data Science and Business Systems for their constant encouragement and support.

We are highly thankful to my Course project Internal guide Subject handling Dr. P Madhavan, Associate Professor, Computer Science and Engineering, for his assistance, timely suggestion and guidance throughout the duration of this course project.

We extend my gratitude to Student HOD Dr. S. S. Sridhar, Computer Science Engineering and my Departmental colleagues for their Support.

Finally, we thank our parents and friends near and dear ones who directly and indirectly contributed to the successful completion of our project. Above all, I thank the almighty for showering his blessings on me to complete my Course project.

# TABLE OF CONTENTS

### CHAPTER 1 INTRODUCTION

STATEMENT OF THE PROBLEM

* 1. OBJECTIVES
  2. NEED FOR THE STUDY

### Page Number

* 1. SCOPE & SIGNIFICANCE OF THE STUDY
  2. LIMITATIONS

### CHAPTER 2 REVIEW OF LITERATURE

* 1. NEWSPAPER ARTICLES
  2. MAGAZINE ARTICLES
  3. INFORMATION FROM WEB

### CHAPTER 3 RESEARCH METHODOLOGY

* 1. TIME SPAN
  2. HYPOTHESIS
  3. SAMPLING PROCEDURE
  4. DATA COLLECTION
  5. INSTRUMENT
  6. TOOLS USED FOR ANALYSIS

### CHAPTER 5 INTERPRETATION & FINDINGS

* 1. DATA INTERPRETATION

**CHAPTER 6 SUGGESTIONS & CONCLUSION**

# CHAPTER 1

**INTRODUCTION**

**Problem Statement:**

This game is generally played by using our hands and fingers between 2 players but today in this modern technology everything in online and fully based on technology, so we tried to build this game using socket programming in python in order to facilitate the users connect with server and play the game on the system.

Objectives:

1. To develop a Rock paper scissors game that uses Sockets to allow players (clients) to play against other clients on different (or the same) computers that are on the same network.
2. This game is very popular and is fairly simple by itself. It is actually a two-player game.
3. This application should be a fun game for students to verse one another in a game of tic tac toe

Scope and Significance:

1. Program the designed user interface with python Pygame and make each button perform the correct function
2. Research on how to use sockets with python and pickling Python objects. Sockets and pickling go beyond the advance higher course; however, it will be required to complete this application.
3. It’s used by both client and server to send pickled python objects to each other in bytes. (Then unpickled the bytes back to the python object when the data is received).
4. There is no development cost for this project because I am using pythons’ built-in libraries to develop the whole application. Therefore, there are no third-party dependency costs

Limitations:

1. The server will automatically close a game session when one player is inactive or disconnected.
2. Users can play game only on localhost computer not on their different computers/networks.
3. No database to store the users winning progress or no database to show leader board.

**CHAPTER 2: REVIEW OF LITERATURE**

The literature indicates that players in the RPS game are consistent with the Nash strategy, at least some of the time. However, this literature has frequently used a payoff matrix that cannot distinguish between Nash and random play, and strategies other than random have often been reported. Research also demonstrates that players have a preference for repeating specific sequential patterns. For example, Eyler et al. noted that players tended to repeat choices consecutively (e.g., rock-rock-rock) or to cycle in a sequence (e.g., rock-paper-scissors). This “cycling” behavior includes attempts to “upgrade” (i.e., each choice beat the previous one such as rock-paper-scissors) or to downgrade (i.e., each choice was beaten by the previous one such as scissors-paper-rock) their choices. Importantly, results also indicate that players had a weak tendency to stay with the previous selection following a win and to switch the selection following a loss. Dyson et al. concluded that this strategy reflects the classic behavior principle that reinforced responses are more likely to be used again (“win-stay”), whereas non-reinforced responses are less likely to be picked immediately after (“lose-change”). This WSLC strategy in RPS has been observed by others as well. Forder and Dyson argued that humans follow a WSLC strategy, but that “win-stay” relies more on System 2 (e.g., rational and strategic planning) processes, while “lose-change” relies on System 1 (intuitive and impulsive) processes.

