ITWS-2 Assignment 3

DEADLINE: Mar 21, 2018 11:55 PM

Part-1: Object Oriented Programming

Space Invaders is a classic arcade video game created by Tomohiro Nishikado and released in 1978 - the port of which was also responsible for the massive popularity of gaming consoles in the 80s - and hence played a major role in pushing the gaming industry towards mainstream media. With this assignment, we give you the opportunity to re-create this iconic game in Python.

You'll be required to draw a 8x8 board on your terminal. Consider the board to be as such:

```
          1     2     3     4     5    6    7    8
          ______________________________
8        |                              |
7        |                              |
6        |                              |
5        |                              |
4        |                              |
3        |                              |
2        |2x1 2x2   ...    ...          |
1        |______________________________ |
```

You'll be required to make 3 different elements (all of them having their own classes:

a. **Spaceship**: Denote/draw this using any character (preferably a special character). The spaceship can only be moved horizontally on Row number 1. That is, it's movement is restricted from 1x1 to 1x8. Move it using key 'A' to move left and key 'D' to move it to the right.

b. **Aliens**: Again, denote them using any random character. The must be randomly spawned anywhere in rows 8 and 7. A new alien must be spawned every 10 seconds and each alien must last for 8 seconds, after which it self destructs.

c. **Missiles**: There are two types of missiles:
   i. Use the character 'i' to denote a missile. A missile is spawned each time the spacebar is clicked and is always spawned in the (row+1, column) block if the spaceship is in (row, column) when spacebar is clicked. For example, if the spaceship is in (1, 2) when spacebar is hit, the missile spawns in (2,2). The missile must move one row up every second. If a missile and alien collide, the alien gets destroyed.

ii. Use the character ‘I’ to denote the second form of missile. This missile is shot when the ‘S’ key is clicked, and unlike the first kind of bullet, will move two rows up every second. If this bullet collides with an alien, the alien will exist on the board for another 5 seconds. Also slightly change the look of the alien when this missile collides with it.

Use inheritance here: with a parent class ‘missile’, and two classes that then inherit from this.
If the user presses the ‘Q’ button, the game is quit.

Keep a counter for number of aliens shot down by missiles. Aliens that self-destruct obviously do not contribute to this counter!. This counter will be score of player.

Marks distribution:

- **Implementation**: You game should work. For example, **Spaceship** should not move out of grid, **grid** should be intact etc. Don’t worry, if you don’t have all the features. There will be partial marking.
- **Modular:** Your code should be modular. You should different files for different classes and should use ***imports*** to use them.
- **Code Quality:** Your submission should not have redundant code.
- **OOP Principles:** Your code will heavily be judged on OOP norms. You should use appropriate classes, rules etc.

Note: Use of ***pygame*** is optional. There will be no extra marks for using that. Your code will be judged based on above criteria only.

Part-2: Web scraping

Aalekh has started losing interest in research at CVIT and in his business too. The only thing left in his life is his girlfriend, Avni. She knows that only thing that can cheer him up are books. So she goes to amazon.in and amazon.com and start searching for bestseller books for him. Now, she doesn’t want to take any chance so she decides to buy best books available.

As Avni is not from programming background, you need to help her to scrap the list of all the bestseller books available on amazon.in and amazon.com.

URLs:

Amazon.in - https://www.amazon.in/gp/bestsellers/books/
Amazon.com - https://www.amazon.com/best-sellers-books-Amazon/zgbs/books/

Output example:

Name	URL	Author	Price	Number of Ratings	Average Rating

A Wrinkle in Time (Time Quintet)	https://www.amazon.com/Wrinkle-Time-Quintet/dp/0312367546?_encoding=UTF8&psc=1	Madeleine L'Engle	$4.77	2996	4.4 out of 5 stars

Note: If any field is missing for a particular book, mention **"Not available"** without quotes. Everything is case-sensitive. Also you need to include header(Name, URL, Author, Price….) in both the output files.

Deliverables:

You need to create folder of name "**question 2**" containing the following files:

```
question 2
├──── com_bestseller.py
├──── in_bestseller.py
├──── output
    ├──── com_book.csv
    └──── in_book.csv
```

com_bestseller.py: It will contain logic to scrape data from amazon.com.
in_bestseller.py: It will contain logic to scrape data from amazon.in.
output/com_book.csv: List of all bestseller books available on amazon.com in above mentioned format.
output/in_book.csv: List of all bestseller books available on amazon.in in above mentioned format.

Restrictions:

You can use python 2 or python 3(Recommended). You can use following python modules to complete the task :-

- BeautifulSoup - https://www.crummy.com/software/BeautifulSoup/bs4/doc/
- Mechanize - http://mechanize.readthedocs.io/en/latest/browser_api.html
- Requests - http://docs.python-requests.org/en/master/
- CSV - https://docs.python.org/3/library/csv.html

If you want to use any other module, please confirm with one of the TAs.

Bonus Marks:

There will be bonus marks if your code follows **PEP-8** standards.

Submission:

You need to submit **zip** of folder **"<RollNo>_Assignment2"** containing two folder on moodle:

```
201401007_Assignment2
├──── question 1
└──── question 2
```

- **“question 1”** - Contains deliverables of Part 1.
- **“question 2”** - Contains deliverables of Part 2.

Evaluation:

There will be **manual** evaluation for this assignment. **We will be running MOSS for this assignment so do not copy.**
There will be marks for code quality, so try to write as much as beautiful code as you can.

Copying in the Assignments can lead to Zero in the current assignment.