Table of Contents

Table of Figures

Introduction

This is the report of the individual assignment heading Software Unit Testing for the module S222 PRT582 SOFTWARE ENGINEERING: PROCESS AND TOOLS. In the given task, we Scissor Paper Rock game using Test Driven Development in Python.

The assigned task was done to meet the requirements, where the game was developed in python, and the player gets to choose one of the options between scissors, paper, and rock. This is then compared against the computer's selection and determines who the winner is as per the rules given in the requirements. The basic game requirements were:
A player is then given the option to enter one of the options of scissors, paper, and rock.
The computer randomly picks one of the options of scissors, paper, and rock.
One point is given to the winner.
The first to get five points wins the game. The total number of rounds played in total will also be displayed.
Once the winner is determined, the player is asked to quit or restart the game.
Players can also quit the game at any time.

The primary objectives of the assignment were:

- To develop a Scissor Paper Rock game using Test Driven Development in Python.
- To create a Git directory
- To do testing in every step and repeat the action till it passes the testing
- To write the report and submit it via learning the line.

Different types of testing tools should be used, like flake8 and pyunit. Test Driven Development (TDD) is a method of software development combining test-first development and refactoring. In other words, we write a test before we write just enough coding to fulfill that test.

Flake8 is the python library, the wrapper that verifies pep8, pyflakes, and circular complexity. With the help of flake8 helps us by preventing us from making simple syntax errors, bad formatting, and incorrect styling. It improves our code by checking against its standard and is easy to use.

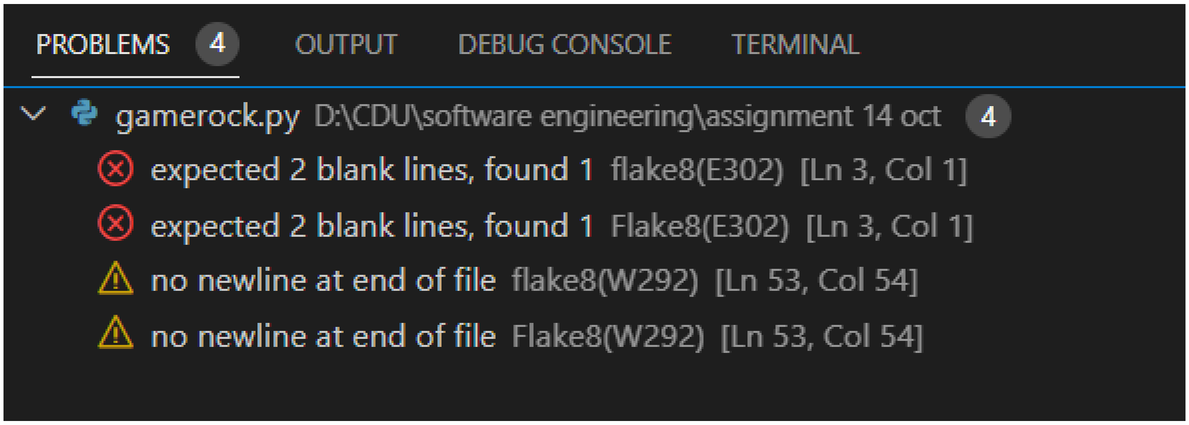
PyUnit for Unittest is the software testing method with a small piece of code called units. Some unit testing libraries are "PyUnit and PyTest."

Process

While developing the program, different automated tools are used. TDD reduces the number of bugs in code. The package requires the flake8 program should is installed on our system. To install flake8, run pip install flake8 or easy_install flake8 from the executed command line.

Testing basic requirements:

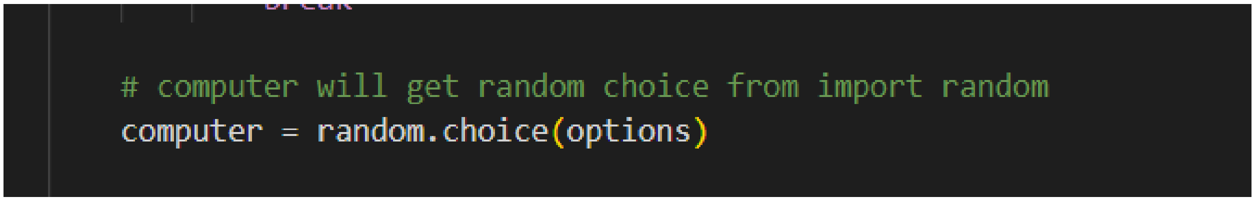
- Here some error was shown by our installed library flake which helped to maintain a good standard of the code.



```
gamerock.py  D:\CDU\software engineering\assignment 14 oct  4
  ⊗ expected 2 blank lines, found 1  flake8(E302)  [Ln 3, Col 1]
  ⊗ expected 2 blank lines, found 1  Flake8(E302)  [Ln 3, Col 1]
  ⚠ no newline at end of file  flake8(W292)  [Ln 53, Col 54]
  ⚠ no newline at end of file  Flake8(W292)  [Ln 53, Col 54]
```

