A close up of a sign

Description automatically generated

**Department of Computer Science and Engineering**

**Jain Global Campus, Jakkasandra Post, Kanakapura Taluk, Ramanagara District, Pin Code: 562 112**

**2022-2023**

**A Mini Project Report on**

**“ROCK PAPER SCISSOR GAME”**

**Submitted in partial fulfillment for the internal assessment of**

**Session: Jan 2023 - June 2023**

**Semester: IV ‘B’**

**Course in**

**Python Programming [21CIC45]**

**VARUN SHARMA (21BTRCS095)**

**Under the guidance of**

**Dr. Chandrasekar Venkatachalam**

**Professor**

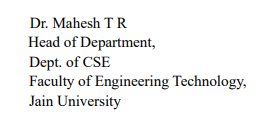
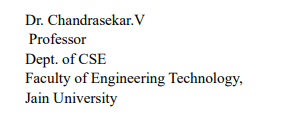
**Department of Computer Science and Engineering**

**Faculty of Engineering & Technology**

**JAIN (DEEMED-TO-BE) UNIVERSITY**

**CERTIFICATE**

This is to certify that the project work on “ROCK PAPER SCISSOR GAME” as part of 21CIC45 – Python Programming is carried out by VARUN SHARMA (21BTRCS095) The Bonafede students of Bachelor of Technology in Computer Science and Engineering at the Faculty of Engineering & Technology, Jain University, Bangalore, during the year 2023-2024.

****

**ABSTRACT**

This project is a ROCK PAPER SCISSOR game implemented using Python programming language. The game allows one player to play against the system. The game starts with an empty board and the player has to take choice between rock,paper,scissor. The game logic is implemented using a combination of conditional statements, loops, and functions. The program includes a graphical user interface (GUI) that allows users to interact with the game using mouse clicks. Overall, this project demonstrates the use of Python programming language to implement a simple but popular game

# 

**INTRODUCTION**

ROCK PAPER SCISSOR is a simple single player game that is played against PC. The goal of the game is to be the player to select between three of the option given. The game will be played on the command line, and we will be using Python's built-in data structures and functions to create the game logic. The game will start by displaying an empty ROCK PAPER SCISSOR board on the command line, and then prompting the first player to make a move. The program will then validate the move, update the game board, and display the new board with the move made by the player. The game will continue until one player has won the game by getting three of their marks in a row, or until the game ends in a tie if all spaces on the board are filled without a winner. At the end of the game, the program will display the winner or indicate that the game ended in a tie, and then ask the players if they would like to play again

**CODE FOR ROCK PAPER SCISSOR**

#ROCK PAPER SCISSOR BY VARUN SHARMA

print("\n\n\nRock Paper Scissors by Varun Sharma\n\npresss Ctrl+. when you feel like exitting the game.")

input("\n\n<<PRESS ENTER TO BEGIN>>\n")

print("\n\n\n\n\n\n\n\n\n\n\n\n\n\n")

#Declaring varibale

MoveSet = [1,2,3]

Cmove=0

Pmove=0

Cscore=0

Pscore=0

#Declaring Framework

from time import sleep

from random import choice

def game():

print("\n\n\n\n\n\n\n\nDecide your move, enter \"R\" for Rock, \"S\" for Scissors and \"P\"for Paper")

tempVal=input()

while tempVal != 'R' and tempVal != 'S' and tempVal != 'P' and tempVal != 'r' and tempVal != 's' and tempVal != 'p':

print("Decide your move, enter \"R\" for Rock, \"S\" for Scissors and \"P\"for Paper")

tempVal=input();

sleep(0.13)

if tempVal == "R" or tempVal == "r":

Pmove=1

elif tempVal == "P" or tempVal == "p":

Pmove=2

elif tempVal == "S" or tempVal == "s":

Pmove=3

else:

print("Custom\_ERROR-01 : can't recognize user's input")

raise SystemExit(0);

sleep(0.13)

Cmove=choice(MoveSet)

if Cmove == 1:

PCmove='Rock'

elif Cmove == 2:

PCmove='Paper'

elif Cmove == 3:

PCmove='Scissor'

else:

print("Custom\_ERROR-00 : PC's move unrecognizable")

raise SystemExit(0)

del tempVal;

sleep(0.038)

return Cmove, Pmove, PCmove

def decisive(Cmove,Pmove):

if [Cmove,Pmove] == [1,3] or [Cmove,Pmove] == [2,1] or [Cmove,Pmove] == [3,2]:

winner=1

elif [Cmove,Pmove] == [1,2] or [Cmove,Pmove] == [2,3] or [Cmove,Pmove] == [3,1]:

winner=2

elif [Cmove,Pmove] == [1,1] or [Cmove,Pmove] == [2,2] or [Cmove,Pmove] == [3,3]:

winner=0

else:

print("Custom\_ERROR-03 : Set [PCmove,playermove] was assigned an unexpected value")

raise SystemExit(0)

return winner

def Scoreboard(winner,cm,cs,ps):

if winner == 1:

print("\nPC won the round by choosing",cm)

cs=cs+1

elif winner == 2:

print("\nPlayer won the round!")

ps=ps+1

else:

print("Custom\_ERROR-02 : Scoreboard function recieved explicit values")

raise SystemExit(0)

print("\nSession Score:")

print("Player : ",ps)

print("PC : ",cs)

return cs, ps

#Driver Code

while True:

Cmove,Pmove,PCmove=game()

winner=decisive(Cmove,Pmove)

while winner == 0:

print("WoW! A Draw!\nBoth Player and PC have chosen",PCmove)

print("let's go again")

Cmove,Pmove,PCmove=game()

winner=decisive(Cmove,Pmove)

sleep(0.13)

cmCache=PCmove;

print("PC - ",cmCache)

RoundWinner=winner

cs,ps=Scoreboard(RoundWinner,cmCache,Cscore,Pscore)

Cscore=cs; Pscore=ps

input("<<< PRESS ENTER TO CONTINUE >>>");

**OUTPUT**:

