



PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

100 Feet Ring Road, BSK III Stage, Bengaluru-560 085

Department of Computer Science & Engineering

ROCK-PAPER-SCISSORS GAME

(USING PYTHON)

TEAM MEMBERS

BONTHA SREENIVASA REDDY-PES1UG21CS142

NAGIREDDY SIVA KRISHNA REDDY-PES1UG21EC156

ROCK-PAPER-SCISSORS GAME

ABSTRACT :

The Rock-Paper-Scissors game is a single player interactive game made using Tkinter , GUI library for Python. This program hides the logic and the output in the form of images , text and scores are shown .

There are a total of 9 testcases in game depending upon the choice of player . The result of the game depends on these cases . Test-cases are written in the program in the form of loops ,then result is decided ,scores are shown on the window. Using this program we can play , record score and also reset the game if needed.

TABLE OF CONTENTS:

S.NO	TOPIC	PAGE NO.
1	INTRODUCTION	Page 3
2	DESIGN AND IMPLEMENTATION	Page 4
3	TESTING	Page 10
4	RESULT AND ANALYSIS	Page 11
5	CONCLUSION AND FUTURE ENHANCEMENTS	Page 21
6	REFERENCES	Page 22

INTRODUCTION:

The Rock Paper Scissors game is a well known game played from kids to teens for fun. It is a simple game involving 3 choices , a minimum of 2 players to play the game. This is often used as a fair choosing method between two people, similar to coin flipping or throwing dice in order to settle a dispute or make an unbiased group decision. Unlike truly random selection methods, however, rock paper scissors can be played with a degree of skill by recognizing and exploiting non-random behavior in opponents.

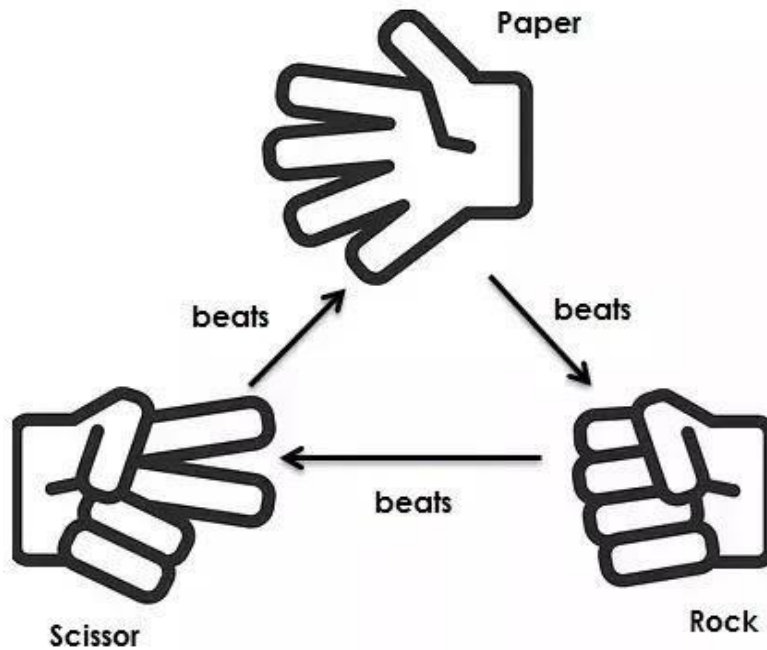
The game need 2 players to play but using our program a single player can compete with the computer and keep record of his scores regardless of having a single-player also. The program is similar to the classic rock paper scissors game except the game is played against the computer with a random odds of winning , losing or having a tie game. This program have these following features:

- **Playing the game against computer**
- **Keeping the record of score**
- **Reset the game**

Provided the user to select his choice in the game window.

DESIGN AND IMPLEMENTATION:

The simple logic involving in this game is depicted below:



Code for the program :

```
#importing required modules
```

```
from tkinter import *
```

```
from PIL import Image, ImageTk
```

```
from random import randint
```

```
# initiating main window
```

```
root = Tk()
```

```
root.title("Rock Scissor Paper")
```

```
root.geometry("860x450")
root.resizable(False, False)

# storing the pictures

rock_img = ImageTk.PhotoImage(Image.open("rock-user.png")) paper_img =
ImageTk.PhotoImage(Image.open("paper-user.png")) scissor_img =
ImageTk.PhotoImage(Image.open("scissors-user.png")) rock_img_comp =
ImageTk.PhotoImage(Image.open("rock.png")) paper_img_comp =
ImageTk.PhotoImage(Image.open("paper.png")) scissor_img_comp =
ImageTk.PhotoImage(Image.open("scissors.png"))

# labeling the initial picture

user_label = Label(root, image=scissor_img) comp_label
= Label(root, image=scissor_img_comp)
comp_label.grid(row=1, column=0)
user_label.grid(row=1, column=4)

# initiating scores

playerScore = Label(root, text=0, font=100, fg="black")
computerScore = Label(root, text=0, font=100, fg="black")
computerScore.grid(row=1, column=1) playerScore.grid(row=1,
column=3)

# player indicators

user_indicator = Label(root, font=50, text="USER", fg="black")
comp_indicator = Label(root, font=50, text="COMPUTER",
```

```
        fg="black")
user_indicator.grid(row=0, column=3)

comp_indicator.grid(row=0, column=1)

# messages
msg = Label(root, font=50, fg="black")
msg.grid(row=3, column=2)

# update message

def updateMessage(x):
    msg['text'] = x

# update user score

def updateUserScore():
    score = playerScore["text"]

    score += 1
    playerScore["text"] = score

# update computer score

def updateCompScore():
    score = (computerScore["text"])

    score += 1
    computerScore["text"] = score

# check winner
```

```
def checkWin(player, computer): if
    player == computer:
        updateMessage("Its a tie!!! 😊")
    elif player == "rock":
        if computer == "paper":
            updateMessage("You loose 😞")
            updateCompScore()
        else:
            updateMessage("You Win 😊")
            updateUserScore()
    elif player == "paper":
        if computer == "scissor":
            updateMessage("You loose 😞")
            updateCompScore()
        else:
            updateMessage("You Win 😊")
            updateUserScore()
    elif player == "scissor":
        if computer == "rock": updateMessage("You
            loose 😞 ") updateCompScore()
        else:
            updateMessage("You Win 😊")
            updateUserScore()

    else:
        pass

# update choices
```



```
choices = ["rock", "paper", "scissor"]
```

```
def updateChoice(x):
```

```
    # for computer
```

```
    compChoice = choices[randint(0, 2)] if
```

```
    compChoice == "rock":
```

```
        comp_label.configure(image=rock_img_comp)
```

```
    elif compChoice == "paper":
```

```
        comp_label.configure(image=paper_img_comp)
```

```
    else:
```

```
        comp_label.configure(image=scissor_img_comp)
```

```
# for user
```

```
    if x == "rock":
```

```
        user_label.configure(image=rock_img)
```

```
    elif x == "paper":
```

```
        user_label.configure(image=paper_img)
```

```
    else:
```

```
        user_label.configure(image=scissor_img)
```

```
    checkWin(x, compChoice)
```

```
#reset game
```

```
def reset_scores():
```

```
    playerScore.configure(text=0)
```

```
    computerScore.configure(text=0)
```

```
    msg.configure(text="")
```

```
user_label.configure(image=scissor_img)
comp_label.configure(image=scissor_img_comp)

# buttons

rock = Button(root, width=20, height=2, text="ROCK",
               bg="#FF3E4D", fg="black", command=lambda:
updateChoice("rock")).grid(row=2, column=1)
paper = Button(root, width=20, height=2, text="PAPER",
               bg="#FAD02E", fg="black", command=lambda:
updateChoice("paper")).grid(row=2, column=2)
scissor = Button(root, width=20, height=2, text="SCISSOR",
                 bg="#0ABDE3", fg="black", command=lambda:
updateChoice("scissor")).grid(row=2, column=3)

reset=Button(root,text="Reset Game",command=reset_scores,font = 10, fg
= "red").grid(row=4,column=0)

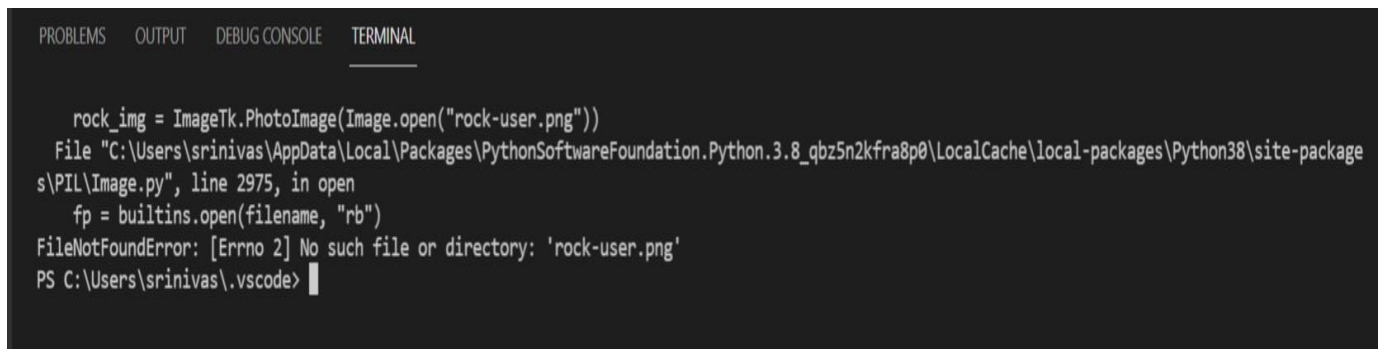
root.mainloop()
```

TESTING :

In this program we can play Rock-Paper-Scissors game using python. The program allows the player/user to play the game against the computer and keep a record of the scores , given the user selects his choice in the game window. This program consists of 4 buttons and a message box which gives us the result of the game ,after which scores are stored .

This program uses three modules Random , Tkinter and pillow(PIL) to run the code and to display the images.

