

Assignment 1 – PRT 582

Khang Nguyen – S331735



October 14, 2022

Lecturer: Charles YEo

Charles Darwin University – Semester 02/2022

Table of Contents

[1. Introduction 2](#_Toc116261937)

[a. Objectives: 2](#_Toc116261938)

[b. Requirements: 2](#_Toc116261939)

[c. Automated Unit Testing Tool: 3](#_Toc116261940)

[2. Process 4](#_Toc116261941)

[3. Conclusion 9](#_Toc116261942)

# Introduction

## Objectives:

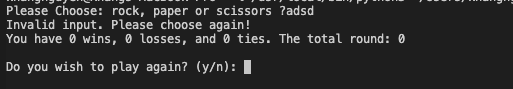
This assignment requires the developer to write, test and develop a "Scissor, Paper, Rock" game using the Python coding language. Moreover, the game must implement TDD, also known as "Test Driven Development", and an automated unit testing tool to create the game.

In this software unit testing report, the rules and requirements of the game will be demonstrated. Moreover, the report also addresses how the TDD (Test-Driven Development) methodology and automated unit testing tool are implemented in the development phases. Moreover, a Git directory is created to store the programming code and all the pdf documents; the link will be included at the conclusion.

## Requirements:

The requirements of the game are relatively simple and are outlined below with support screenshots:

1. The player will choose one of the three options: paper, rock, or scissors. The player must type the correct input, or the system will reject the input and ask the player to type again.



1. The computer will be using "Random Module" to select its option. The player's options are then compared with the computer's options to determine who the winner is. The winning rules are outlined below:
   * Rock vs Paper -> Paper wins
   * Rock vs Scissors -> Rock wins
   * Paper vs scissors -> Scissors wins
2. One point will be assigned to the winner in each round. The first to achieve 5 points wins the game. Furthermore, the total number of rounds must be also displayed.
3. The players are asked if they want to quit or restart the game whenever the winner is determined. If the player chooses to restart the game, all the scores must be restarted from the beginning.
4. Moreover, the player can also choose to quite the game at any time during the game.

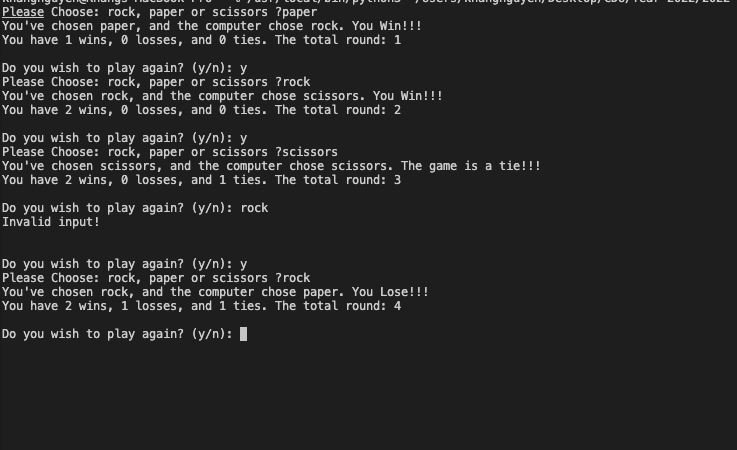


Figure : Choose the options and display the results + total round

Text

Description automatically generatedOnce the winner is determined, the player wants to restart the game, and all the scores have been restarted again.

Figure : Restart the game

## Automated Unit Testing Tool:

TDD or Test-Driven Development emphasises the creation of unit test cases before writing any actual code. It is an iterative process that includes refactoring, unit testing, and programming.

The software can be tested using automated tools, frameworks, and scripts through the practice of automation testing. This kind of software testing involves creating reusable test scripts to test an app's functionality, which cuts down on overall regression time and speeds up software releases. Unittest is used as the Unit Testing Framework in this coding practice. It tests a single piece of code—typically modularised as a function—performs as intended. Furthermore, Unittest will be used in this assignment to test all the test-case codes.

# Process

Identifying bugs and errors during the software development process is essential for improving the final product's quality and reducing the time-consuming for the developers and the team. Therefore, software testing is essential in the software 'life cycle' and primarily focuses on the "paper, rock and scissors" game.