The studies discussed above provide some evidence that people are capable of predicting the opponent’s selection and acting to counter such predictions. This predictive ability means that a player’s own strategy is also exposed to the opponent, and thus the player can also be “predicted”. Imagine that Player 2 is temporarily choosing paper more often. Player 1 could choose scissors as a result, but Player 1 would also expect that Player 2 would know that Player 1 is expected to choose scissors, and consequently Player 2 chooses rock more often. Such recursive thinking and assumptions on knowing what Player 2 knows about Player 1 are known in psychology as “theory of mind” (ToM) . Generally speaking, ToM refers to the ability to infer and interpret the beliefs, desires, and intentions of others.

ToM is known to be an essential component of human learning of social phenomena, including the acquisition of social norms and social beliefs. For example, predicting that the opponent Player 2 will select rock can lead Player 1 to choose paper, which is named as the zero-order ToM. In addition, Player 1 can assume that Player 2 can also deduce that Player 1 would play paper (since Player 1 knows that Player 2 will play rock), thus Player 2 might switch to paper instead. In this scenario, Player 1’s assumptions about Player 2’s knowledge about Player 1 could make Player 1 choose scissors instead (i.e., first-order ToM). De Weerd et al. implemented computational agents that use simulated ToM play the RPS game and found that first-order and second-order agents outperformed agents of the lower order of ToM. Although the results are based only on computer simulations, it suggests that players who consider others’ intentions would have a distinct advantage in the RPS game.

**INFORMATION FROM WEB:**

The current study provides the following contributions. As previous noted, past research that focused on players’ aggregated frequency strategy observed deviations from the Nash Equilibrium strategies among players. However, playing randomly implies a 33% of each action (R, P, and S) is also the optimal Nash Equilibrium strategy. Thus, it is not possible to know whether participants deviate from Nash or deviate from the random behavior given that random and Nash strategies are confounded in the payoff matrix. In the current research, we clarify this behavior by using a novel payoff matrix that allows us to make such distinctions. Second, past research examining sequential choices has shown that players play in agreement with the WSLC, or other types of complex strategical patterns, to exploit the opponents’ actions. In this research, we look at the circumstances in which these WSLC, and other sequential patterns, reflect human strategies.

**CHAPTER 3: RESEARCH METHODOLOGY**

**TIME SPAN:**

1. Five days to research process for collecting resources and installing in built python
2. 7 days of coding both server and client end.
3. 2 days of testing on various systems, operating systems.
4. Testing for different environments working.
5. Final 1 day of finishing touch to the project.

**HYPOTHESIS:**

* Both server and both the players are present on the same system i.e., they are playing on localhost.
* Only Two players are playing at the same time.
* Client is responsible for closing the connection
* No database to store the winning record so client is responsible for monitoring their

winning records.

* Python module is already installed on users’ systems to run the game on command line.

**SAMPLING PROCEDURE:**

Let i = {pR1, pP1, pS1} be the probabilities with which Player 1 chooses rock, paper, and scissors and j = {pR2, pP2, pS2} be the probabilities for Player 2. In the Nash Equilibrium play, Player 1’s payoff ERock = EPaper = EScissors for Player 2’s combination of selections. Thus, in the payoff matrix design in Table 1, the expected payoff for Player 1 for playing rock ERock = 2 × pR2 + 1 × pP2 + 4 × pS2 equals the expected payoff of playing paper EPaper = 3 × pR2 + 2 × pP2 + 1 × pS2 and playing scissor EScissor = 0 × pR2 + 3 × pP2 + 2 × pS2 when Player 2 chooses with probabilities j = {pR2 = 1/4, pP2 = 1/2, pS2 = 1/4}. Because the payoff matrix is symmetrical, Player 1 should also have the same choice probability so that Player 2’s expected value for choosing each action is the same as well, which gives the solution for the Nash Equilibrium.

Participants have to choose one of three buttons with the names written on it

as rock , paper, and scissor s. a player could not change the selection after clicking. Once both players made a choice, they were notified of the points obtained from the outcome, as well as their total accumulated points.

