



**UNIVERSAL ENGINEERING AND SCIENCE COLLEGE  
AFFILIATED TO POKHARA UNIVERSITY**

**A Minor Project Progress  
on  
“ROCK PAPER SCISSOR GAME”**

**Submitted By:**

Samir Shrestha (21070664)

Bhawana Ojha (21070631)

**Submitted To:**

Department of Computer Science and Engineering  
UESC, Chakupat, Lalitpur

**Under The Supervision of:**

Er. Hemant Joshi

July, 2023

## **DECLARATION**

We hereby declare that the report of the project entitled “Rock Paper Scissor Game” which is being submitted to the Department of Computer Science and Engineering, Universal Engineering and Science College, Chakupat Lalitpur, in the partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Computer Engineering, is a bonafide report of the work carried out by us. The materials contained in this report have not been submitted to any University or Institution for the award of any degree and we are the only author of this complete work and no sources other than the listed here have been used in this work. .

Sincerely,

Samir Shrestha (21070664)

Bhawana Ojha (21070631)

Date: July, 2023

## CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the Department of Computer Science and Engineering, Universal Engineering and Science College, a minor project work entitled “Rock Paper Scissor Game” submitted by Bhawana Ojha, and Samir Shrestha, in partial fulfillment for the award of Bachelor’s Degree in Computer Engineering. The Project was carried out under special supervision and within the time frame prescribed by the syllabus.

We found the students to be hardworking, skilled and ready to undertake any related work to their field of study and hence we recommend the award of partial fulfillment of Bachelor’s degree of Computer Engineering.

.....

Project Supervisor

**Er. Hemant Joshi**

HOD, Department of Computer Science and Engineering, UESC

.....

External Examiner

Mr. .....

HOD, Department of Computer Engineering, .....

## **COPYRIGHT©**

The author has agreed that the library Department of Computer Science Engineering, Universal Engineering Science Campus , may make this report freely available for inspection. Moreover, the author has agreed that the permission for extensive copying of this project work for scholarly purpose may be granted by the professor/lecturer, who supervised the project work recorded herein or, in their absence, by the head of the department. It is understood that the recognition will be given to the author of this report and to the Department of Computer Science Engineering, Universal Engineering Science Campus in any use of the material of this report. Copying of publication or other use of this report for financial gain without approval of the Department of Computer Science Engineering, Universal Engineering Science Campus and author's written permission is prohibited.

Request for permission to copy or to make any use of the material in this project in whole or part should be addressed to:

.....

Head of Department,  
Department of Computer Engineering,  
UESC,Chakupat,Lalitpur

## **ACKNOWLEDGEMENT**

Apart from the efforts from our side, the success of any kind of project largely hinges upon the encouragement and guidance provided by many other parties.

First of all, We would like to express our sincere gratitude towards the department of computer engineering for the project work which provided us with the opportunity to learn and implement my knowledge in the form of Project work. We would like to acknowledge the authors of various papers and developers of the programming libraries that I have referenced for building my project. We also respect all the researcher's important time being used in the research and writing such articles. This Project work wouldn't have been initiated without them being kind enough to share their knowledge.

We are also thankful to our project supervisor **Er.Hemant Joshi** for his continuous monitoring and suggestion. We want to show our gratitude to all the teaching and non-teaching faculties of Department of Computer Engineering,UESC,Chakupat,Lalitpur for providing us with the resources and support which is needed for this Project work. Finally, We would like to thank all the people who are directly or indirectly related during my study and preparation of this project.

Samir Shrestha (21070664)

Bhawana Ojha (21070631)

## ABSTRACT

Our rock-paper-scissors game is a digital version of the classic hand game that is often used as a decision-making tool or for entertainment purposes. The game consists of user and computer who simultaneously choose one of three options representing rock, paper, or scissors. The winner is determined by the rules: rock beats scissors, scissors beats paper, and paper beats rock. Our target audience is casual gamers of all ages, and our goal is to provide a fun and interactive experience for players. The game contains attractive GUI and features like scorekeeping system. The game is developed using a python programming languages and tools.