There are several approaches in terms of software testing, and TDD or Test-Driven Development is one of them. It is a crucial technique for extreme programming and implies that only unit testing can develop or change the coding unit. Test-first development is a component of the software-driven TDD approach. To complete the test and refactoring, the developer must create a fully automated test case before writing the production code. Writing a test that fails before beginning to code a feature is a practice. The feature code is improved till the unit test passes. The diagram below demonstrates the process of TDD.

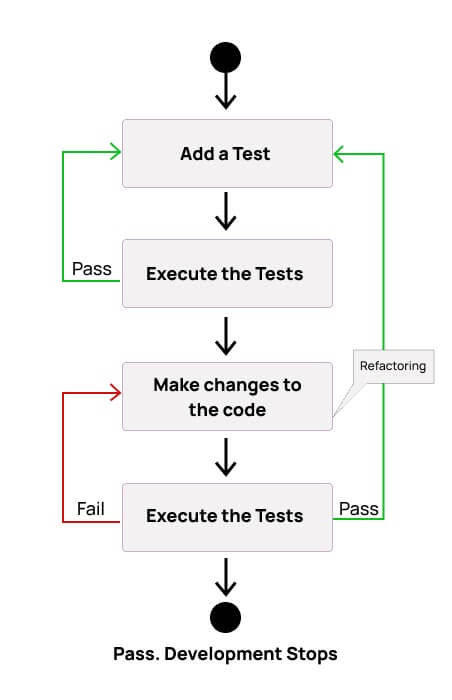


Figure : Test-Driven Development (TDD) Process and Agile Process

The game "Paper, rock and Scissors" is developed with Python coding language, and the code's structure and the game are separated by its requirements. Mainly, the game's requirements have been introduced in the Introduction of this coursework; a single functionality of the code is created for each requirement. "Paper, Rock and Scissors" is built via a class to handle the game, and the methods are created based on the requirements.

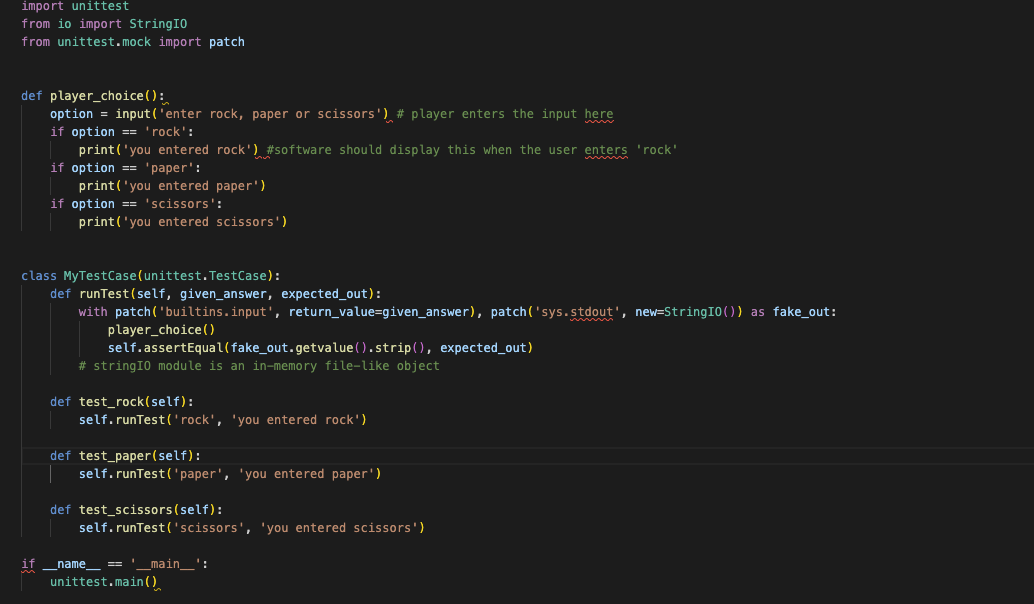
By implementing the diagram of TDD (Test-Driven Development), the game is building on these steps:

1. Firstly, developer will add a test.
2. Developer will run all the tests and see if any test fails.
3. For tests that fail, developer makes changes to the code.
4. Re-test the new code and if they fail, they should be refactored and repeat the process

In other words, unit testing is designed to test a single, small software functionality without waiting for other parts of the code to be completed. By doing this, it can be verified that every piece of software operates as intended. In Python, a unit test framework known as unittest is explicitly developed for testing purposes and is mainly used in this coursework. The unit testing framework in Python comes with several capabilities, including test automation, sharing setup and shutdown code, grouping tests into collections, and independence from the reporting framework.

