STONE PAPER SCISSORS GAME

OBJECTIVE:

To Develop a Python-based Rock, Paper, Scissors game with a user-friendly text interface, random computer choices, scoring, and clear outcome messages, allowing players to play against the computer and producing a report detailing the development process and project outcomes.

ABSTRACT:

The Rock, Paper, Scissors game project is implemented in Python, allowing users to engage in the classic hand game against a computer opponent. The game adheres to the established rules, providing players with the ability to input their choices and receive clear outcome messages. The project aims to create an enjoyable and interactive gaming experience while offering educational insights into Python programming.

INTRODUCTION:

The project focuses on the creation of a digital adaptation of the traditional Rock, Paper, Scissors game using the Python programming language. Rock, Paper, Scissors is a widely recognized hand game played by millions of people globally, and its digital implementation serves both educational and recreational purposes.

In this project, the classic rules of Rock, Paper, Scissors are faithfully replicated. Players are presented with three choices: "Rock," "Paper," or "Scissors." The computer serves as the opponent, randomly selecting its own choice. The outcome of each round is determined according to the established rules: rock defeats scissors, scissors defeat paper, and paper defeats rock. These outcomes are presented to the players through clear and informative messages.

The Project serves as a valuable educational tool for those interested in learning or practicing Python programming. The game's codebase offers a practical platform for beginners and students to explore various programming concepts, such as user input, conditional statements, and random number generation. Additionally, it provides a hands-on experience for debugging and problem-solving, essential skills in software development.

The game's digital nature also enables several advantages, including the potential for customization. Beyond the basic implementation, the project can be extended to incorporate additional features, such as a graphical user interface (GUI), sound effects, or more advanced AI opponents. This flexibility encourages creativity and innovation in its development.

In summary, the Rock, Paper, Scissors project combines the educational value of programming with the joy of playing a timeless game, offering a versatile platform for learning and entertainment.

HARDWARE/SOFTWARE REQUIREMENTS:

Visual Studio Code

CONCEPTS/WORKING PRINCIPLE

1. The program starts by asking the user how many rounds they want to play.
2. It then enters a loop to play the specified number of rounds. In each round, the player is prompted to choose either Stone, Paper or Scissors.
3. The computer randomly selects one of these options for itself.
4. The program compares the choices of the player and the computer to determine the winner for that round. The scores for both the player and computer are updated accordingly.
5. Clear feedback is provided, showing the choices made and the current score after each round.
6. After all rounds are played, the program announces the overall winner or if it's a tie.
7. The user is then asked if they want to play the game again. If they choose "yes," the game restarts, and if they choose "no," the program exits.

