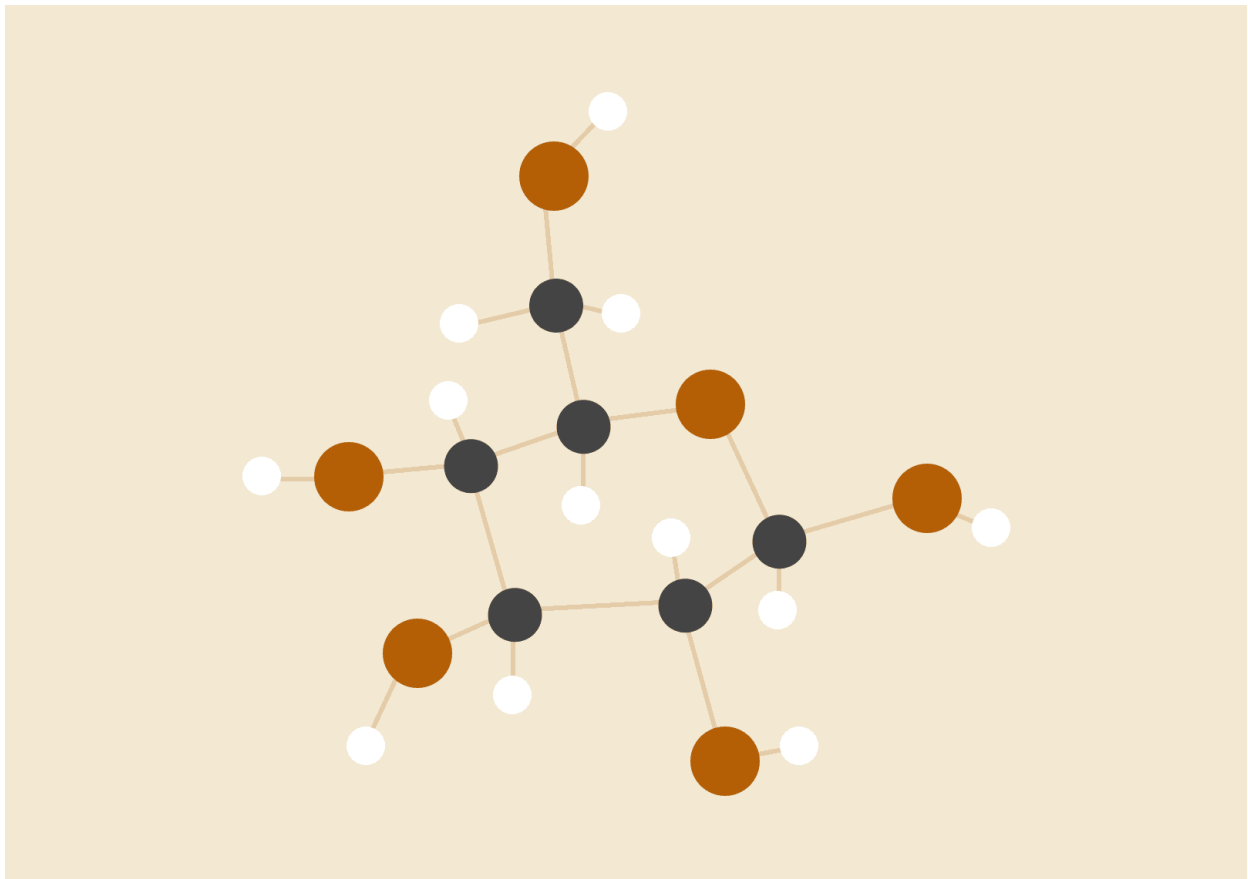


Mini project Report for 18ECO109J

(Embedded System DesignRaspberry Pi)

Rock paper scissors game using GUI



BACKGROUND and MOTIVATION

Rock paper scissors game is also known as stone paper scissors. It is a hand game that is usually played between 2 people, each player can randomly form any one of three from their hand.

