**Report of Rock Paper Scissors Game**

**Chen Chen**

**(S360496)**

**Introduction**

**Objectives**

This application is designed for Rock Paper Scissors Game. The objective of Rock Paper Scissors Game is to defeat your opponent, which is the computer, by using a weapon that you choose from rock, paper or scissors. Rock beats scissors, scissors beat paper and paper beats rock.

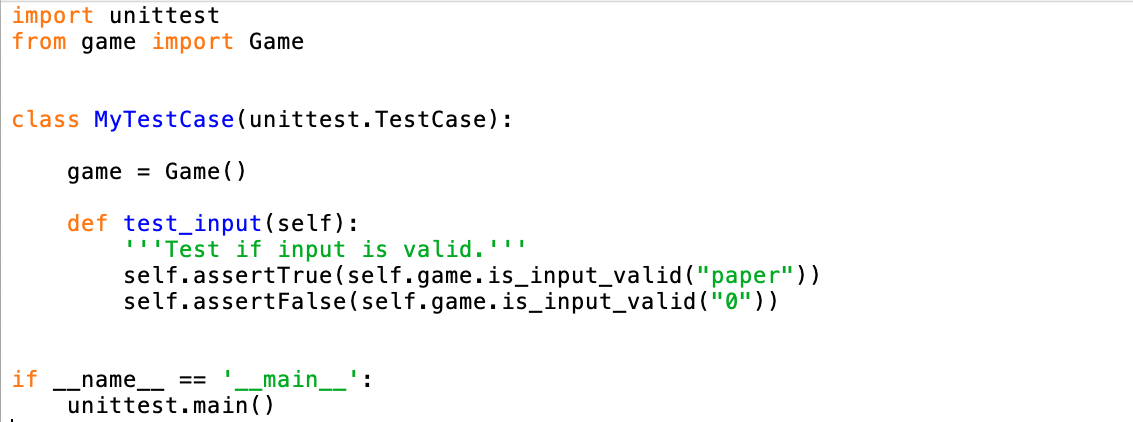
**Requirements**

1. Scissors win paper, paper wins rock, and rock wins scissors.
2. The user can decide what he/she wants to choose from rock, paper and scissors.
3. The computer randomly picks one of the options of rock, paper and scissors.
4. The winner of each round gets one point.
5. The game ends when one of the players earns five points, or the user chooses to quit.
6. The user can leave the game anytime he/she wants to.
7. The total number of rounds that have been played will be displayed.
8. An error input from the user will be prompted.

**Process**

**TDD**

To start, I wrote a test file with the filename test\_game.py. Then I add a test to see if the input function of the game works properly as shown below:

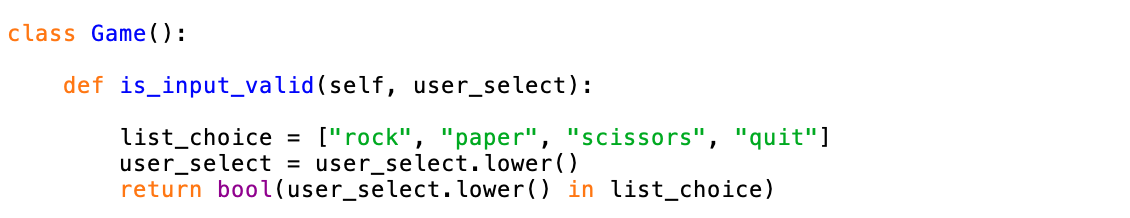


I ran the test file and it failed. It shows that the Game class is not defined yet, so I created a new empty file game.py as shown below.



I ran the test file again and this time I received a different error message AttributeError: 'Game' object has no attribute 'is\_valid\_input'.

Next, I created the is\_valid\_input functionality to the Game. Update the game.py file accordingly:

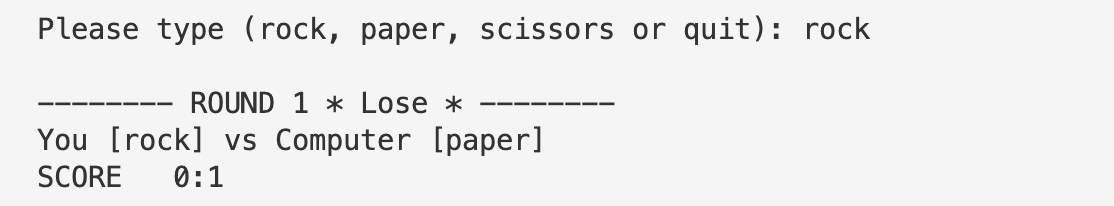


Then I ran the file again and I found the test is passed.

Continually, I set up more tests and add more functionalities until I have fulfilled all the requirements for the game.

**Game process**

Finally, I can run the file as below.

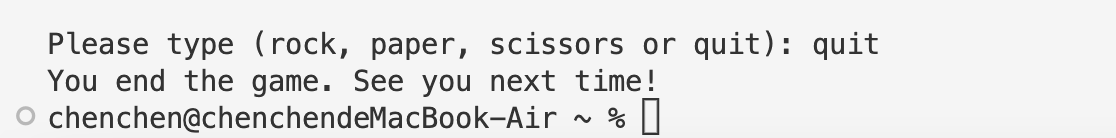


1. When I start the game, I will be asked what I want to choose from rock, paper and scissors.
2. The number of rounds and whether I win or lose will be displayed.
3. The computer randomly picks one of the options of scissors, paper and rock. I can see what I have chosen and what the computer has chosen.
4. The winner of this round will get one point.

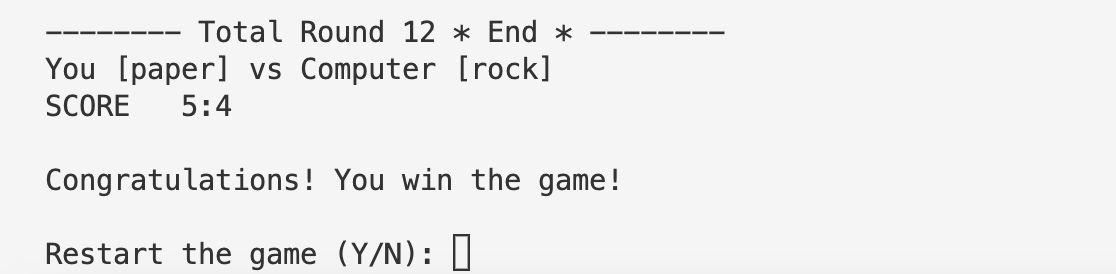
图表

描述已自动生成

1. A wrong input will be prompted.



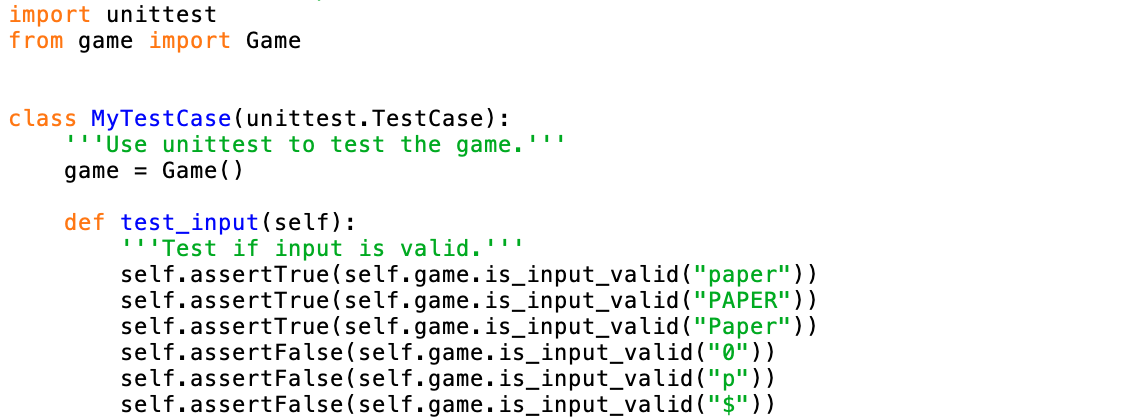
1. I can leave the game anytime by typing ‘quit’.



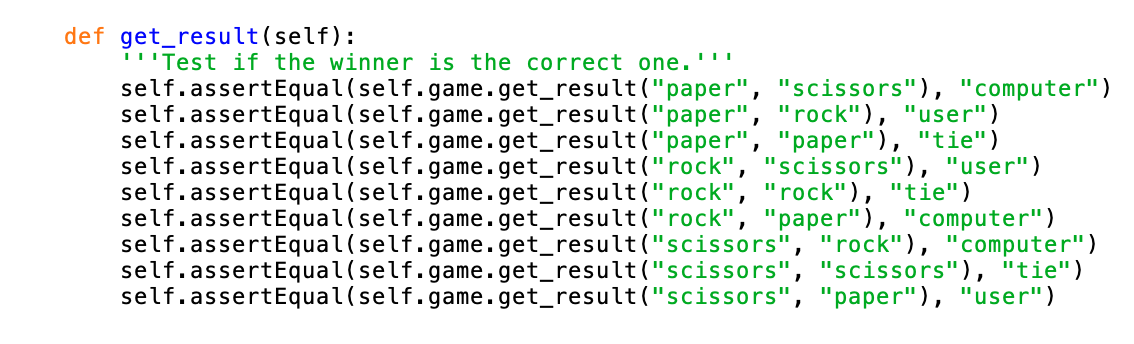
1. The first to get five points will be the winner.
2. I can choose if I want to restart the game.
3. The total rounds will be displayed at the end.

