VIT®

BHOPAL

VITYARTHI PROJECT
PYTHON ESSENTIALS

SHARANYA MITRA

25MIP10035

Project Objective

The goal of this project is to allow students to apply the subject concepts in a real-world context by:

- **Identifying a meaningful problem**
- **Designing a technical solution**
- **Implementing the solution using the tools/methods learned in the course**
- **Demonstrating understanding through documentation and evaluation**

1.1 PROBLEM DEFINATION

This project is a score-based, command-line implementation of Rock-Paper-Scissors. It runs as a 'First to N Points' match, which starts by setting the target number of points for winning. It then goes into an interactive, repetitive round where the User enters their choice (rock, paper, or scissors) against a Bot that randomly selects one. This program keeps track of points on both sides and uses some conditionals to determine who wins the current round and how to update the scores. The game instantly stops when one of the opponents-User or Bot-reaches the score limit set earlier, and it announces the final winner and the conclusive scores.

1.2 REQUIREMENT

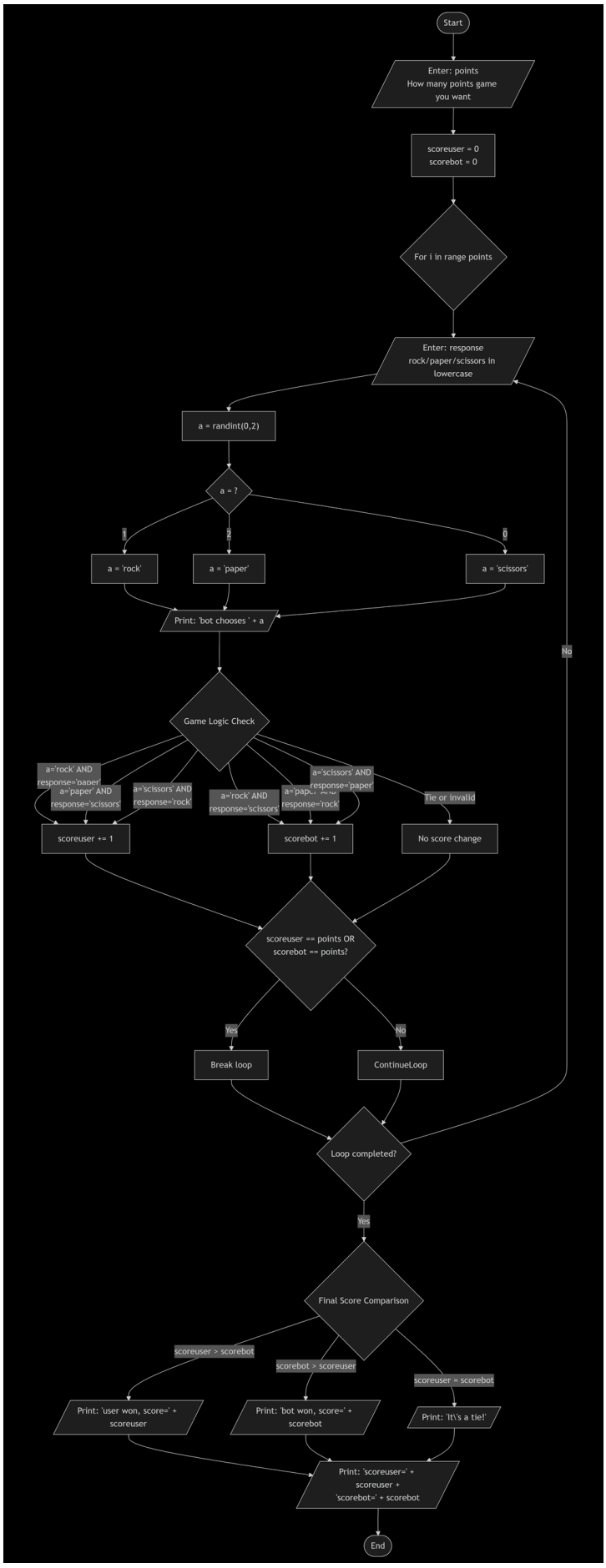
- 1. Software Requirements
- OS: Any modern operating system that is capable of running Python.
- Examples include: Windows, macOS, Linux (e.g., Ubuntu, Fedora).
- The programming language is in Python. Version 3.x is highly recommended since that is the present standard.
- Execution Environment: Any terminal, command prompt, or IDE can be used for running the .py file.
- Examples: IDLE, VS Code, PyCharm or standard system terminal.
-
- 2. Hardware Requirements
- Since this is a very simple text-based program, hardware requirements are minimal.
- Processor (CPU): Any modern processor capable of running an OS and Python.
- Memory (RAM): Minimal. Any system with 1 GB of RAM or more will run this code without any issues
- Storage: A few kilobytes of free space to store the Python file. Input/Output: Keyboard- A conventional keyboard for inputting point limit, and choices. Display/Monitor- Display to view output.