**DATA COLLECTIONS:**

* Imported the socket module from python library.
* Installed and imported the PyGame module.
* Imported threading module to run our threads.
* Imported the time module to introduce time gap between playing chances.
* How to use sockets in python.
* Pickling in python

**INSTRUMENTS:**

PYTHON SOCKETS & PICKLING PYTHON OBJECTS

Sockets allowed us to connect 2 nodes on a network. We need to use sockets to create a connection between the players and the server. This will then allow us to update both players that are in the same game session and allow them to play against each other on different clients.

On server and client side we have used socket module in python. On client side we have used Pygame module present in python to make it interactive.

The following client user interfaces will be created using python Pygame. Each design aims to:

1. Making each label clear, predictable and easy to follow.
2. Considers the user's experience, by including a dedicated section to inform the client as to what’s happening (errors and action messages.)
3. The designs illustrate navigational buttons or events in a clear way.

**TOOLS USED FOR ANALYSIS:**

* Command Line on windows: To test the outputs
* Linux Terminal.: To test the project on different environments.
* GitHub: For saving the files.
* Microsoft Vs Code.: For Coding.
* PyCharm IDE: For checking the interface design.
* PyGame Module: For User Interface (UI) design.

**CHAPTER 4: INTERPRETATION & FINDINGS**

**DATA INTERPRETATION:**

When considering only the state of the game there are 3 choices for both the players on the screen. A combination study of the game shows that when player 1 chooses rock and player 2 chooses scissor p1 wins, if p2 chooses paper p1 loses, and if p2 chooses rock game tie.

Simultaneously scissor will win against paper, paper will win against stone and rock will win against scissor.

If the choices of the players are same then the game will be tied.

There are 33.33% chances of winning a game after choosing either of 3 choices(rock, paper, and scissor).

At least, that's what chance predicts. But people do not play randomly - they follow hidden patterns that you can predict to win more games than you should, a study has revealed.

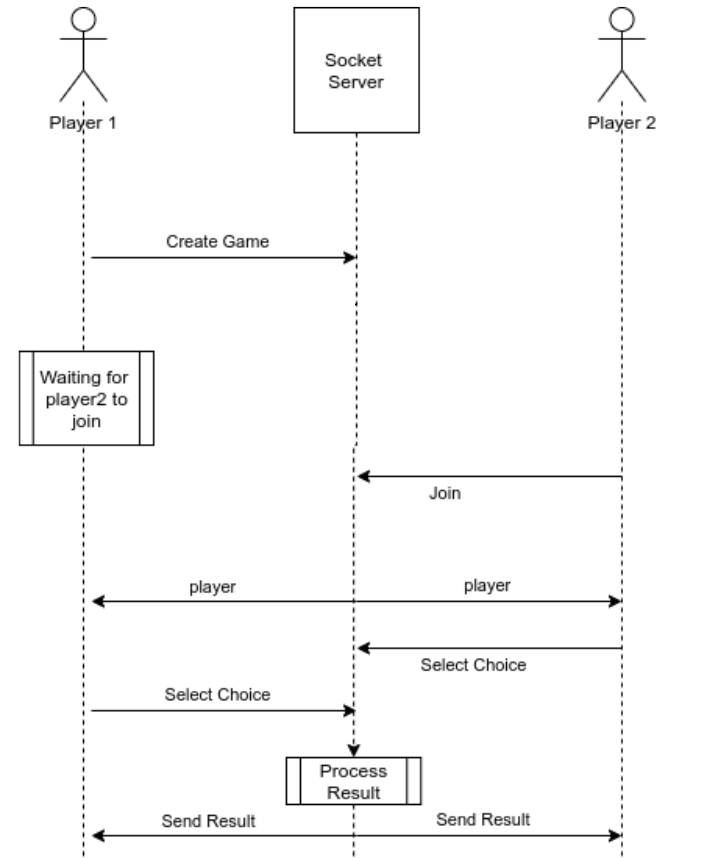
**CHAPTER 5: IMPLEMENTATIONS AND DESIGN**