**Unittest**

I used unittest to test the game. I conducted 2 test cases to check if the game runs properly. The first test is designed for input from the user. To start the game, the user needs to choose rock, paper or scissors as his/her weapon. It’s allowed to type both upper-case letters and lower-case letters. But only the correct spelling will be accepted. Therefore, I wrote a test code to see if the input in upper-case letters and lower-case letters is passed. And I wrote a test code to check if the potential invalid input returns a false.



The second test is conducted to test if the winner of each round is correct. By following the principle that rock beats scissors, scissors beat paper and paper beats rock, the test will check if the return of every possible combination is correct. E.g., When the user input is paper and the computer randomly chooses scissors, it will return the winner of this round is “computer”.



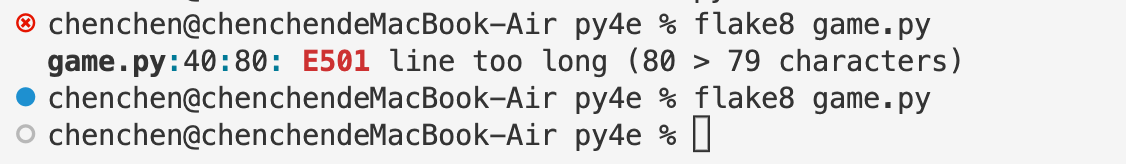
Finally, when I ran this test, I got the result that shows “OK”.

图片包含 文本

描述已自动生成

**Flake8**

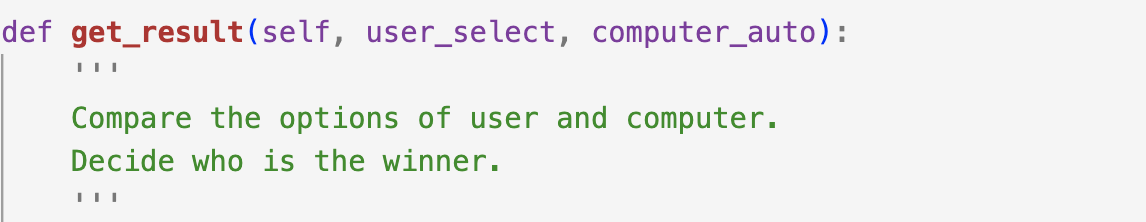
After I finished all the coding. I ran flake8 to test my files. I didn’t get many errors prompts from flake8. The only problem that I should be careful about is “line too long”. To solve this problem, I change some variable names. For example, I use “computer\_auto” instead of “computer\_select”. After some small changes, the test with Flake8 is passed.



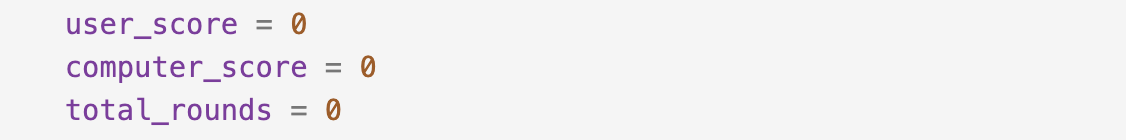
**Pylint**

Also, I used Pylint to test the files. The comments from Pylint are mainly as below. And I followed the suggestions and rewrite my code as shown in the screenshots.

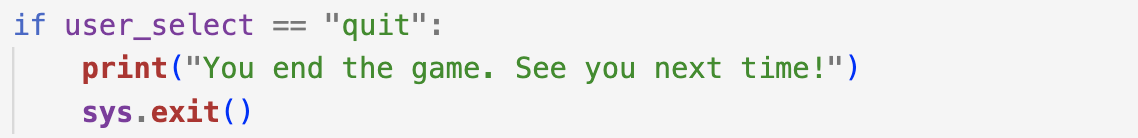
1. Missing function or method docstring.



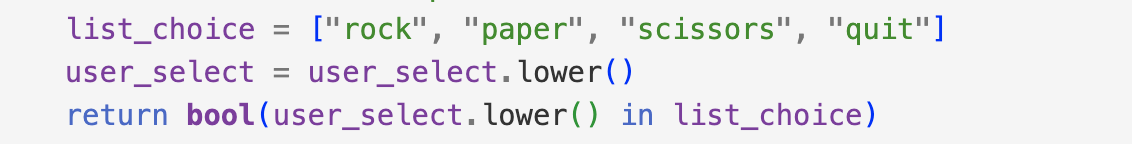
1. Variable name "playAgain" doesn't conform to snake\_case naming style.