**Keywords:** *Digital, casual, winner, options, score*

## TABLE OF CONTENTS

<b>DECLARATION</b>	i
<b>CERTIFICATE OF APPROVAL</b>	ii
<b>COPYRIGHT</b>	iii
<b>ACKNOWLEDGEMENT</b>	iv
<b>ABSTRACT</b>	v
<b>TABLE OF CONTENTS</b>	vi
<b>LIST OF FIGURES</b>	viii
<b>LIST OF TABLES</b>	ix
<b>LIST OF ABBREVIATIONS</b>	x
<b>1 INTRODUCTION</b>	1
1.1 Background . . . . .	1
1.2 Motivation . . . . .	1
1.3 Problem definition . . . . .	2
1.4 Objective . . . . .	2
1.5 Project Scope and Applications . . . . .	2
<b>2 LITERATURE REVIEW</b>	3
<b>3 SYSTEM ANALYSIS</b>	5
3.1 Requirement Analysis . . . . .	5
3.1.1 Hardware requirement . . . . .	5
3.1.2 Software Requirements . . . . .	5
3.2 Feasibility Analysis . . . . .	5
3.2.1 Technical Feasibility . . . . .	5
3.2.2 Economic Feasibility . . . . .	6
3.2.3 Legal Feasibility . . . . .	6
<b>4 METHODOLOGY</b>	7
4.1 System Block Diagram . . . . .	7
4.2 Flow chart . . . . .	8
<b>5 IMPLEMENTATION DETAIL</b>	9

5.1	Importing Modules . . . . .	9
5.2	Global variables . . . . .	9
5.3	Options tuple . . . . .	9
5.4	Opengame function() . . . . .	10
5.5	Playerchoice(playerinput) . . . . .	10
5.6	Other Helper Functions . . . . .	11
5.7	Start window . . . . .	11
5.8	Additional function . . . . .	11
<b>6</b>	<b>RESULT AND ANALYSIS</b>	<b>12</b>
<b>7</b>	<b>Work Schedule</b>	<b>16</b>
<b>8</b>	<b>CONCLUSION AND FUTURE WORK</b>	<b>17</b>
8.1	Conclusion . . . . .	17
8.2	Future Work . . . . .	17
<b>REFERENCES</b>		<b>18</b>

## LIST OF FIGURES

4.1	Block diagram of the system . . . . .	7
4.2	Flow chart of game . . . . .	8
6.1	GUI of starting window of RPS game . . . . .	12
6.2	GUI of main game window of our Rock Paper Scissor game . . . . .	12
6.3	Displaying users win . . . . .	13
6.4	Displaying computers win . . . . .	13
6.5	Displaying tie . . . . .	14
6.6	Displaying user as overall champion of the game . . . . .	14
6.7	Displaying computer as overall champion of the game . . . . .	15
6.8	Tutorial of the game . . . . .	15

## **LIST OF TABLES**

7.1 Time Schedule for the Project Work . . . . .	16
--	----

## **LIST OF ABBREVIATIONS**

AI	Artificial Intelligence
CPU	Central Processing Unit
GUI	Graphical User Interface
GPU	Graphics Processing Unit
OS	Operating System
RPS	Rock Paper Scissor

# CHAPTER 1

## INTRODUCTION

Rock paper scissors is a simple game played by two people. The game has three possible outcomes: rock beats scissors, scissors beats paper, and paper beats rock. The players count to three in unison and simultaneously choose one of three hand signs representing rock, paper, or scissors. The winner is determined by the rules: rock crushes scissors, scissors cuts paper, and paper covers rock. If both players choose the same hand sign, the game is a draw and is usually played again to determine the winner.

### 1.1 Background

The exact origins of rock paper scissors are not known, but it is believed to have originated in China and spread throughout the world through the Japanese culture. The game has been known by many different names and variations, including roshambo and scissors paper stone.

One theory suggests that the game dates back to the Han Dynasty in China (202 BC – 220 AD), where it was played with other forms of hand gestures representing animals such as birds and monkeys. Another theory suggests that the game originated in Japan, where it was played with paper, stone, and water.

### 1.2 Motivation

There are several possible motivations for making the rock paper scissors game:

- It is a fun project to learn new programming language or practice coding skills.
- It provide a challenging and entertaining experience for players.

### **1.3 Problem definition**

Design and implement a rock paper scissors game that allows a player to play against a computer. The game should have a user-friendly interface and should allow the player to choose their gestures (rock, paper, or scissors). The computer should also randomly select one of the three gestures. The game should then determine the winner based on the rules of rock paper scissors (rock beats scissors, scissors beats paper, paper beats rock). If both the player and the computer choose the same gesture, the game should be a draw and should be played again to determine the winner. The game should also keep track of the score and declare an overall winner when one player reaches a predetermined number of wins.

### **1.4 Objective**

This Project work aims to fulfill the following objectives:

- To develop the players strategy and decision-making skills through gameplay.
- To create a fun and engaging game.

### **1.5 Project Scope and Applications**

- It can be used as casual game for people who seek a quick and simple diversion from their daily routine.
- It can be used as an education tool to teach basic game design concept.
- It can be used as a decision making tool to make quick decision.