Figure 1 Testing via Flake8

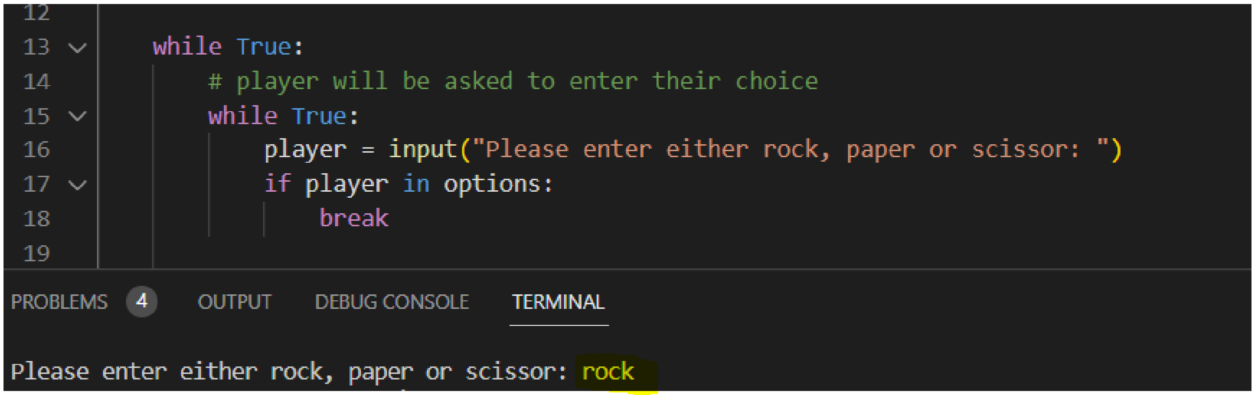
- The computer should select its choice randomly. It is doing so with the help of a package we install named random.



```
    # computer will get random choice from import random
    computer = random.choice(options)
```

Figure 2 Computer chooses random

- Players are allowed to choose any option, as shown in the figure below. They should enter any chance to compete against the computer, which picks a random choice.



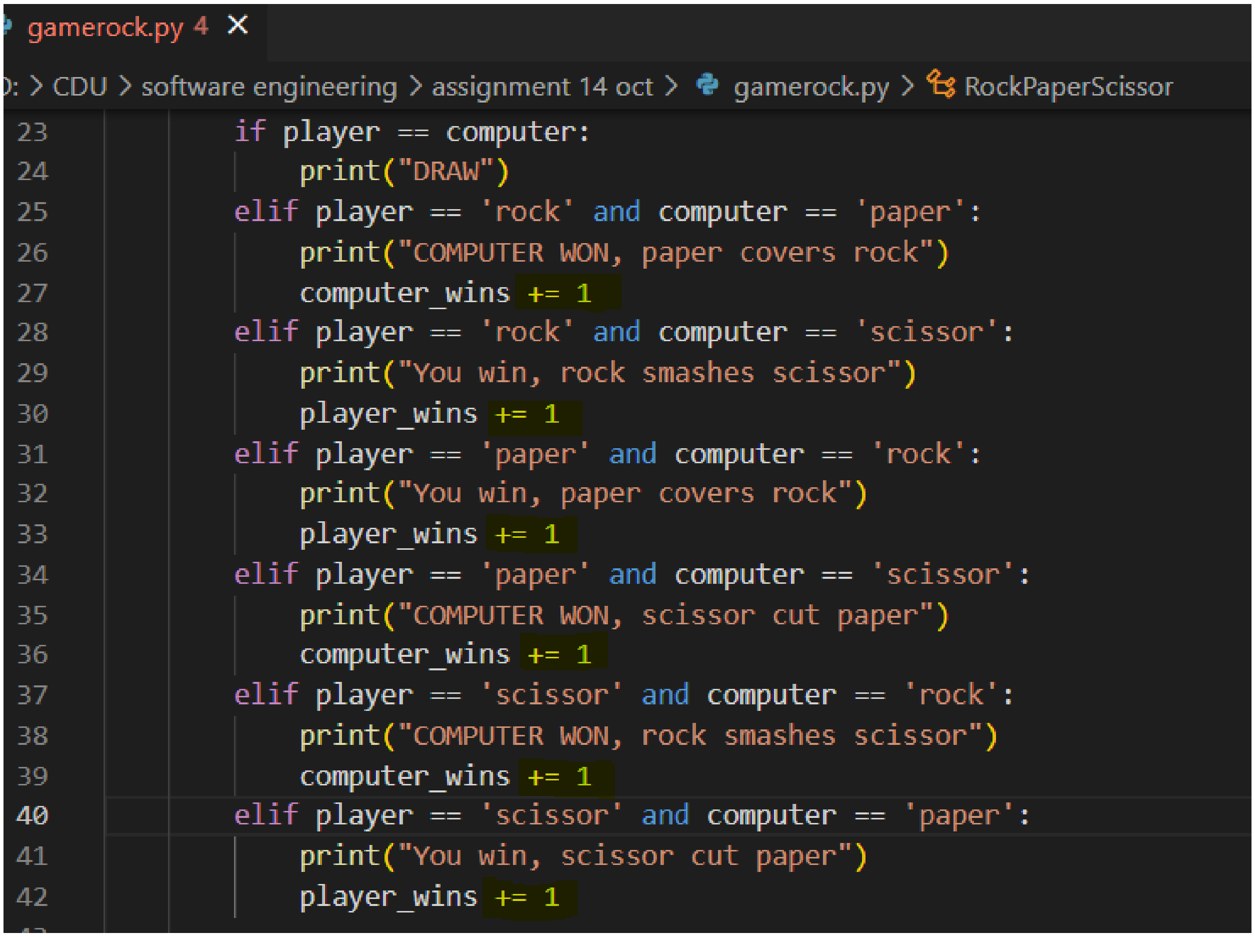
```
        while True:
            # player will be asked to enter their choice
            while True:
                player = input("Please enter either rock, paper or scissor: ")
                if player in options:
                    break

