SSAD Assignment 3

Deadline: 27th October 2016

Overview:

The assignmemt will be in two parts.

1. Testing using Pytest and running tests with tox.
2. Code Reviews.

Pytest: 40 Points

Use your source file of Assignment 1 (i.e. Tetris game). Write testcases for all the classes implemented along with tests for functionality (Minimum **5** testcases for **each** functionality).

Make sure you cover these cases

1. Write testcases for all the three classes you were asked to implement in Assignment 1.
   - example tests:
     - `class Test_Block()` : This class covers the functionality of the Block class implemented during the Assignment1.
     - Capture the movement or rotation of Block.
     - When the rotation key is pressed, the Block rotates accordingly. Write tests to verify the same.
2. `setup.py` and `tox.ini` files must be setup. These are required for setting up tox.

Note: Those who have not done/completed Assignment 1 properly are requested to fill up this form here (same has been posted on the moodle forum as well) by Friday 14 October. You will be assigned a different code project, and you have to write test cases for the same.

Help : Follow the exact steps specified in the setup.py and tox.ini and tox should be up and running.

Code Review: 60 Points (10 Self Review + 50 Other Reviews)

Self Review: The students are required to do a self review of their Assignment 1 code. Note: Those who have not completed Assignment 1 properly are requested to do a code review of the alloted code project in Part 1 of this Assignment. This will count as self review.

Other Review: Students will be given access to tool(s) containing 3 code projects after Mid-2 exams. A student has to review all the three code projects using the given tool (more instructions will be sent out later, and explained in tutorial).

Code review consists of

- Entering review comments within the tool provided to each student.
- Find and report Bugs in the code written/given to you.
- Find and report whether the required functionality of Tetris game (self review) or another game (other review) is implemented or not.
- Find and report the code smells with in the code.
- Find and report whether the code written satisfies the coding standards set by PEP-8.

Format of the code review report

Student Name	<name>
Student Roll Number	<number>
Code Review of	<Self>/<Code Project Number>
# lines of code reviewed	<count>
# classes	<count>
# methods	<count>
#Bugs identified	<count>
#code smells identified	<count>

Bug Number	short description
#1	example description

Code smell	Short description
#category	example description

A total of four reports per student are to be submitted (1 report for self review and 3 for other code review) along with the review comments given on the tools provided.

Students are required to submit the reports into a google form, which will be provided later.

Help :

- students can run pylint to find the code smells and code style issues.
- Pep-8 guidelines
- Students can also refer to message codes and find similar bugs in the Tetris code / code project you're reviewing.
- Instructions to use the pylint:
  - `sudo apt-get install pylint`
  - `pylint <python files>`
  - Online Tutorial