2. DESIGN

2.1 ALGORITHM

1. **START the program.**
2. **Prompt the user to input the desired winning score and store this value in the variable points.**
3. **Initialize two score tracking variables, scoreuser and scorebot, both to 0.**
4. **Begin a game loop that is intended to run for a maximum of points iterations.**
5. **Inside the loop, prompt the user to enter their choice (rock, paper, or scissors) and store it in response.**
6. **Generate a random integer between 0 and 2 (inclusive), storing it in variable a.**
7. **Translate a into the Bot's choice: if a=1, the choice is "rock"; if a=2, the choice is "paper"; otherwise (a=0), the choice is "scissors".**
8. **Display the Bot's choice to the user.**
9. **Check the six possible winning conditions for the round using conditional logic (if statements) comparing the Bot's choice a and the User's choice response.**
10. **If a condition where the User wins (e.g. Bot is "rock" and User is "paper") is met, increment scoreuser by 1.**
11. **If a condition where the Bot wins (e.g. Bot is "rock" and User is "scissors") is met, increment scorebot by 1.**
12. **Check the game termination condition: if scoreuser is equal to points OR scorebot is equal to points, immediately BREAK out of the loop, ending the game rounds.**
13. **If the loop was not broken, continue to the next iteration.**
14. **After the loop terminates (either by breaking, or completing all its iterations), compare the final scores.**
15. **If scoreuser is greater than scorebot, print a message stating that user won and the final scores for both players.If scorebot is greater than scoreuser, print a message stating bot won and final scores for both players.**
16. **END the program**

2.1 FLOWCHART



3.1 CODE IMPLEMENTATION

```
import random as r
points=int(input("enter how many points game you want"))
scoreuser=0
scorebot=0
for i in range(points):
    response=input("enter a choice in lower case rock/paper/scissors")
    a=r.randint(0,2)
    if a==1:
        a="rock"
        print("bot chooses",a)
    elif a==2:
        a="paper"
        print("bot chooses",a)
    else:
        a="scissors"
        print("bot chooses",a)
    if a=="rock" and response=="paper":
        scoreuser+=1
    elif a=="rock" and response=="scissors":
        scorebot+=1
    elif a=="paper" and response=="scissors":
        scoreuser+=1
    elif a=="paper" and response=="rock":
        scorebot+=1
    elif a=="scissors" and response=="paper":
        scorebot+=1
    elif a=="scissors" and response=="rock":
        scoreuser+=1
    elif scoreuser==points or scorebot==points:
        break
if scoreuser>scorebot:
    print("userwon,score=",scoreuser)
    print("scoreuser=",scoreuser,"scorebot=",scorebot)
elif scorebot>scoreuser:
    print("bot won,score=",scorebot)
    print("scoreuser=",scoreuser,"scorebot=",scorebot)
else:
    print("It's a tie!")
    print("scoreuser=",scoreuser,"scorebot=",scorebot)
```

```
enter how many points game you want 5
enter a choice in lower case rock/paper/scissors rock
bot chooses paper
enter a choice in lower case rock/paper/scissors paper
bot chooses scissors
enter a choice in lower case rock/paper/scissors scissors
bot chooses paper
enter a choice in lower case rock/paper/scissors paper
bot chooses scissors
enter a choice in lower case rock/paper/scissors rock
bot chooses rock
bot won,score= 3
```

CODE DESCRIPTION

This is a Python script that plays a classic game of Rock-Paper-Scissors against the computer (the "bot") over a number of rounds specified by the user. Its main use is to offer an interactive, simple text-based gaming experience. The description is as follows: The program asks the user to input the total number of points/rounds for the game at the beginning. Then it enters a loop in which, in each round of the game, the user inputs his choice (rock, paper, or scissors), while the bot randomly generates its move. The code compares the choices, gives a point to the winner of the round, and keeps track of scores (scoreuser and scorebot). Other than that, an early exit condition exists in the game-if scoreuser==points or scorebot==points: break, meaning that the game will end with the first player reaching the target score, even if the total number of defined rounds has not been completed. Such code has the following advantages: It is a very simple and clear example that provides straightforward implementation of game logic, random number generation, and conditional scoring. This program is perfect for illustrating some of the elementary concepts of Python programming, including loops, conditional statements, and ways of handling user input.

REFERENCES

https://github.com/sharanyamitra1-ship-it/stone_-paper-scissors-project

https://docs.python.org/3/library/random.html

https://www.geeksforgeeks.org/python/python-random-module/