SSAD Assignment 3

Overview:

This Assignment will comprise 2 parts:
- Code Review of Mario
- Testing using Pytest and running tests with Tox

***Note* For both parts, you will be using the code you have written for your Assignment-1 Mario game. If you have not submitted the assignment, ping one of the TAs and you will be given a code base to work with.**

1) Part 1: Code Review (50M)

You will:

1.1) Find and report all bugs and code smells within the code.

1.2) Refactor your code to account for the bugs found, and follow proper practices like PEP8 standards.

How:

- Report known bugs in functionalities and code.
- Code smells can be found by reading through the code and looking for certain features.
  (Read: https://blog.codinghorror.com/code-smells/, https://en.wikipedia.org/wiki/Code_smell)
- Use **pylint** (https://www.pylint.org/) to check adherence to PEP8 standards.
- Refactoring:
  - Try to correct most of the smells and bugs(at least the small ones). Also, make your pylint scores per file > 9. (Check code format in the submission directory structure). Create a README.md to list out all the changes made to the code

Put findings of code smells and bugs within a document called: rollnum_codereview.pdf

The document should contain:

- At the Top:

    - Student Name:
    - Student Roll Number:
    - Code Review of Project:
    - Number of Lines Reviewed:
    - Number of Classes:
    - Number of Methods:
    - Number of Bugs Identified:
    - Number of Code Smells Identified:

- Rest:

  Each Bug and Code Smell should be reported as:

Bug Number	short description
#1	example description

Code smell	Short description
#category	example description

(The Bugs and Smells can be grouped together. ie. Putting all the bugs in one part of the document, then all the smells)

Part 1 Submission Format:

```
RollNumber_part1/
|
|—_old
|   |—code.py
|—_refactored
|   |—code.py                              - {Refactored version}
|—rollnum_codereview.pdf
|— README.md
```

Note

- You will/may have multiple files in your game. So organize accordingly(ie. There will be multiple files under both old and new)

2) Part 2: Testing (50M)

You will be:

2.1) Writing test cases using Pytest for each of the classes you have. Write test cases for all the classes implemented along with tests for functionality. Ensure that each functionality is tested(functionalities can be referred to from Assignment 1). You will then automate the testing process using Tox.

2.2) Use your tests to expose bugs, and then fix them.

How:

- Using a refactored copy of your code write tests using pytest (follow their recommended structuring and naming conventions).

- Create a setup.py and tox.ini file according to the tox documentation. It should be such that once you cd into the directory, typing tox should run everything required.

- Your tests should then be run to find bugs.(Ex. Expected behavior is that when Mario jumps, he will come down after 2 seconds. However, when he stands next to a block, he doesn't come back down. So you would write a test that checks whether or not Mario comes down, and that would fail in this instance). You should then attempt to fix these bugs.

- If bugs are found, your tests should be written such that when run on the refactored version(from part 1) they fail some tests. However once the bugs have been fixed, the tests pass. The bug-fixed code should be placed into a directory called final

Part 2 Submission Format:

RollNumber_part2/
|—_refactored - {From Part 1}
| |- code.py
| |- tests/
| |- tox.ini
| |- setup.py
|
|—_final
| |- code.py
| |- tests/
| |- tox.ini
| |- setup.py

Note
- The code.py here represents all the files you will have from your implementation of Mario.
- If you do not do the tox part, ensure that your tests are all inside a tests folder.
- If you do the tox part, you are free to structure your tests as you want, however on entering the <RollNumber>_part2/refactored or final directory and then running tox, the tests should work.

Overall Submission Format:
Submit a zip file: <RollNumber>_assignment3.zip

The zip file should contain:

RollNumber/
|
|—RollNumber_part1/
|—RollNumber_part2/