## CHAPTER 2

### LITERATURE REVIEW

In 22 June 2021, Hanshu Jhang published the research article. This research studied the strategies that players use in sequential adversarial games. They took the Rock-Paper-Scissors (RPS) game as an example and ran players in two experiments. The first experiment involved two humans, who played the RPS together for 100 times. Importantly, their payoff design in the RPS allowed them to differentiate between participants who used a random strategy from those who used a Nash strategy. They found that participants did not play in agreement with the Nash strategy, but rather, their behavior was closer to random. Moreover, the analyses of the participants' sequential actions indicated heterogeneous cycle-based behaviors: some participants' actions were independent of their past outcomes, some followed a well-known win-stay/lose-change strategy, and others exhibited the win-change/lose-stay behavior. To understand the sequential patterns of outcome-dependent actions, They designed probabilistic computer algorithms involving specific change actions (i.e., to downgrade or upgrade according to the immediate past outcome): the Win-Downgrade/Lose-Stay (WDLS) or Win-Stay/Lose-Upgrade (WSLU) strategies.[1]

Bryan C. McCanon Research about Rock Paper Scissors in 2007. He found that it is used to resolve conflict when a compromise is not possible. Individuals playing “games” tend to have biases they prefer to play. He showed that Rock Paper Scissors with biased players results in a player with a high valuation for winning being victorious with a probability greater than a player with a low valuation. Thus, it frequently achieves the efficient outcome. Furthermore, as the benefit to victory increases for all players, victory for each player becomes equally likely. Therefore, it is often efficient for minor conflicts while it performs no better than a coin-flip for major conflicts.[2]

The article written by Cate Dunn in 2015 described two different settings for a computer opponent; novice and veteran. A novice computer opponent does not look back at the human's history of previously played moves but instead plays a random move every time. They expected the results of a game with a human versus a novice computer opponent to be relatively equal with the computer winning approximately 50 percent of the time and the human winning approximately 50 percent of the time. For a veteran computer player, the computer would notice the pattern of the human's past moves and would take these into account when playing its next move. They expected the computer to win more and more as more games.[3]

Aman Kharwal in May 12, 2022 created play rock paper scissors, he used if and elif statements in Python. He prepared this game to be played between two players. Player-1 was made the user, and player-2 as the computer where player one manually select the rock paper or scissor, while player two chooses randomly. He used the random module in Python to create this game.[4]

## CHAPTER 3

### SYSTEM ANALYSIS

#### **3.1 Requirement Analysis**

##### **3.1.1 Hardware requirement**

- Computer or Laptop to display the games user interface and running the game.
- Input devices like mouse and keyboard for user to make their selection in game.

##### **3.1.2 Software Requirements**

- Python language
- Integrated Development Environment (IDE) like Visual studio.
- Python library Like Tkinter for GUI,Random to generate random numbers,pygame for sound effects.
- Voice changer
- Operating system like windows, MacOs for coding and running game.

#### **3.2 Feasibility Analysis**

A feasibility study is conducted to determine the viability and practicality of a strategy, plan, or design. It assesses whether the project is achievable and makes sense to pursue. The feasibility study takes into account various factors, including economic, technical, and so on. The analysis covers the following areas:

##### **3.2.1 Technical Feasibility**

All the essential resources like Tkinter, Python, and VS Code, required for game development and maintenance, are readily accessible. It can run seamlessly on

desktop computers and laptops regardless of the operating system they use. The game has user friendly interface, is responsive and have seamless gameplay. Hence, it is technically feasible.

### **3.2.2 Economic Feasibility**

The game has been developed utilizing readily available resources. Hardware components, such as computers and internet connections, necessary for the development process, are easily obtainable in the market. Software tools like VS Code are also freely accessible. Both Seekers and Providers can utilize the system without any financial obligations. It is accessible to anyone with a standard device capable of running a game . Considering these factors, the game is feasible and accessible to users without any financial barriers.