NOTE: When the code and images are not present in the same folder/ destination the program will throw an error stating ‘rock-user.png ’ not found. So we have to take care while executing this program.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
rock_img = ImageTk.PhotoImage(Image.open("rock-user.png"))
File "C:\Users\srinivas\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-packages\Python38\site-packages\PIL\image.py", line 2975, in open
  fp = builtins.open(filename, "rb")
FileNotFoundError: [Errno 2] No such file or directory: 'rock-user.png'
PS C:\Users\srinivas\.vscode> |
```

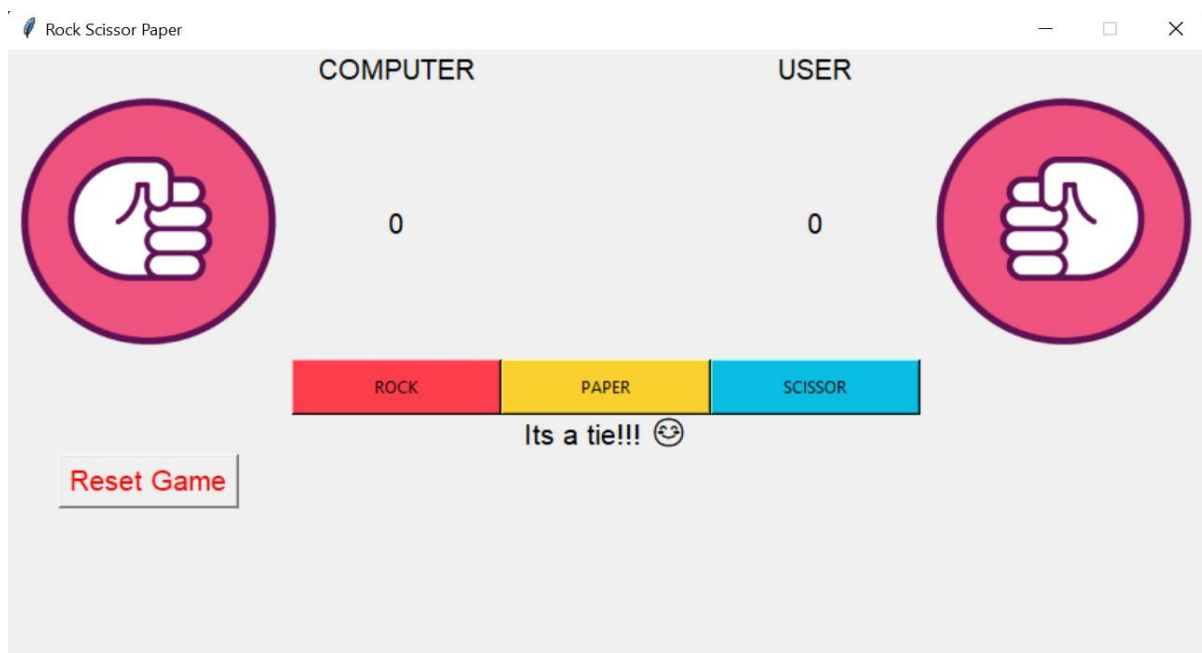
- Error while executing

RESULT AND ANALYSIS:

As discussed earlier there are a total of 9 possibilities in which the game's result can end with winning, losing or getting a tie. Here we will look at all the 9 test cases and see how our program reacts to them.

1. The user chooses Rock then there are three cases :

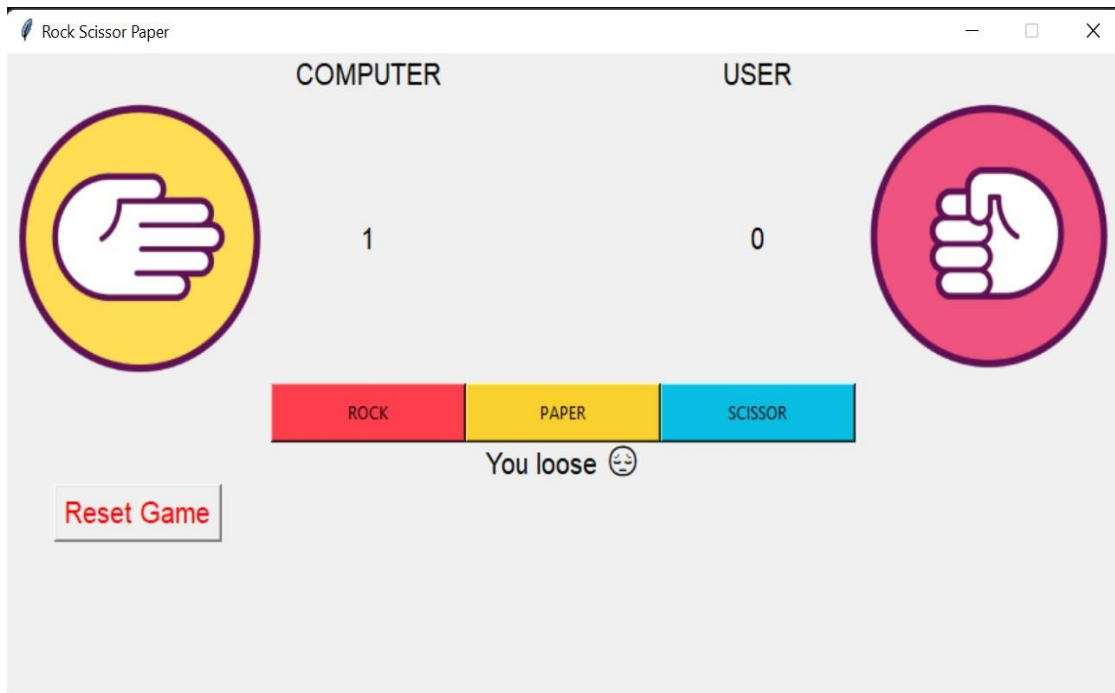
a. The user chooses rock and the computer chooses rock:



The program reacts to this case in the following ways:

- The game is a tie because both user and computer chose same choice.
- And the program gives the result as a tie by giving the message as "It's a tie 🤐"
- And updates the score to 0 on both sides.

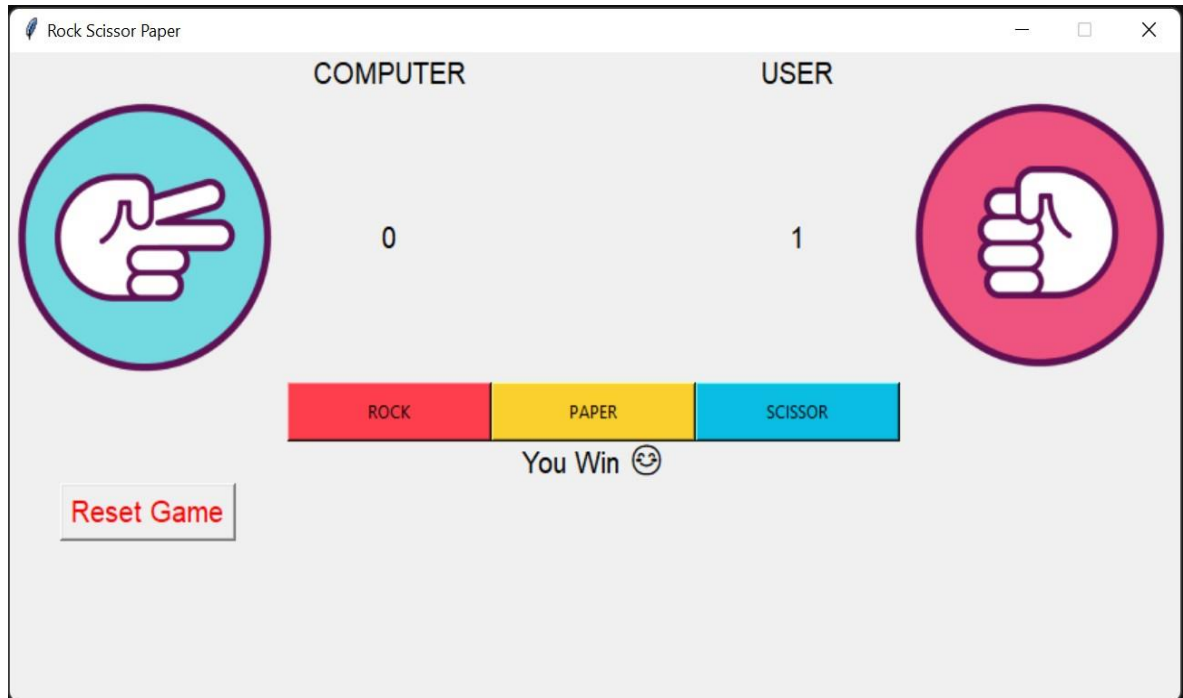
b. The user chooses rock and the computer chooses paper:



The program reacts to this case in the following ways:

- The user lost because the user chose rock and computer chose paper.
- And the program gives the result as you lost by giving the message as “You loose 😞”
- And updates the score to 0 on user’s side and to 1 on computer’s side.

c. The user chooses rock and the computer chooses scissors:

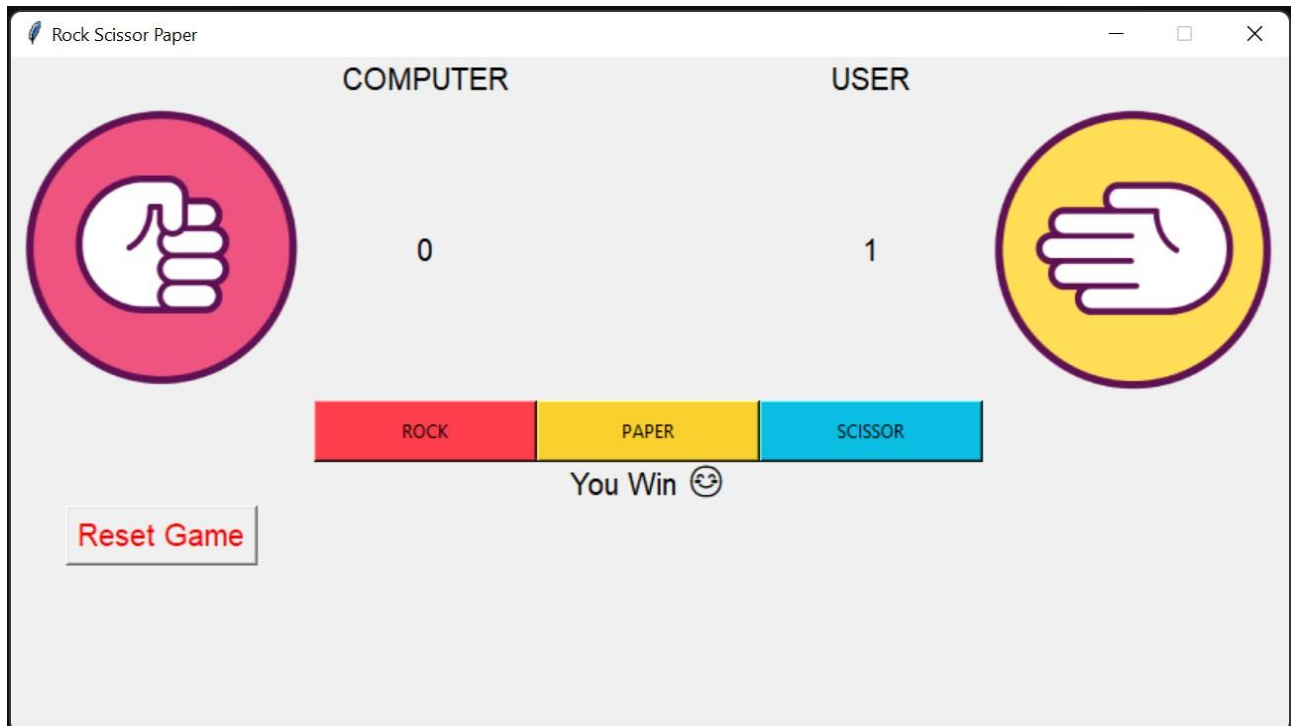


The program reacts to this case in the following ways:

- The user wins as he chose Rock and the computer chose Paper.
