Final Exam Programming Assignment

Description

To replace your final exam programming assessment, you are required to submit an open ended programming assignment. This assignment is to create any computer game using the concept of object-oriented programming. You will get additional mark if you can make use of a State Machine concept with libdw's `SM` class in your game.

Sample Code

You can download the some sample codes of simple games from these links:

- Hit the target: Simple game without state machines. See
  - Video
  - Sample Code.
- Spells: Text-based Wizard game with state machine example. See
  - Video
  - Sample Code.

You can use these to start. To run these sample code, follow this instruction:

1. Save it to your **Desktop**.
2. Open Anaconda Prompt (Windows) or Terminal (Mac OS).
3. Go to your **Desktop** by typing: `cd Desktop`.
4. Run the Python code by typing: `python game_template.py` or `python spells.py`.

Categories of Game

You can create game in any categories you can think of as long as it makes use of Object Oriented Programming. Below are some ideas:

- keyboard-based game, arcade
- guessing game
- GUI-based using Kivy

- text-based game on Terminal or Command prompt
- role-playing game (text-based or GUI)
- simulation game
- puzzle game
- idle game.
- etc

Deliverables

You need to submit on eDimension week 14, the following:

- Python's code of your game. **Your lines of code should be between 30 to 1000 lines**.
- README.md (Use markdown syntax if you want), which is a text file that describes the following:
  - Your game
  - How to play your game
  - Describe your code
- URL to a video showing how to play the game with all its features. **The video must not exceed 3 mins in duration**.

Assessment

After your code submission, you will be scheduled for a Web Interview with one instructor. This interview will be done on the:

- Monday, 4th of May 2020
- The schedule for the interview is given in this link: Interview Schedule.

Rubrics and Grades

You will be graded using the following rubrics:

0	1	2	3
Students have no idea on the code. Very	Students have minimal idea of the	Students have good idea of the code. Good	Students knows the code very well. Very

weak programming skills. Unable to answer questions.	code. Weak programming skills. Able to answer few questions.	programming skills. Able to answer all questions.	strong programming skills. Able to answer all questions clearly.
Game is the same as the template or very similar.	Game is simple with minimal modifications from some online code.	Game is moderately complex.	Game is designed well, has sufficient complexity.

You will get an extra of 1 pt if you make use of `SM` class from `libdw` package AND/OR your game is GUI-based using Kivy. This only applies if you either obtain 2 or 3 from the rubrics above.

Maximum points = 3 points + 1 extra point

Total weightage = 15% out of total scores.

Plagiarism Policy

We will submit your code to our plagiarism system based on MOSS. Plagiarism may result either in the following:

- failing the course, or
- being reported to University disciplinary committee which may result in being expelled by the University.

We take Integrity seriously and will not tolerate any plagiarism. You are not to copy other people's code either your peer or those from the internet.