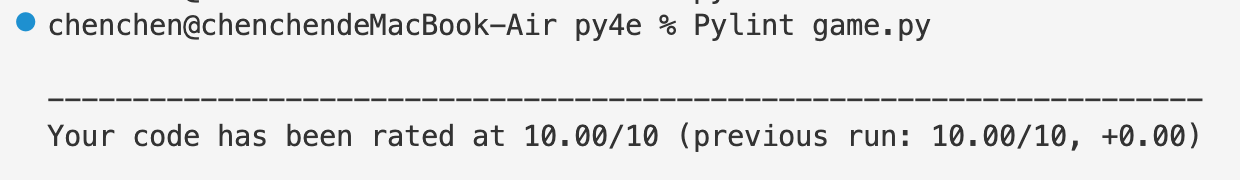
1. Consider using sys.exit() instead of quit().



1. Consider using bool() instead of “return True” and “return False”.



After I rewrote it, the test gets 10 points from Pylint.



**Conclusion**

**Lesson learnt**

This is my first program that is written in TDD. I changed the way I used to code. From considering how I should start to write the code to think about what I want to test in the first stage. It’s hard for me because I must have a coding structure in my mind to write a test, but I still need to start with the test, not coding the project. So I try to keep the first test simple and follow the error prompt to code a file that matches the test. When the simple test works, I spend time on more complex test cases. This is a process in which coding and testing progress together.

I learned from this project, not only the TDD but also some coding skills. Since I used some tools to help me test the files. Flake8 and Pylint give me some advice on my code. For example, I start to write docstring for every function and method. Also, I learned how to make the code precise and simple.

There is one more thing I want to improve, and that is user-friendly design. In this rock paper scissors game, the player must input exactly the same word to continue the game. But I realized I should put something simple to let the user choose, such as using numbers to represent rock, paper and scissors. I will pay more attention to user-friendly design in future coding.

**GitHub link**

[**https://github.com/chen1621/Rock-Paper-Scissors-Game.git**](https://github.com/chen1621/Rock-Paper-Scissors-Game.git)

**Appendix: Test Case**

Project Name: Rock Paper Scissors Game Module Name: Game Functionality

Created By: Chen Chen Created Date: 9/10/2022

Executed by: Chen Chen Executed Date:9/10/2022

| **Test Case ID** | **Description** | **Test Step** | **Preconditions** | **Test Data** | **Expected Result** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TC\_GAME\_01 | Verify functionality with valid input | Start the game |  |  | Able to see options of rock, paper and scissors | As expected | Pass |
|  |  | Enter valid selection | Valid input | Input: rock | Able to see the result of this round and continue the game | As expected | Pass |
|  |  | Enter valid selection | Valid input | Input: paper | Able to see the result of this round and continue the game | As expected | Pass |
|  |  | Enter valid selection | Valid input | Input: quit | Able to quit the game anytime | As expected | Pass |
|  |  | Enter valid input to restart the game | Valid input | Input: Y | Restart the game again | As expected | Pass |
| TC\_GAME\_02 | Verify functionality with invalid input | Start the game |  |  | Able to see options of rock, paper and scissors | As expected | Pass |
|  |  | Enter invalid input | Invalid input | Input: 0 | Able to see the error prompt | As expected | Pass |
|  |  | Enter invalid input | Invalid input | Input: p | Able to see the error prompt | As expected | Pass |
|  |  | Enter invalid input | Invalid input | Input: $ | Able to see the error prompt | As expected | Pass |
|  |  | Enter to restart the game or quit | Invalid input | Input: G | Able to select again until input a valid value | As expected | Pass |
| TC\_GAME\_03 | Verify winner selection function with valid input | Start the game |  |  | Able to see options of rock, paper and scissors | As expected | Pass |
|  |  | Input: rock | Valid input | Input: rock | Able to see what is the randomlt choice of computer.  Rock beats scissors  Scissors beat paper  Paper beats rock | As expected | Pass |
|  |  | Input: paper | Valid input | Input: paper | Able to see what is the randomlt choice of computer.  Rock beats scissors  Scissors beat paper  Paper beats rock | As expected | Pass |
|  |  | Input: scissors | Valid input | Input: scissors | Able to see what is the randomlt choice of computer.  Rock beats scissors  Scissors beat paper  Paper beats rock | As expected | Pass |
|  |  | Quit the game when all the options of computers have been checked | Valid input | Input: quit | Game ends. | As expected | Pass |