A player who chooses rock will win by another player who chooses scissors but lose by the player who chooses paper; a player with paper will lose by the player with the scissors.

If both players choose the same then the game is tied. Rock paper scissors is mainly played among kids.

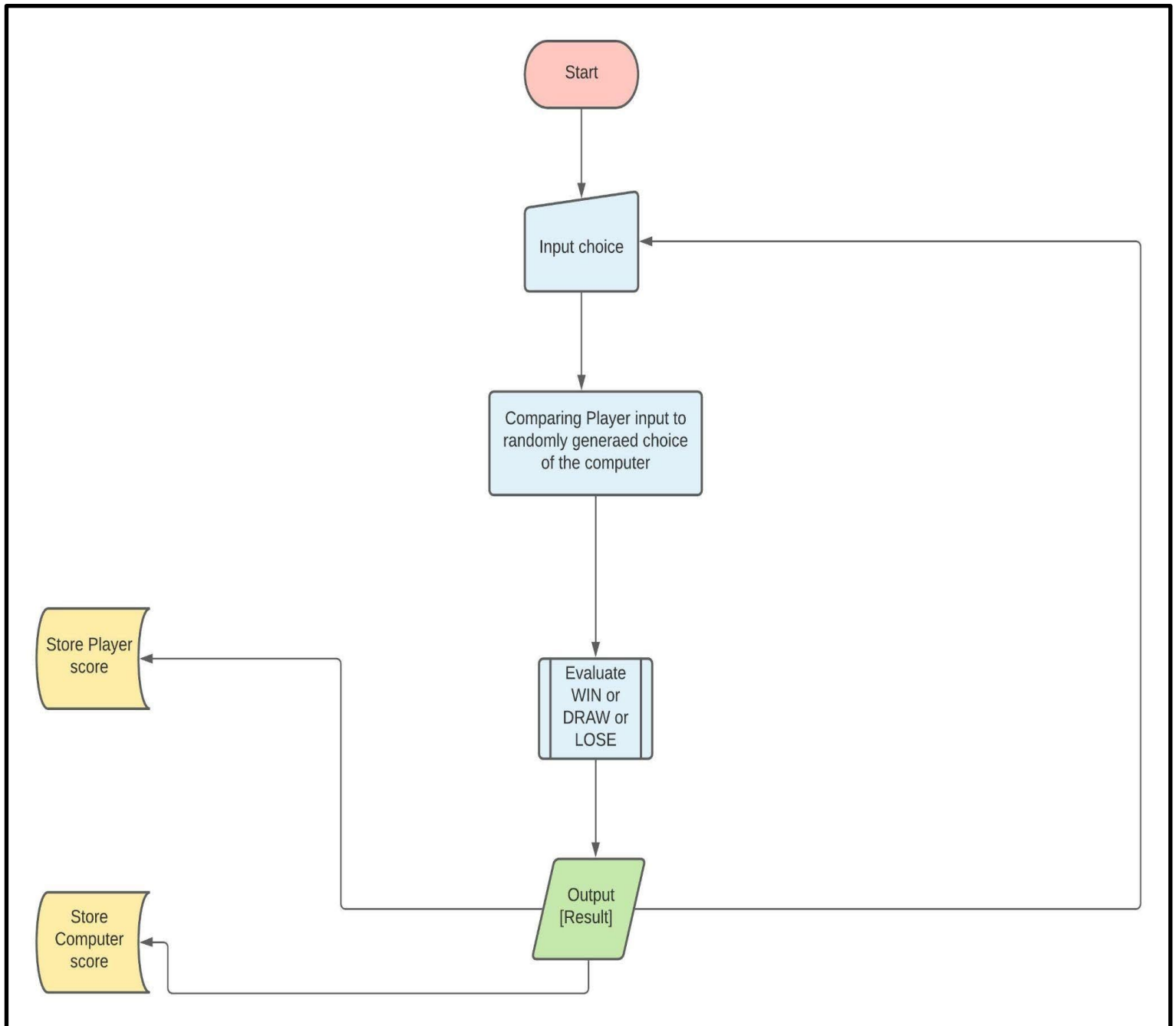
KEY PROBLEMS SOLVED

- A quick stress reliever for anyone
- A quick game interactive game for kids
- One beautiful thing about the Rock Paper Scissors game is that you can improve your wellbeing and lifestyle by merely playing the game
- The participants of the game can decide to use the game to fulfill a particular task. The following are a few examples of ways the game can be made fun and satisfy the original purpose of the game:
 - Used between siblings to decide who clears the table or trims the garden. Colleagues can also play the game to decide who pays the bill for lunch.
- Addition and substitution of throws to extend the way the game is being played

OBJECTIVE

- The objective is to make a rock-paper-scissors python project for single player that plays against computers, in Python using GUI.
- The game must have interactive buttons and a score counter.

BLOCK DIAGRAM



METHODOLOGY

Steps:

- Import required libraries
- Initialize window
- Code for user choice
- Code for computer choice
- Define functions
- Define buttons

updatechoice() function:

```
choices = ["rock", "paper", "scissor"]  
  
def updateChoice(x):  
    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)  
  
    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)
```

- Define the choices
- Randomize the choice from computer.

- We define a variable x for the input from the user.
- We pass the user's input and random choice from the computer to a defined function to compare win/loss.

checkwin() function:

```
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