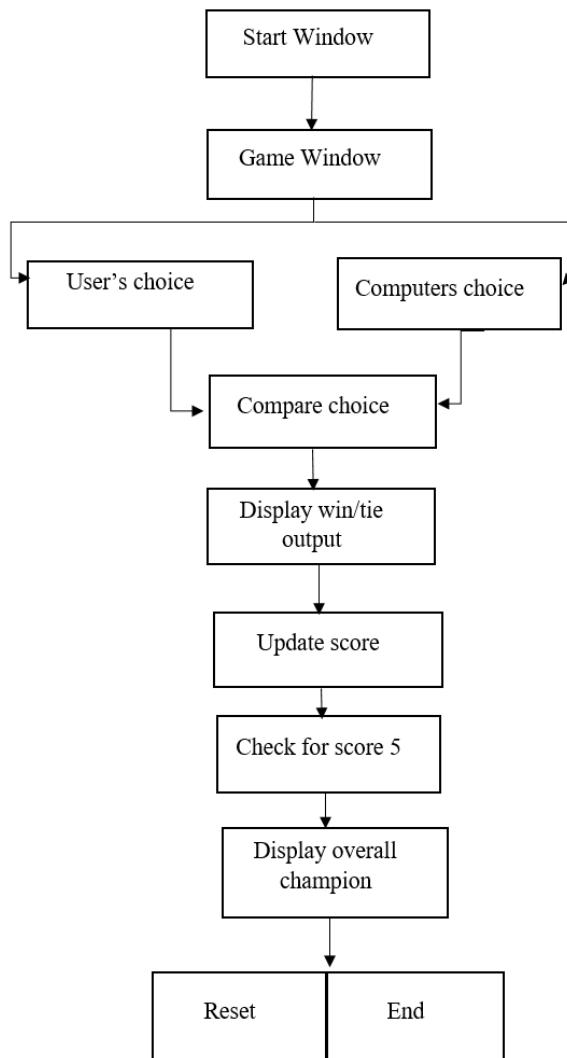
### **3.2.3 Legal Feasibility**

The game does not violate any of the laws and regulations. So it is legally feasible.