```

```
Please enter either rock, paper or scissor: rock
```

Figure 3 Player chooses one option

- Every time one point is added to the winner

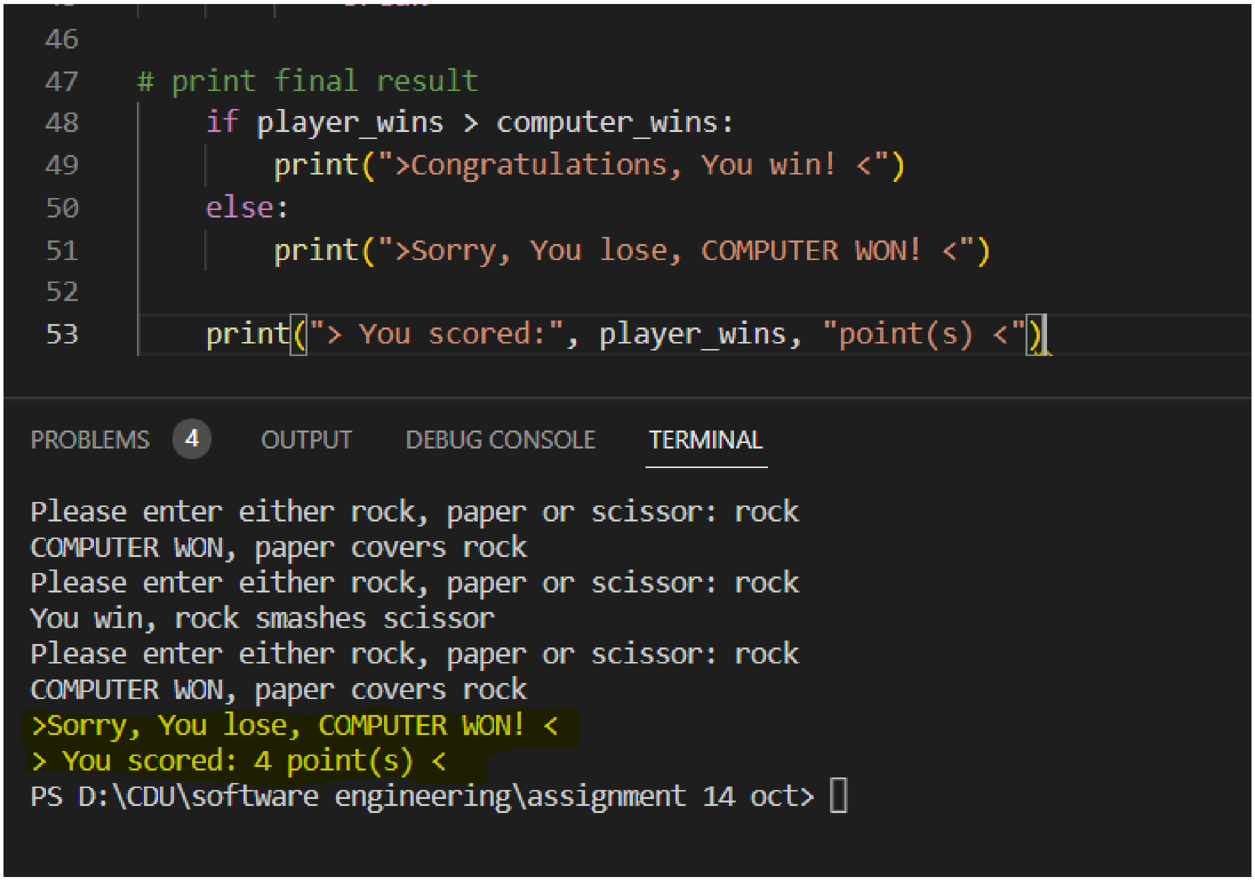


gamerock.py 4 ×
D: > CDU > software engineering > assignment 14 oct > gamerock.py > RockPaperScissor

```
            if player == computer:
                print("DRAW")
            elif player == 'rock' and computer == 'paper':
                print("COMPUTER WON, paper covers rock")
                computer_wins += 1
            elif player == 'rock' and computer == 'scissor':
                print("You win, rock smashes scissor")
                player_wins += 1
            elif player == 'paper' and computer == 'rock':
                print("You win, paper covers rock")
                player_wins += 1
            elif player == 'paper' and computer == 'scissor':
                print("COMPUTER WON, scissor cut paper")
                computer_wins += 1
            elif player == 'scissor' and computer == 'rock':
                print("COMPUTER WON, rock smashes scissor")
                computer_wins += 1
            elif player == 'scissor' and computer == 'paper':
                print("You win, scissor cut paper")
                player_wins += 1
```

Figure 4 One point adding to the winner's account

- Finally, the final score and result will be shown when whoever gets 5 points first.



```
    # print final result
        if player_wins > computer_wins:
            print(">Congratulations, You win! <")
        else:
            print(">Sorry, You lose, COMPUTER WON! <")

        print("> You scored:", player_wins, "point(s) <")
```

```
Please enter either rock, paper or scissor: rock
COMPUTER WON, paper covers rock
Please enter either rock, paper or scissor: rock
You win, rock smashes scissor
Please enter either rock, paper or scissor: rock
COMPUTER WON, paper covers rock
>Sorry, You lose, COMPUTER WON! <
> You scored: 4 point(s) <
PS D:\CDU\software engineering\assignment 14 oct>
```

Figure 5 Final score and result

Different types of testing were done to meet the game's requirements, and the report was written after testing and developing the program.

Conclusion