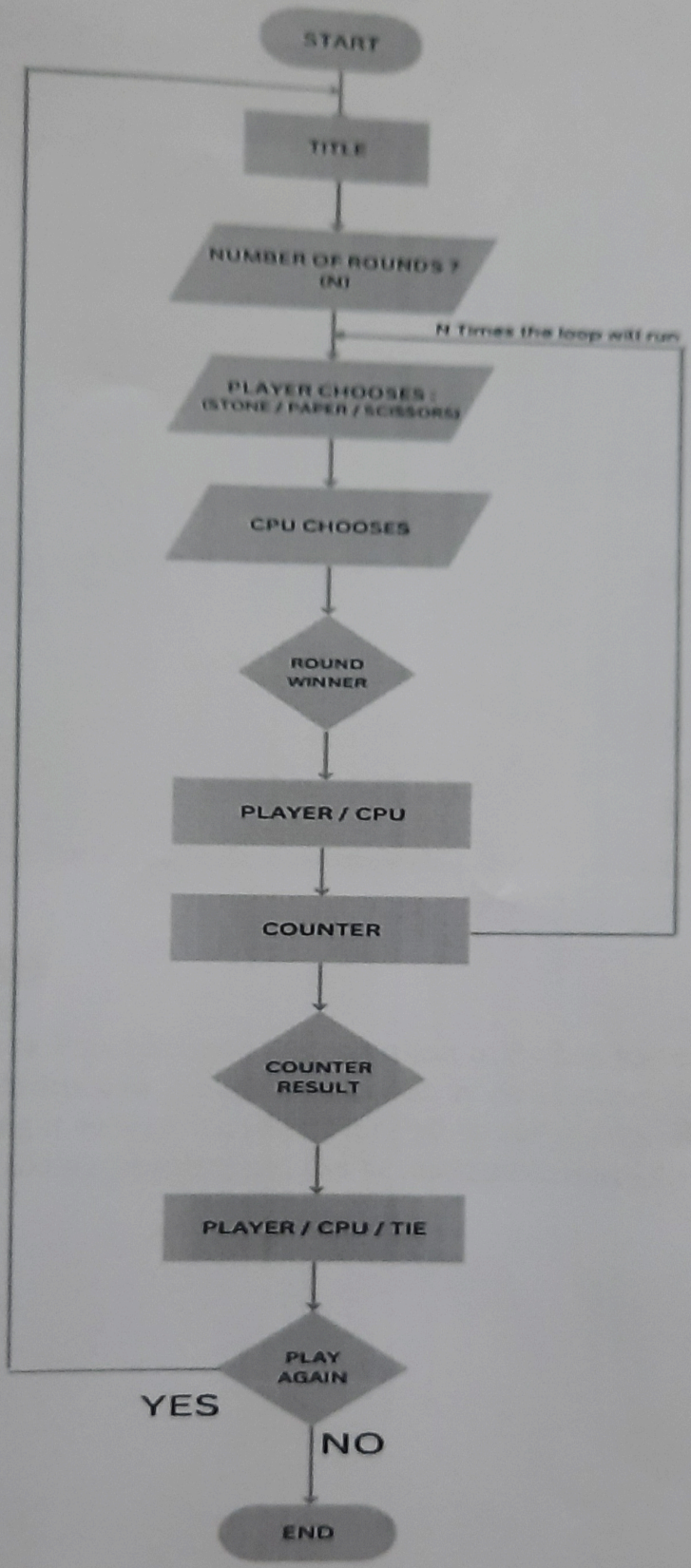
APPROACH/METHODOLOGY/PROGRAMS:

```
import random
import time

while True:
    print("\n")
    print("STONE PAPER SCISSORS GAME\n\n")
    print("HOW MANY ROUNDS WOULD YOU LIKE TO PLAY : ", end="")

    choice = int(input())
    print("\n")
    ps = 0
    cs = 0

    for x in range(choice):
        print(x + 1, "   PLAYER :", " CHOOSE STONE PAPER OR SCISSORS : ",
end="")
        choose = input()
        list = ["STONE", "PAPER", "SCISSORS"]
```

```
        r = random.choice(list)
        time.sleep(1)
        print(x + 1, "    CPU : ", r)
        print("\n")
        time.sleep(1)

        if choose == "STONE" and r == "PAPER":
            cs = cs + 1
            print("CPU : ", cs)
            print("PLAYER : ", ps)
            print("\n")
        elif choose == "PAPER" and r == "STONE":
            ps = ps + 1
            print("CPU : ", cs)
            print("PLAYER : ", ps)
            print("\n")
        elif choose == "PAPER" and r == "SCISSORS":
            cs = cs + 1
            print("CPU : ", cs)
            print("PLAYER : ", ps)
            print("\n")
        elif choose == "SCISSORS" and r == "PAPER":
            ps = ps + 1
            print("CPU : ", cs)
            print("PLAYER : ", ps)
            print("\n")
        elif choose == "STONE" and r == "SCISSORS":
            ps = ps + 1
            print("CPU : ", cs)
            print("PLAYER : ", ps)
            print("\n")
        elif choose == "SCISSORS" and r == "STONE":
            cs = cs + 1
            print("CPU : ", cs)
            print("PLAYER : ", ps)
            print("\n")
        else:
            print("CPU : ", cs)
            print("PLAYER : ", ps)
            print("\n")

time.sleep(1)

if cs == ps:
    print("TIE :/")
    print("\n")
elif cs > ps:
    print("CPU WON :(")
    print("\n")
```

```
    else:
        print("YOU WON :)")
        print("\n")

    restart = input("DO YOU WANT TO PLAY AGAIN ? (YES/NO) : ")
    if restart.lower() != "yes":
        break
```

FLOWCHART:


START
TITLE
NUMBER OF ROUNDS ? (N)
N Times the loop will run
PLAYER CHOOSES : (STONE / PAPER / SCISSORS)
CPU CHOOSES
ROUND WINNER
PLAYER / CPU
COUNTER
COUNTER RESULT
PLAYER / CPU / TIE
PLAY AGAIN
YES
NO
END

OUTPUT:

```
STONE PAPER SCISSORS GAME

HOW MANY ROUNDS WOULD YOU LIKE TO PLAY : 3

1     PLAYER :  CHOOSE STONE PAPER OR SCISSORS : STONE
1     CPU :  SCISSORS

CPU :  0
PLAYER :  1

2     PLAYER :  CHOOSE STONE PAPER OR SCISSORS : SCISSORS
2     CPU :  STONE

CPU :  1
PLAYER :  1

3     PLAYER :  CHOOSE STONE PAPER OR SCISSORS : PAPER
3     CPU :  PAPER

CPU :  1
PLAYER :  1

TIE :/

DO YOU WANT TO PLAY AGAIN ? (YES/NO) : NO
```

CONCLUSIONS:

Finally, The 'Rock Paper Scissors' game project in Python has come to life, delivering an interactive and instructive gaming experience. It underscores fundamental programming principles, making it an ideal launchpad for Python enthusiasts. The Project can Further be Customized (by adding graphics) and can be made more interactive for the users.