```

- The function compares various inputs and decides win loss accordingly.
-
- If both inputs are same then, the program declares it a draw.
-
- Win criteria is:
 - ROCK>>SCISSOR

- SCISSOR>>PAPER
- PAPER>>ROCK

PROGRAM

```
from tkinter import *

from PIL import Image, ImageTk

from random import randint

# main window

root = Tk()

root.title("Rock Scissor Paper")

root.configure(background="#96EE77")

# picture

rock_img = ImageTk.PhotoImage(Image.open("rock-u.png"))

paper_img = ImageTk.PhotoImage(Image.open("paper-u.png"))

scissor_img = ImageTk.PhotoImage(Image.open("scissor-u.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"))
```

insert picture

user_label = Label(root, image=scissor_img, bg="#96EE77")

comp_label = Label(root, image=scissor_img_comp, bg="#96EE77")

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

user_label.grid(row=1, column=4)

scores

playerScore = Label(root, text=0, font=100, bg="#96EE77", fg="black")

computerScore = Label(root, text=0, font=100, bg="#96EE77", fg="black")

computerScore.grid(row=1, column=1)

playerScore.grid(row=1, column=3)

indicators

user_indicator = Label(root, font=50, text="USER'S Score", bg="#96EE77", fg="black")

*comp_indicator = Label(root, font=50, text="COMP'S Score",
bg="#96EE77", fg="black")*

user_indicator.grid(row=0, column=3)

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

messages

msg = Label(root, font=50, bg="#96EE77", fg="black")

msg.grid(row=3, column=2)

update message

def updateMessage(x):

msg['text'] = x

update user score

def updateUserScore():

score = int(playerScore["text"])

score += 1

playerScore["text"] = str(score)

update computer score

```
def updateCompScore():  
  
    score = int(computerScore["text"])  
  
    score += 1  
  
    computerScore["text"] = str(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)
```