**GAMEPLAY SYSTEM DESIGNS:**

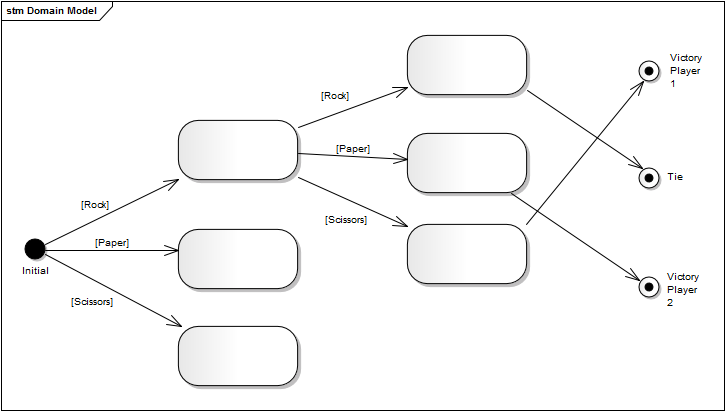
* **Architecture Diagram**: Architectural design is a process for identifying the subsystems making up a system and the framework for sub-system control and communication. The output of this design process is a description of the software architecture. Architectural design is an early stage of the system design process.



* Flow Chart: A flowchart is a type of diagram that represents an algorithm, workflow or process. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. Flowcharts are used in analyzing, designing, documenting or managing a process or program in various fields.



**STM Domain Model Of Rock Paper Scissors :**



**PROJECT IMPLEMENTAION DESCRIPTION:**

* The Rock Paper Scissor game client application provides an intuitive user interface for connecting to the server and playing the game. Initially a connect "frame" is opened where a player enters and connects to the server.
* After connecting, the connect frame is hidden and the main game window is displayed.
* The game server listens for and accepts client’s connections. The server mainly acts as a forwarder i.e., forwards Player 1 name and its marked XY coordinates to Player 2 and vice versa. The server doesn't do any logic — such as executing game rule, coordinating player’s turn and computing scores because we have handled those at the client application for improved communication efficiency.
* When both the players are connected , now either of players can start the game by giving there choices and then he or she will wait for the other player to respond by choosing his option out of three then the game result will be displayed on the screen that which player has won this particular round and simultaneously now again any player can choose his or her choice for the second round.

**CHAPTER 6: SUGGESTIONS & CONCLUSION:**

**SUGGESTIONS:**

* **SOFTWARE REQUIREMENTS:**

1. Windows Command Line Terminal / Linux Terminal / Python IDE with terminal built in.
2. Stable connections and can play only on Localhost PC.
3. PyGame module installed in IDE / Terminals.

* **MAINTENANCE:**

1. Adaptive Maintenance
2. Perfective Maintenance
3. Preventive Maintenance
4. Corrective Maintenance

* **FUTURE ENHANCEMENTS:**

1. Inclusion of sources to play the game on different systems.
2. Introductions of database to store users progress and leaderboard.
3. Introduction of difficulty levels.

**CONCLUSION:**

* Rock Paper Scissors game in Python which we have developed is a cross-platform game that should work on any modern desktop operating systems.
* All the Python scripts in this project are written for Python 3.x. The client GUI requires Python PyGame module.
* Hence, we have tried to developed a game in virtual mode in order to compete and be in today’s technology world.
* The purpose of this documentation is to capture all the requirements by which the user can play a game of Rock Paper Scissor in Graphical User Interface as well as they can develop a logic that what is actually happening

# REFERENCES

(The articles should be arranged in alphabetical order according to the format given below)

1. “Marketing Management” by Philip Kotler, 13th Revised Edition, Pearson Education.
2. “Spice Telcom takes a giant leap in VAS”, [www.thetimesofindia.com,](http://www.thetimesofindia.com/) 21st June 2010.
3. “TRAI holds back number portability”, Business Line, 6th April 2008.