For each requirement in the game, there will be a single functionality of code to fulfil those requirements or rules. In addition, there will also be a unittest python file to test that part of the code. Below is the demonstration of the unittest and its corresponding part of code:

1. **Players options:**

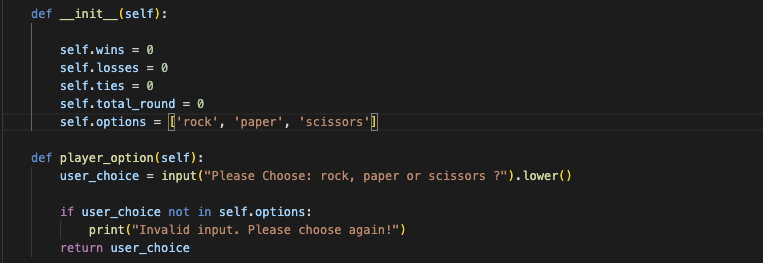
As a part of the game, the player must choose one of the three options: paper, rock or scissors in order to continue the game. The software must take the correct input from the player and display it.

Along with the unittest, the StringIO module is also used. The module is defined as an object that resembles a file in memory. Most functions anticipating a standard file can accept this object as input or output. The function of the StringIO object receives a string as input when it is constructed, which initialises the object. The StringIO will begin empty if no string is given.

Text

Description automatically generatedWhen the player is given input as either "paper, rock or scissors", the software must take the input and display it as "you entered rock – paper -scissors". The result of the test is shown below:

In advanced, the code is refactoring to identify if the player enters incorrect input:

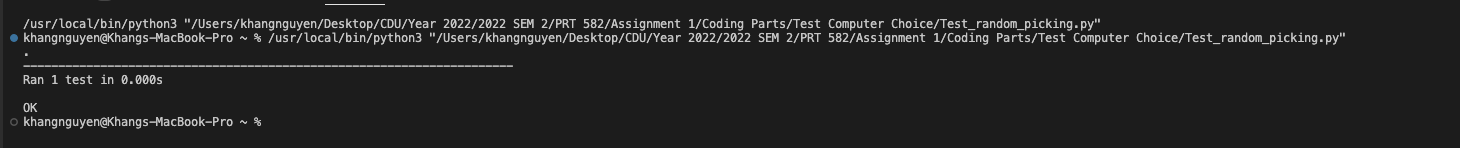


1. **Computer random choice:**

In addition, as a part of the game's requirements, the computer must be able to randomly choose one of the three options just like the player: rock, paper, or scissors; to ensure fairness in the game. Below is the unittest of the computer choice



Firstly, a list includes ['rock', 'paper','scissors'] is defined with the variable 'options'. Then the random module combines with the choices() method. This will allow the computer to choose one of the three options in the list randomly. The test result is shown below:



1. **Test the point system:**

Being mentioned above, the winning rules are listed as:

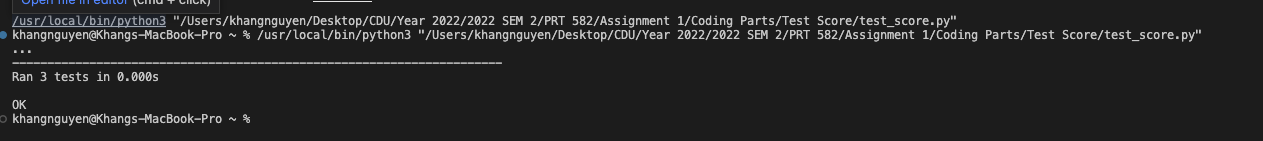
* rock vs paper -> paper wins
* rock vs scissor -> rock wins
* paper vs scissor -> scissor wins.