## CHAPTER 4

### METHODOLOGY

#### 4.1 System Block Diagram



**Figure 4.1:** Block diagram of the system

## 4.2 Flow chart

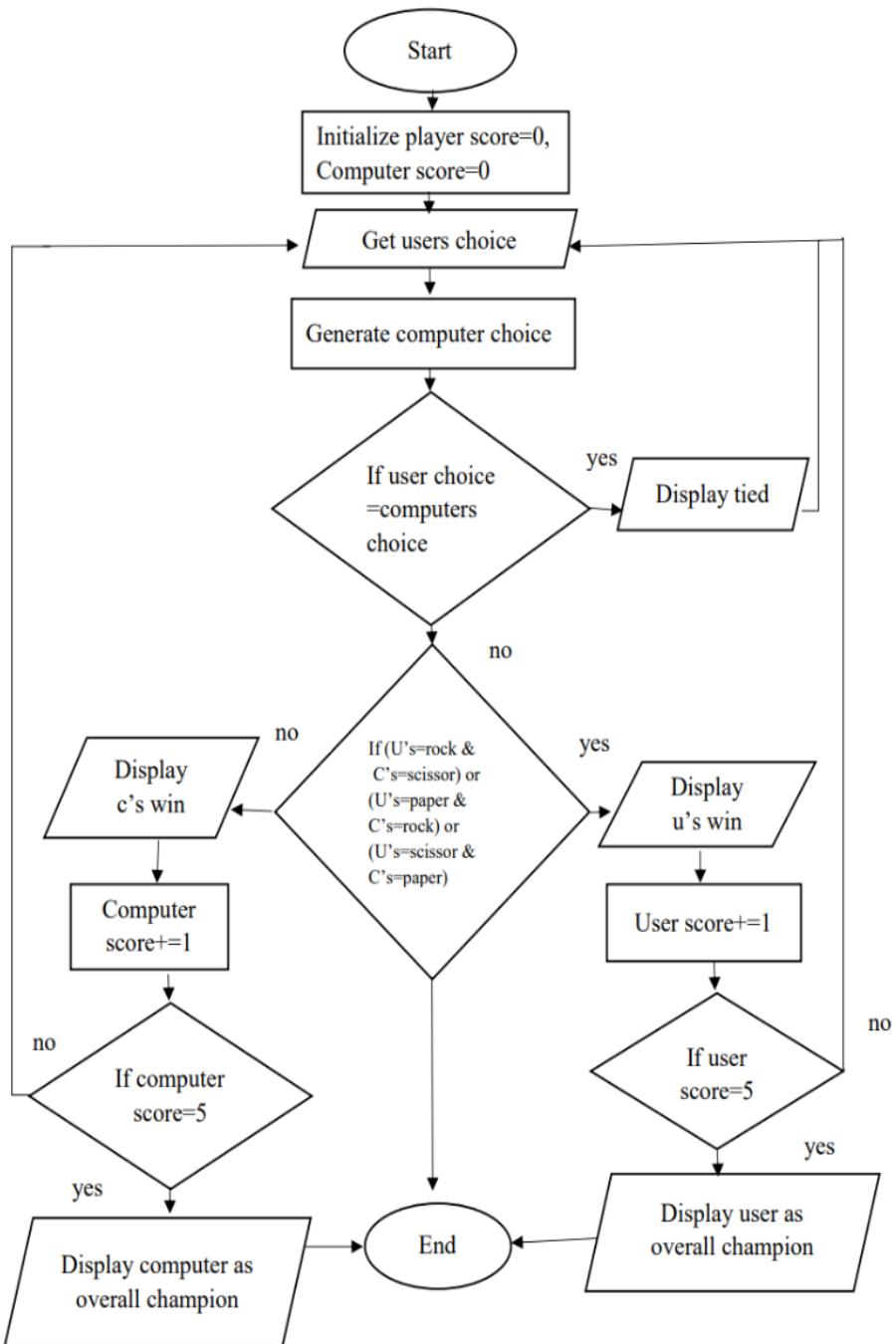


Figure 4.2: Flow chart of game

## CHAPTER 5

### IMPLEMENTATION DETAIL

Our project is a simple GUI-based game of rock-paper-scissors between the user and the computer made using python programming language. Here are the implementation details of the project:

#### **5.1 Importing Modules**

- The tkinter module is imported to create the GUI for the game,
- The messagebox module from tkinter is imported to display message boxes for game results and quitting confirmation
- The Image and ImageTk modules from PIL are imported to handle image manipulation and display.
- The random module is imported to generate random choices for the computer.
- The pygame module is imported to handle sound effects

#### **5.2 Global variables**

- The variables playerscore and computerscore are declared to store the scores of the player and the computer.

#### **5.3 Options tuple**

- The options tuple is defined, which consists of three tuples representing the choices: Rock, Paper, and Scissors. Each tuple contains the choice name and a corresponding number

## 5.4 Opengame function()

- This function is called when the game starts. It sets up the game window and initializes the game elements.
- The startroot window is destroyed, and a new root window is created for the game.
- The game background image is loaded and displayed as a label.
- Various labels, buttons, and frames are created to display the game elements such as headings, player options, game buttons, player and computer choices, scores, reset button, and quit button.
- The necessary images for the game (blank, rock, paper, and scissor) are loaded and resized using the PIL library.
- The game buttons are assigned commands that call the playerchoice() function with the corresponding choice from the options tuple.
- The resetgame() function is assigned to the reset button to reset the game.
- The endgame() function is assigned to the quit button to confirm quitting the game

## 5.5 Playerchoice(playerinput)

- This function is called when the player selects a choice (rock, paper, or scissors).
- The function receives the playerinput as a tuple from the options tuple.
- The playerchoicelabel and computerchoicelabel are updated with the player's and computer's choices, respectively.
- The game images for the player and computer choices are updated based on their selections using the updateimages() function.
- If the player and computer choices are the same, it's a tie.
- If the difference between the player's and computer's choices modulo 3 is 1 then player wins. The player's score is incremented.

- If the above conditions are not met, the computer wins. The computer's score is incremented, and the winner label is updated.
- If the player or the computer reaches a score of 5, a message box is displayed declaring the overall champion of the game.
- The disablebuttons() function is called to disable the game buttons when the game is over.

## 5.6 Other Helper Functions

- getcomputerchoice(): This function returns a random choice from the options tuple.
- disablebuttons(): This function disables the game buttons when the game is over.
- resetgame(): This function resets the game by resetting the player and computer scores to 0 and updating the necessary labels and images.
- updateimages(playerchoice, computerchoice): This function updates the game images based on the player and computer choices.

## 5.7 Start window

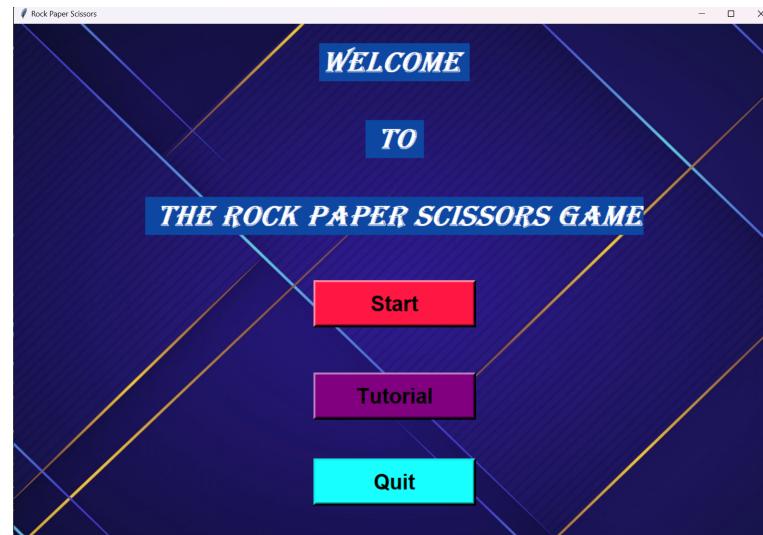
- The startroot window is created using the Tk class from tkinter.
- The game background image is loaded and displayed as a label.
- Buttons for starting the game, opening the tutorial, and quitting the game are created and placed on the window.
- The start window is run using the mainloop() method.