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",

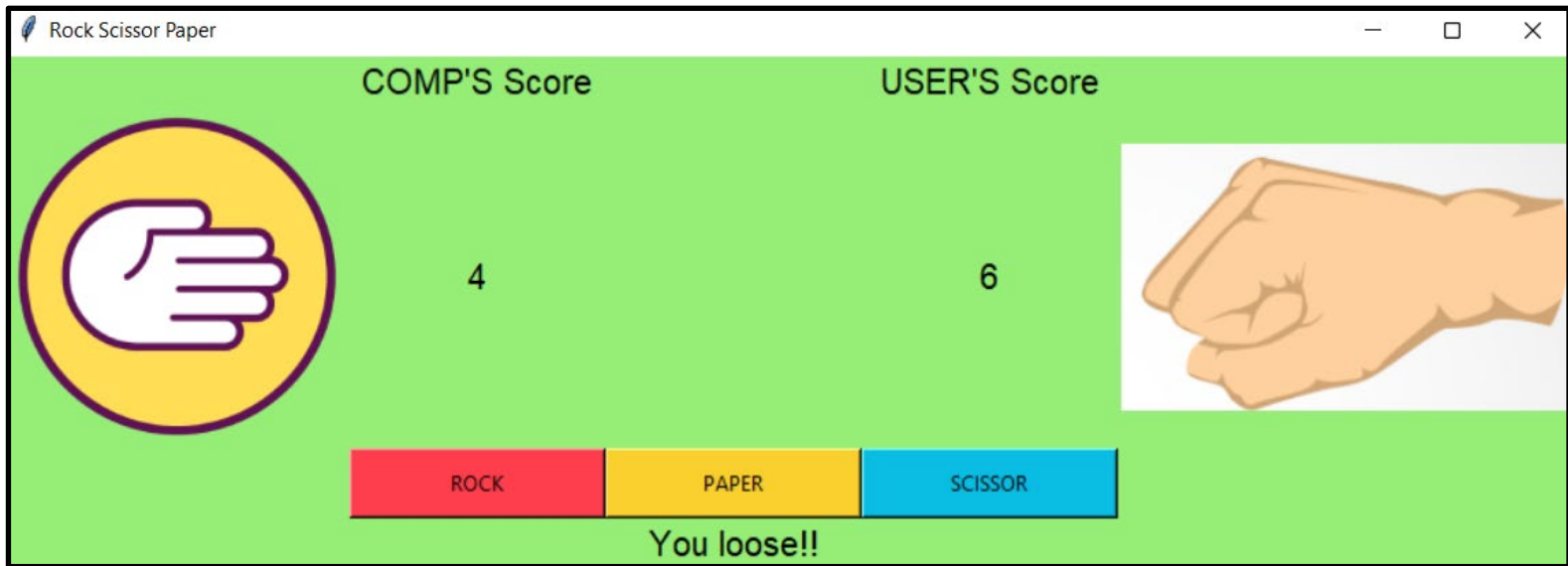
*bg="#FAD02E", fg="black", command=lambda:
updateChoice("paper")).grid(row=2, column=2)*

scissor = Button(root, width=20, height=2, text="SCISSOR",

*bg="#0ABDE3", fg="black", command=lambda:
updateChoice("scissor")).grid(row=2, column=3)*

root.mainloop()

RESULTS



- Rock paper scissor game in python using GUI was implemented
- Each win/loss situation is shown in the following table:

Player 1	Player 2	Result
Rock	Scissors	Player 1 WINS
Rock	Rock	Draw
Rock	Paper	Player 2 WINS
Scissors	Scissors	Draw
Scissors	Rock	Player 2 WINS
Scissors	Paper	Player 1 WINS
Paper	Scissors	Player 2 WINS
Paper	Rock	Player 1 WINS
Paper	Paper	Draw

CONCLUSION

In conclusion, the Rock Paper Scissors game does not follow difficult play patterns and does not conform to complex and unorthodox customs like that of the Olympics.

In other words, it offers simplicity in practice. You can play the Rock Paper Scissors game as an amateur to have fun and subscribe to the honor that it brings.

You can also pursue a career in it by becoming a professional RPS player like in every other professional sport globally.

In any way, the Rock Paper Scissors game is a game for everybody with unique mentalities.

REFERENCES

1. <https://www.geeksforgeeks.org/python-program-implement-rock-paper-scissor-game/>
2. <https://data-flair.training/blogs/python-rock-paper-scissors-game/>