And one point will be assigned to the winner. Below is the unittest of the score system

Text

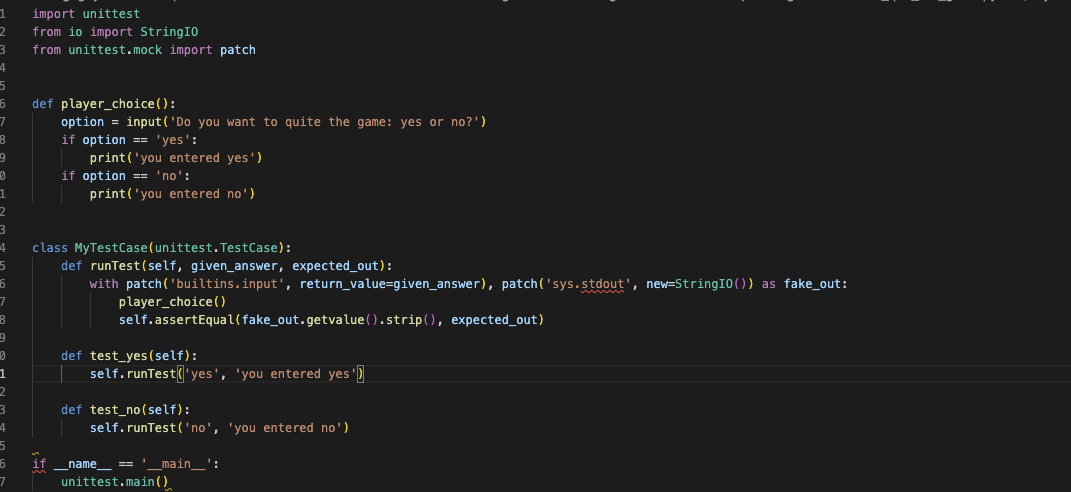
Description automatically generated

There are three variables in the point system: ties, wins, and losses. If the player's option is the same as the computer's, then the ties variable will be added by one point. Moreover, other scenarios are already described in the requirements. Below is the result of the testing:

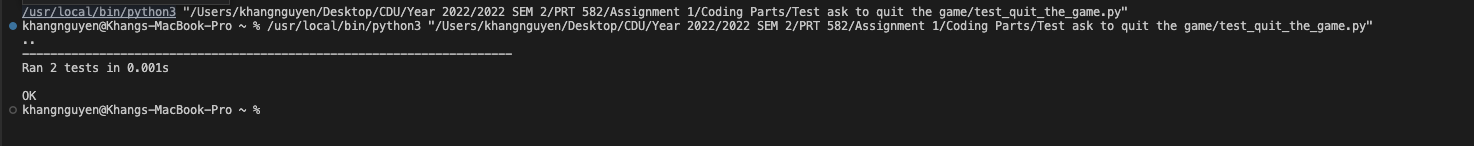


1. **Asking to quit the game**

Moreover, the game allows the player quite the game any time after each round. Similar to the player\_option; the StringIO will be also used in this test. Below is the unittest:



And the result is shown as below:



In advanced, this piece of code has been developed further to ensure fulfill the requirements. If the player or computer reaches 5 points first will win the game and ask to quite or restart the game. The code below demonstrates that:



# Conclusion

In conclusion, overall, the program – "paper, rock, and scissors" has met all the requirements and provides an exemplary user interface. The overall performance of the software has been tested carefully to ensure the game meets its standard. The coding style in the program has been checked against flake8 to ensure the style is well-formatted and appropriate use of language capabilities. Moreover, comments are included in the program to ensure the code is easy to read.

However, there are also a few points that need to be improved. Specifically, the Unittest file should be separated from the code file, and the unit test file should import the module of the code file. Moreover, the code of the program has been repeated and, therefore, is not showing very concisely and is space-consuming. This can be improved by considering the time and space complexity of the program and applying various computer algorithms to improve the code.

The Git directory: https://github.com/Khangs331735/PRT-582---Assignment-1---Khang-Nguyen---S331735