## 5.8 Additional function

- playclicksound(): This function uses the Pygame module to play a click sound when a button is clicked.
- opentutorial(): This function opens a new window to display the game tutorial.

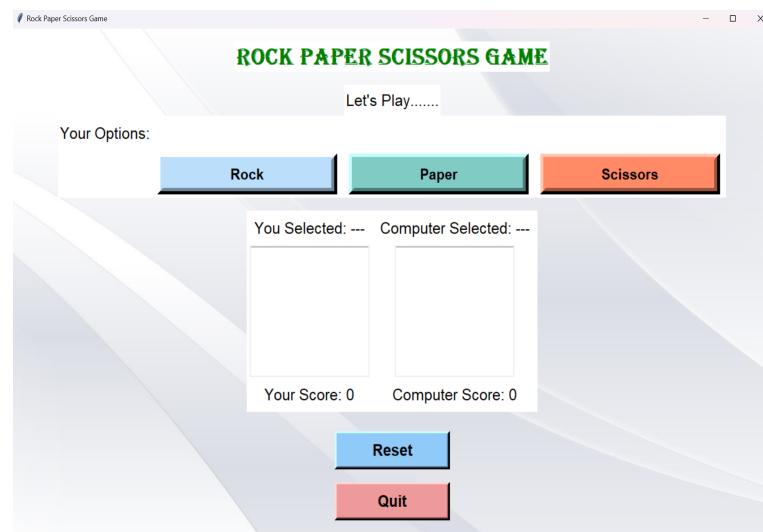
## CHAPTER 6

### RESULT AND ANALYSIS



**Figure 6.1:** GUI of starting window of RPS game

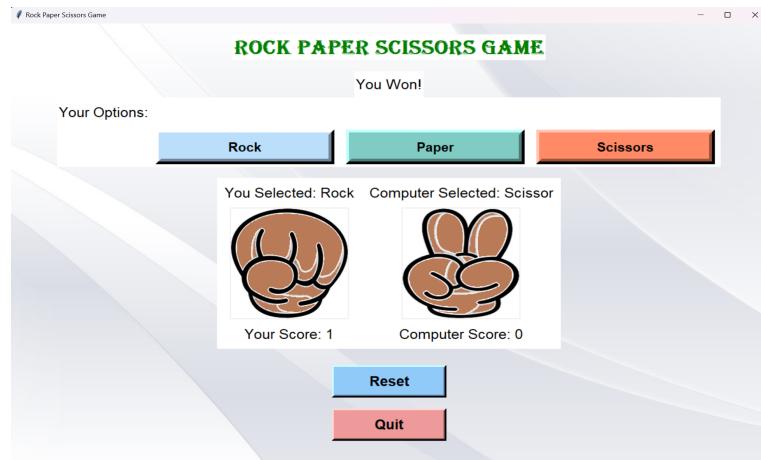
The figure 5.2 shows the simple GUI of starting window of our game. It includes the welcome title, start button for starting the game that opens main game window, tutorial button for game help and end button for ending the window.



