

**UNIT NAME**

**SOFTWARE ENGINEERING:PROCESS AND TOOLS**

**PRO: CHARLES YEO**

**ASSESSMENT**

**SUBMITTED BY: VISHNUMOLAKALA HEMALATHA**

**STUDENT ID: S361048**

**STUDY LOCATION: Casuarina Campus**

ROCK PAPER SCISSIOR USING TDD AND UNIT TESTING

|  |  |
| --- | --- |
|  |  |

# Introduction:

This project mainly aims to develop  **a rock, paper and scissor** using **Test Driven Development** in **Python. In this**game, the player gets to choose one of the options between scissors, paper, and rock. This game can be played between two players. The scoring system is based on points given by the winner. The main objective of the task is to write the rock, paper, and scissor game using Test-driven development(TDD) in python, and it describe how automated unit testing is used.

      Here, we use the python programming language to develop this task because of its availability of libraries, popularity, and ease of understanding.

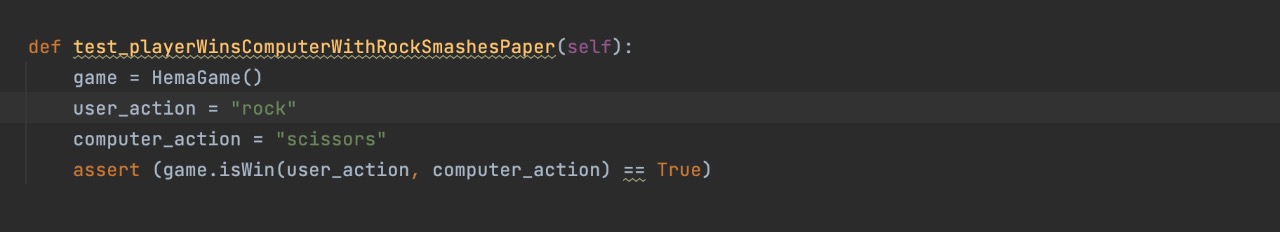
The following are the requirements of the game:

1. The computer can randomly pick any one of the three options (rock, paper, and scissors)
2. It is basically between the two persons, which can be shown by hand gestures.
3. If the rock smashes the scissors, then the rock wins the game. If paper covers the rock, then paper wins the game. Finally, if scissors cut the paper, then scissors win the game. These are the basic conditions of the game.
4. The winner will get one point in each round.
5. If a person has won the five 5 points, then he has a choice whether he can continue with the game or quit from the game.

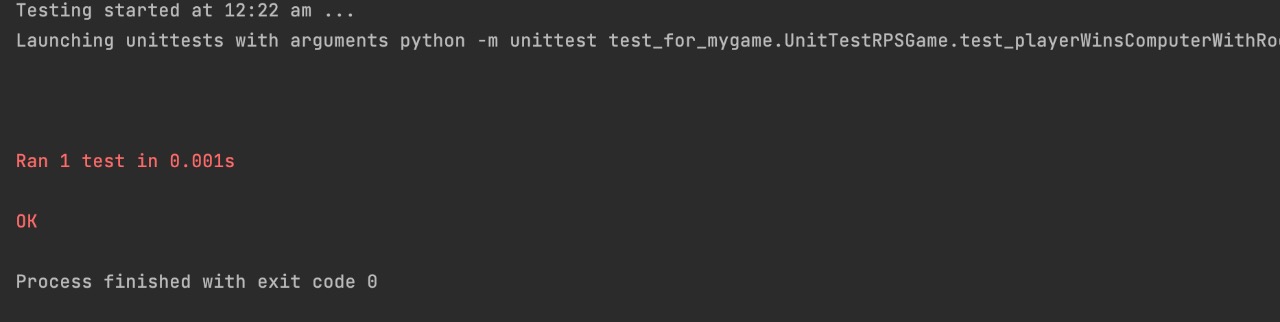
Pytest was used for testing because of its simplicity and ease of getting started with writing unit test cases and TDD. the unit test frame was originally inspired by JUnit and had a similar flavor as major unit testing frameworks in other languages. It supports the test automation and shutdown code for tests and the independence of the tests from the reporting framework. Unit test is an automated test tool that verifies the small piece of unit code in the program. It checks for a specific response to a particular set of inputs. This test can help to reduce the errors in the code and analyze the tool which was used in the pytest.

**PROCESS:**

In this game first we have to write the code then we have to write the test cases by using different methods. The TDD development of the and the test cases are explained below.

Testcase1:

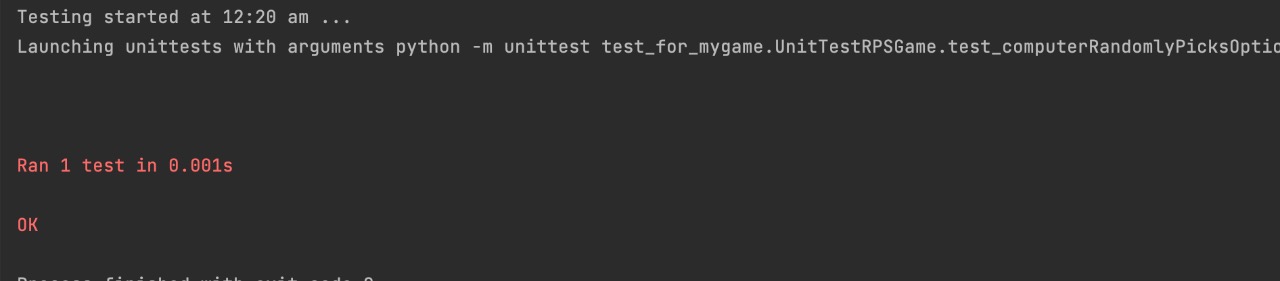
Result:



Test case2;



Result2:



**CONCLUSION:**

Test Driven Development was found to be beneficial in developing the code with different inputs and scenarios. If we can write the test cases before writing the code, it leads to developing the code and less prone to miss out on some edge cases. Also, writing test cases help us to find errors more quickly and solve them easily. for this game, when we can write the different test cases, it would be much easier to write the main code, and errors can be easily identified and solved. It is learned that writing good test cases is key for TDD to be working in an effective way. TDD’s ability is to find bugs quickly might increase the development pace as bugs are found right away when the code is in the nascent stages. The unit test case is not more efficient in finding out the bugs and complex errors in the system, ranging from multiple modules.

Overall, TDD helps in delivering a robust product and increases the pace of development if the test cases are designed smartly.

**https://github.com/hemalath1919/assigniment1\_tools\_SE**