- And the program gives the result as a win by giving a message "You Win 😊 "
- And updates the score to 1 on user's side and to 0 on computer's side.

1. The user chooses Paper then there are three cases :

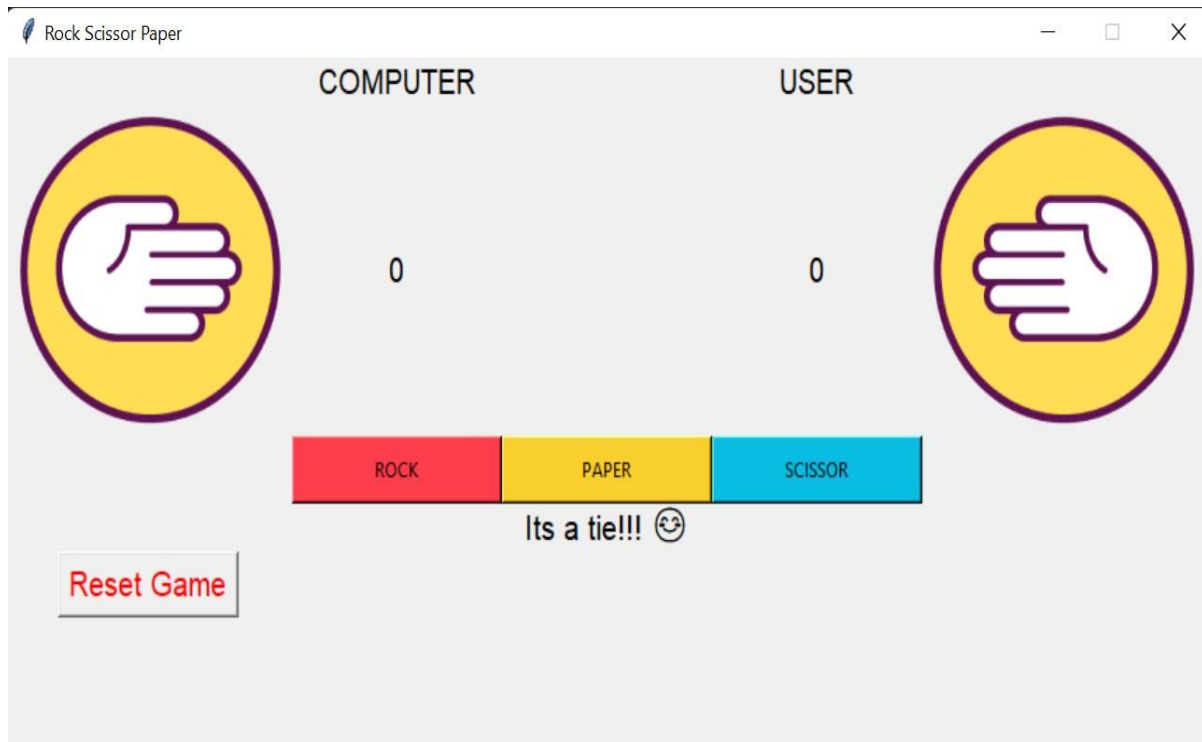
a. The user chooses Paper and the computer chooses Rock:



The program reacts to this case in the following ways:

- The user wins as he chose Paper and the computer chose Rock.
- And the program gives the result as a win by giving a message "You Win 😊"
- And updates the score to 1 on user's side and to 0 on computer's side.

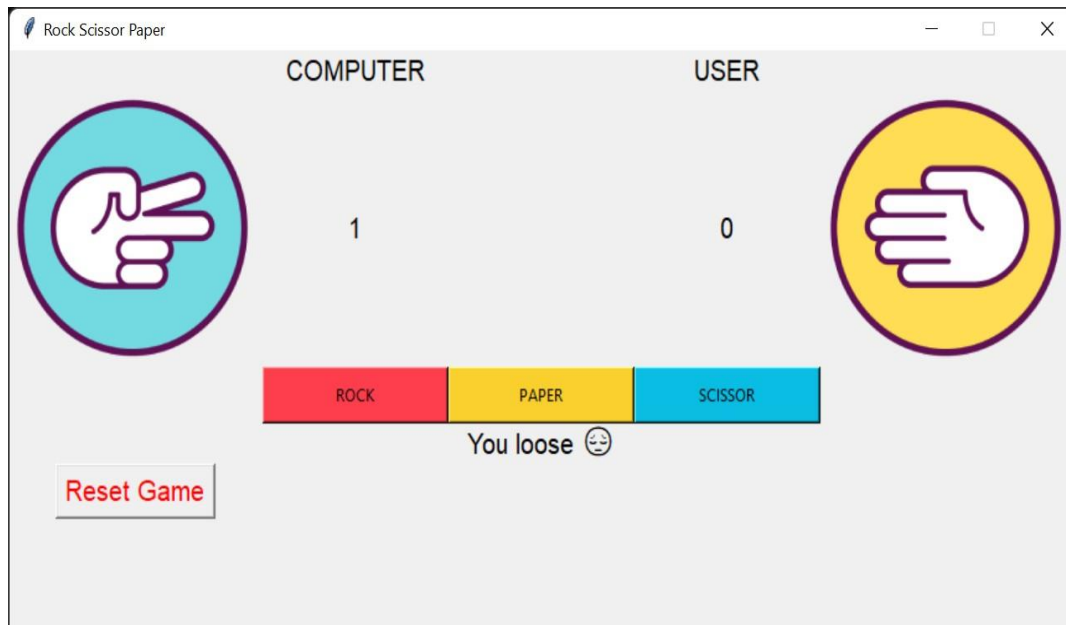
b. The user chooses Paper and the computer chooses paper:



The program reacts to this case in the following ways:

- The game is a tie because both user and computer chose same choice.
- And the program gives the result as a tie by giving the message as “It’s a tie 😊 ”
- And updates the score to 0 on both sides.

c. The user chooses Paper and the computer chooses Scissors:

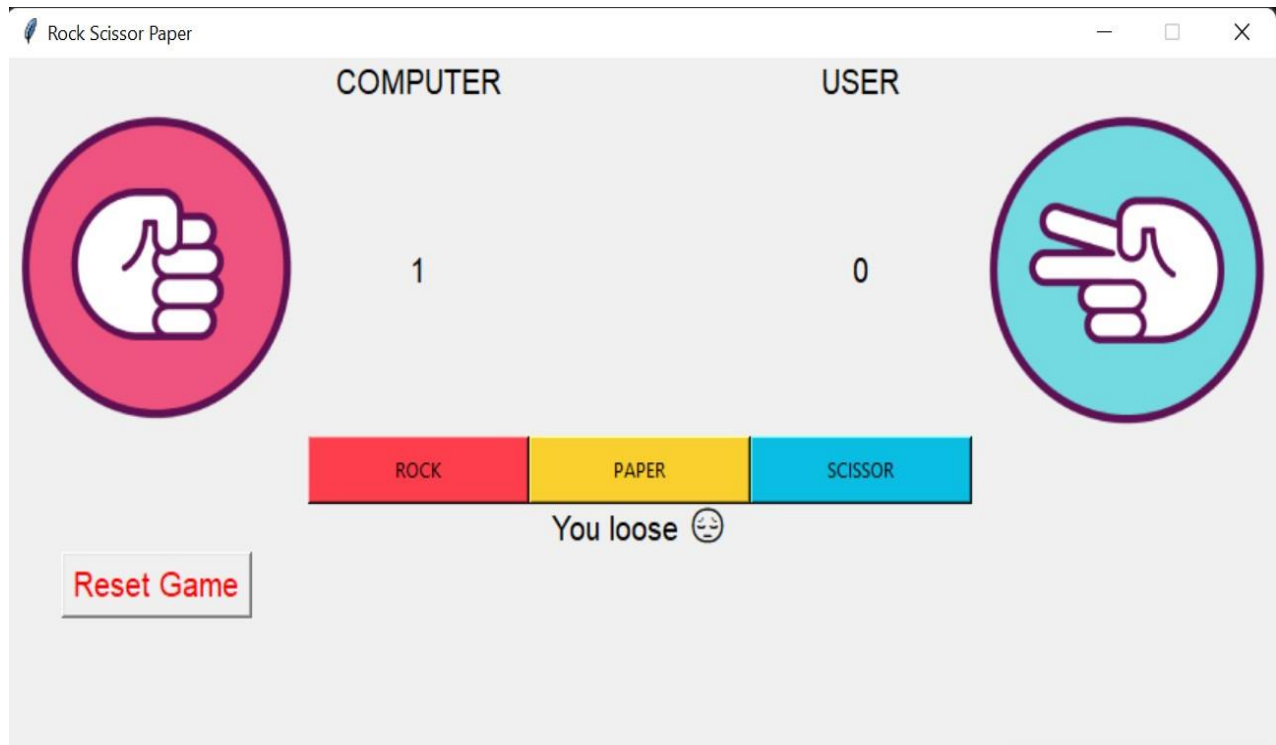


The program reacts to this case in the following ways:

- The user lost because the user chose Paper and computer chose Scissors.
- And the program gives the result as you lost by giving the message as “You loose 😞 ”
- And updates the score to 0 on user’s side and to 1 on computer’s side.

3. The player chooses Scissors then there are 3 cases:

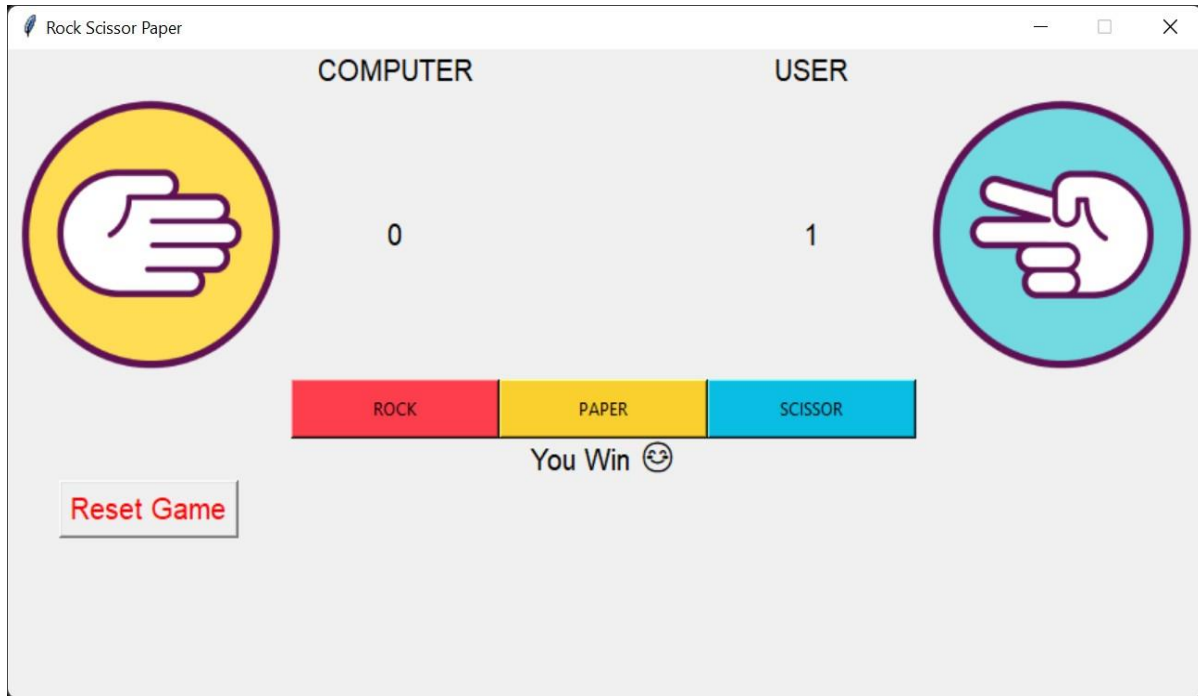
a. The user chooses Scissors and the computer chooses Rock:



The program reacts to this case in the following ways:

- The user lost because the user chose Scissors and computer chose Rock.
- And the program gives the result as you lost by giving the message as “You loose 😞”
- And updates the score to 0 on user’s side and to 1 on computer’s side.

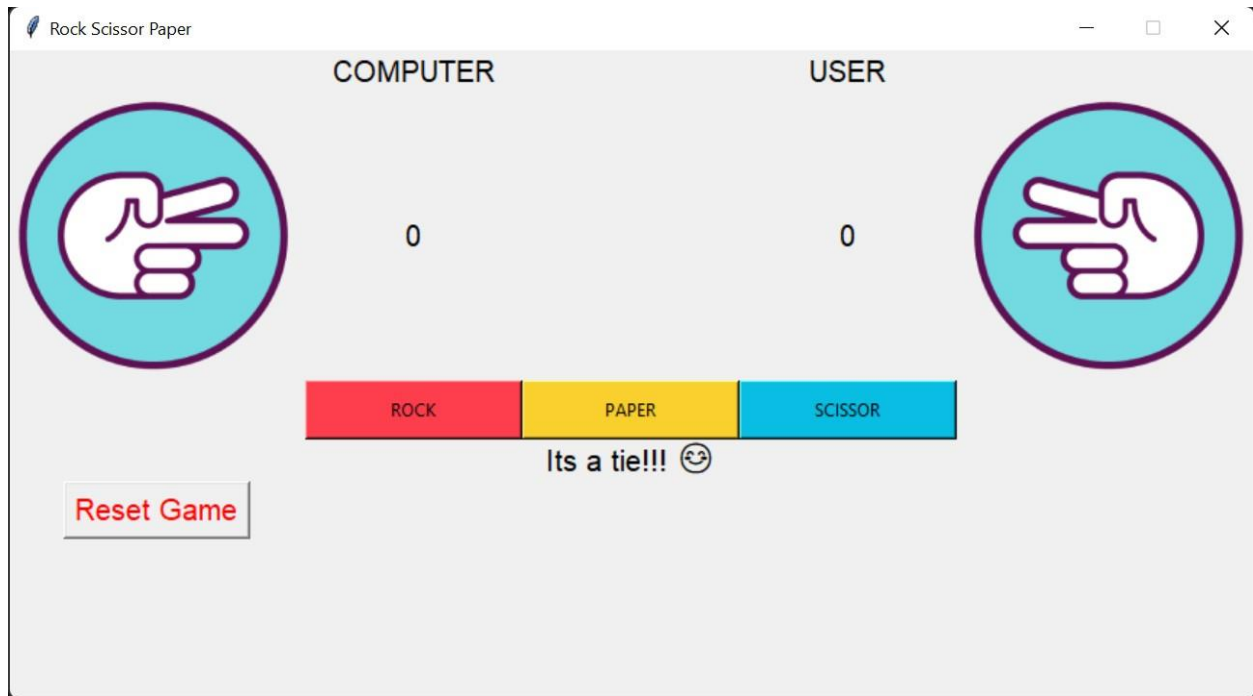
b. The user chooses Scissors and the computer chooses paper:



The program reacts to this case in the following ways:

- The user wins as he chose Rock and the computer chose Paper.