**Figure 6.2:** GUI of main game window of our Rock Paper Scissor game

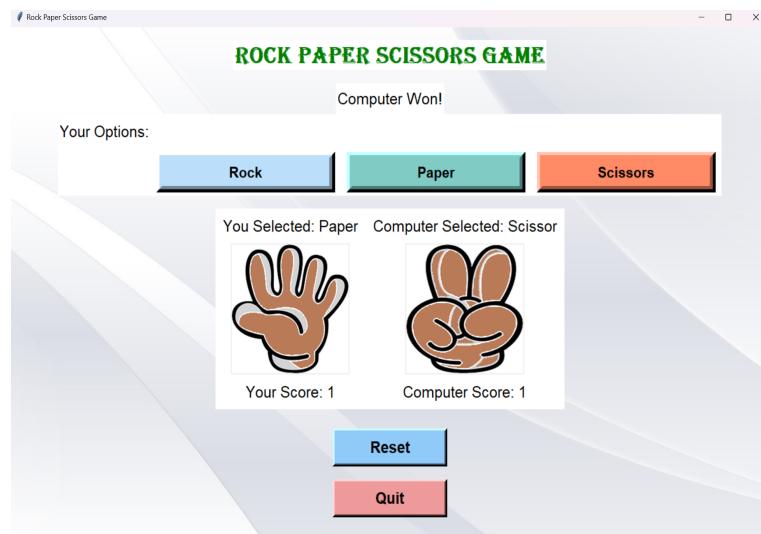
Similarly, Figure 5.2 shows the GUI of main game window of our Rock Paper Scissor

game where user play the game. It contains the title of the game, button for users choice from three option rock, paper scissor. Also contain the label for user and computer selection with their corresponding scores and label for updating images from three options for every selection of computer and user. It also has the end and reset button for ending and resetting the game.



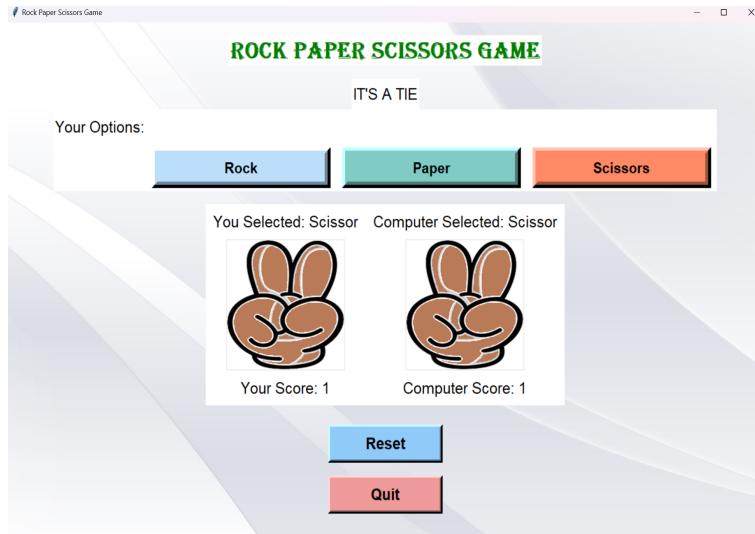
**Figure 6.3:** Displaying users win

In the figure 5.3, the result of the round as users win is displayed as user chose the option that beat computer choice according to game rules or logic.



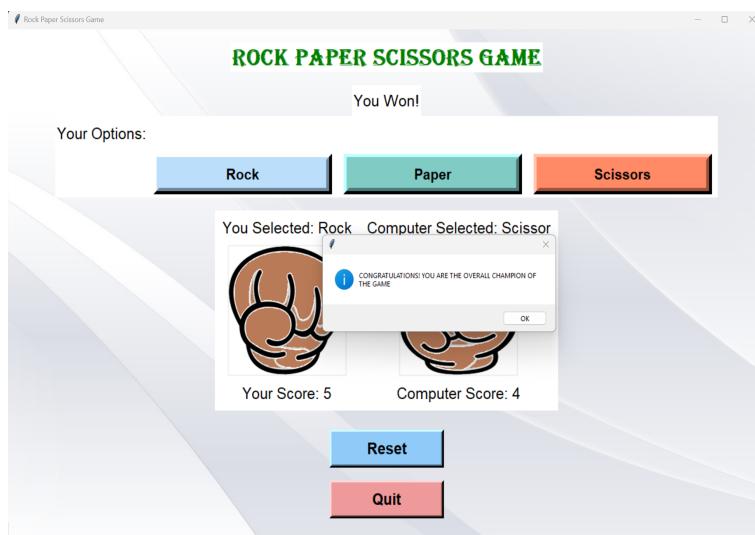
**Figure 6.4:** Displaying computers win

In the figure 5.4, the result of the round as computers win is displayed as user chose the option that cannot beat computer choice according to game rules or logic.



