**Group Project Three – Forestview Gaming**

**Due: April 18th**

**Instructions:**

Complete the programming task and presentation completely, according to assignment rubric, by due date.

Upload to Assignment Folder (in Week 12)

Only ONE person needs to submit.

**Compress Python program(s**), PowerPoint, and other technical documentation, and upload to folder.

Total: 500 points (see rubric)

**Assignment:**

Use Python3 and Pygame to create your own arcade/ videogame game (see details).

**Assignment Details**

Use your creativity to come up with a 2-dimensional “Atari 2600” style arcade-style video game. This will be used by Forestview to potentially port this project output to other platforms.

The goal is to use all your Python knowledge acquired to be creative, according to guidelines.

**Guidelines**

* Game should involve as much of this Python’s course concepts as possible at one time.
* Game CODE and PLOT should be original; you can use elements of other games, so long as you put your own twist on it.
* Game must use past concepts, including Python Exceptions (try / except), and arbitrary arguments, such as function calls (\*args).
* Presentation: Live, in-class only; each group member should present at least one part of their game/activity.
* Details regarding usage of sprites, scoring, collision and event management, will be up to the individual group
* Documentation guidelines: state your case, future recommendations, code repurposing to other platforms (eg. Android) - Time New Roman, 12 pt font (1/2 pages)
* Application coding: doc strong throughout, comments appropriate, checked with Pylint, user input filtered
* For complete details, please see Rubric, next page

Here is how the assignment will be assessed: 500 points Total

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Exceeds Expectations** | **Meets Expectations** | **Falls Below Expectations** |
| **Game Concept** | **Completely original concept or a new take on an existing game**  **(+100 pts)** | **Somewhat creative take on an existing game or activity**  **(+90 pts)** | **Very familiar game or activity or a not very original concept**  **(+80 pts or below)** |
| **Front End** | **Web page tied to program code as Download and design has a 90’s motif. (+100 pts)** | **Web page tied to program code as Download and webpage as a very generic design**  **(+90 pts)** | **Web page has little or no changes from the default template provided**  **(+80 pts or below)** |
| **Documentation / Presentation** | **Submitted document/ in-class presentation matches Documentation Guidelines completely**  **(+100 pts)** | **Submitted document / in-class presentation matches Documentation Guidelines with some omissions**  **(+90 pts)** | **Submitted document / in-class presentation missing**  **(+80 pts or below)** |
| **Application Coding** | **Code uses many concepts taught in this class; code is documented, and conforms to Guidelines, such as using mentioned Python concepts (+100 pts)** | **Code uses many concepts taught in this class, but fails in one area, such as user forgetting to filter any output**  **(+90 pts)** | **Code uses many concepts taught in this class, but fails in more than one area**  **(+80 pts or below)** |
| **Fun Factor!** | **Game designer has researched game concepts, such as game addiction, and has integrated SOME aspect of that (and that is documented) (+100 pts)** | **Game is well designed, but there is no indication of user engagement.**  **(+90 pts)** | **Game is poorly designed.**  **(+80 pts or below)** |