All the tasks assigned in the coursework were finally achieved through numerous trials and bunches of mistakes. The assignments given in the coursework were challenging by any stretch of the imagination. It required lots of exertion and research. Each errand(task) was passed out in steps to successfully complete a considerable number of assignments, conveying the total effort. At first, lots of study and research were done on essential topics like TDD, Unit testing, different tools, python programming, and the different strategies for testing the program. At that point, the schedule for reading documents was created and kept in reasonable data structures for composing the program. The testing tools were used in the next stage, and the program was written. From that point, the python program was composed.

Furthermore, finally, the composing program was tested to verify that it had no bugs and mistakes and conveyed the exact outcome. Finally, after the finish of all the given assignments, the submission was made. This project didn't just finish the given task in the coursework on time.

Additionally, it helped in creating different kinds of new abilities and talented me numerous things which can be extremely valuable in a future career as a programmer. While being associated with this coursework, helpful information about python, its different in-manufactured capacities, and its data structures, while and for loops, comments, if/else conditionals, various testing tools, methods, libraries, and functions were obtained. And this learning would be a great deal supportive in my career improvement as a decent developer or as a professional. Significant experience has been acquired while working on this coursework. With everything considered, however, the tasks were challenging and required lots of work and a ton of effort; finishing those extreme tasks effectively was extremely fun. Since this project was anticipated to practical finish and accommodation of the considerable number of functions given in the coursework, also, this project was still plenty of things that should be improved to be perfect. This project can be made perfect by doing individual testing and going through every testing method, but this project lacks that due to many issues. Although I believe it meets the basic requirements as a project in wholesome. To be a decent tester, I need to learn more and implement my learning on further assignments.

Before finishing this coursework, I profited a lot, and it helped me develop my base of python and the testing method and process.

Link for the project in github

https://github.com/Anup-kadariya/SE/blob/main/gamerock.py