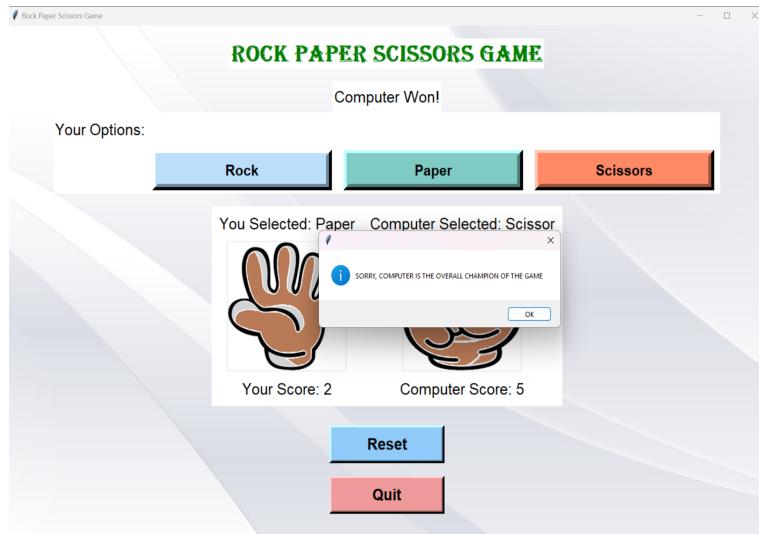
**Figure 6.5:** Displaying tie

In the figure 5.5, the result of the round as tie is displayed as user and computers choice is same.In this both have chosen the same option that is scissors.Also the there is no update of score when game is tied.



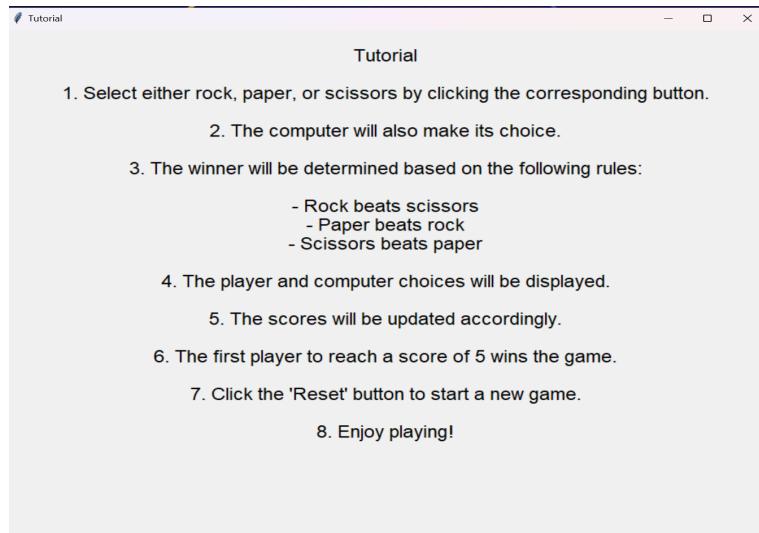
**Figure 6.6:** Displaying user as overall champion of the game

In the figure 5.6,there is appearance of small messagebox that display user as overall champion of the game as user has scored five first after winnng five game round .



**Figure 6.7:** Displaying computer as overall champion of the game

In the figure 5.7, there is appearance of small messagebox that display computer as overall champion of the game as user has scored five first after winning five game round .



**Figure 6.8:** Tutorial of the game

The figure 5.7 displays tutorial of the game.It is displayed when the user clicks on the tutorial button in starting window.It helps user to know about how game is played.

## CHAPTER 7

### Work Schedule

The working schedule for this project is as follows:

Task	Jan	Feb	March	April	May	June
Planning	■					
Game analysis		■	■			
Game design			■	■		
Coding				■	■	
Testing and debugging				■	■	■
Documentation	■	■	■			

**Table 7.1:** Time Schedule for the Project Work

## CHAPTER 8

### CONCLUSION AND FUTURE WORK

#### **8.1 Conclusion**

Overall, Our Rock Paper Scissors game is digital version played with computer that is a simple yet engaging game made using Python language and its various modules and libraries. Our project demonstrates the understanding of GUI programming, event handling, and sound integration, image processing making it a valuable learning experience in game development with Python. Our game contains features like score system, images and sound effects.

#### **8.2 Future Work**

As our game is simple offline desktop game played against AI. But in Future we are looking forward TO make it multiplayer game played between two players with more additional features like time mode, different levels like difficulty, easy and normal. Also making the GUI more attractive than now with animations.

## REFERENCES

- [1] F. . G. C. Zhang, Hansu ; Moisan, “Rock-paper-scissors play: Beyond the win-stay/lose-change strategy,” *Games* 12, vol. 52, p. 3, 2021.
- [2] B. Mccannon, “Rock paper scissors,” *Journal of Economics*, pp. 67–88, 2007.
- [3] C. Dunn, “Rock paper scissor project,” 2015.
- [4] A. Kharwal, “Rock paper scissors game using python,” 2022.