- And the program gives the result as a win by giving a message "You Win 😊"
- And updates the score to 1 on user's side and to 0 on computer's side.

c. The user chooses Scissors and the computer chooses Scissors:

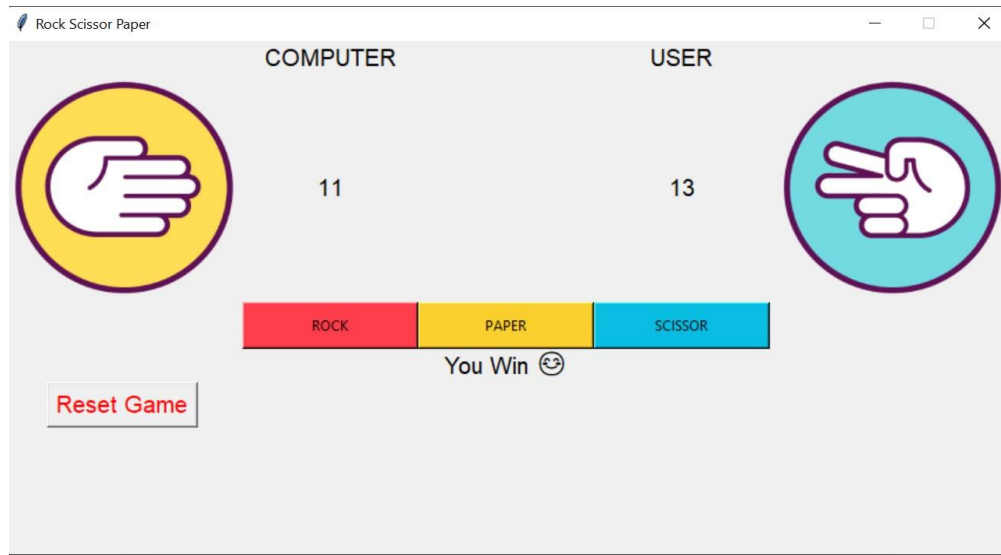


The program reacts to this case in the following ways:

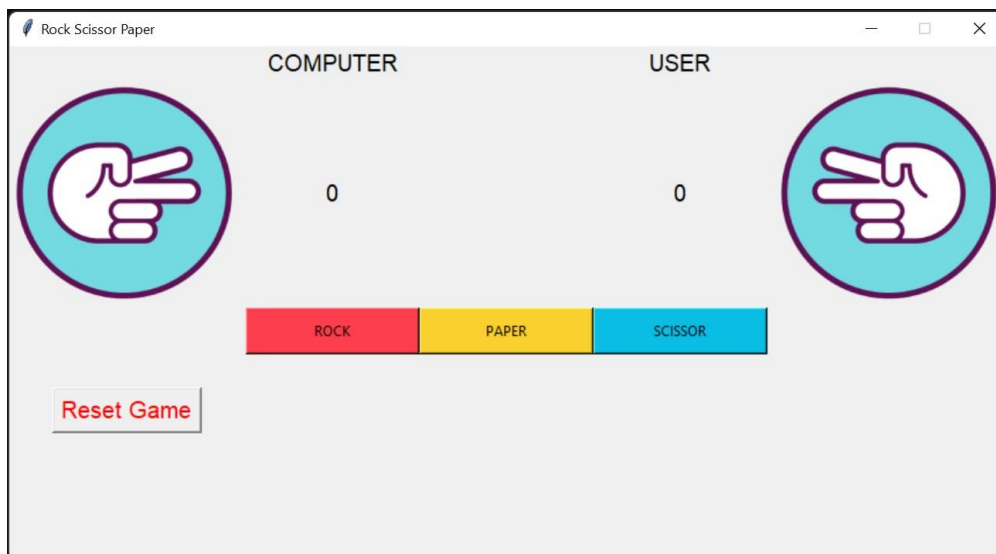
- The game is a tie because both user and computer chose same choice.
- And the program gives the result as a tie by giving the message as "It's a tie 🤝"
- And updates the score to 0 on both sides.

After playing for a while the player the reset the game and the scores using reset button provided on the bottom left side of the window.

Before reset:



After reset:



Analysis:

The rock paper scissors game is a simple program involving a single player , and shows the user the result and choices in the form of images and emojis in the program window without involving the logic behind the program in the window. A simple program involving less than 160 lines of code , used to play the Rock-Paper-Scissors Game digitally without the need of second player .

CONCLUSIONS AND FUTURE ENHANCEMENTS:

This is all the functionality of the code .This code helps us to play a simple game called as rock paper scissors against cpu (computer) in a digital way. This concludes the ROCK-PAPER-SCISSORS GAME project.

FUTURE ENHANCEMENTS:

- **Adding one player and two player modes .**
- **Showing leaderboards among the players.**
- **Adding start menu at the start and a dialogue box when about to close.**
- **Graphics may be added instead of images.**

REFERENCES:

- 1.Information for tkinter: <https://docs.python.org/3/library/tkinter.html>
- 2.For PIL module: <https://pypi.org/project/Pillow/>
- 3.Opening image in PIL module: <https://www.geeksforgeeks.org/python-pillow-a-fork-of-pil/>
- 4.For picking color codes: https://www.w3schools.com/colors/colors_picker.asp