CPSC 386 – Fall 2019

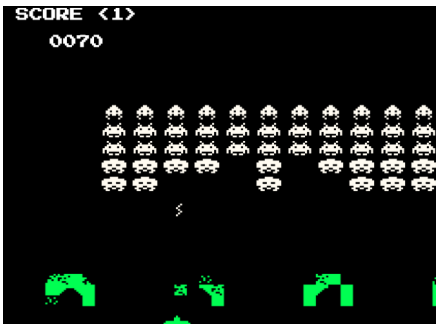
Introduction to Game Design & Production

Project Three: Space Invaders

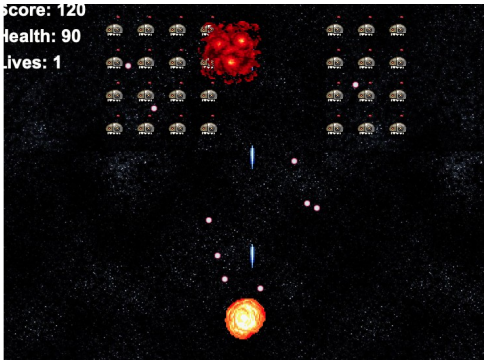
due at beginning of class – 386-01: 2 Oct (W)
386-02: 4 Oct (F)

In this assignment, you will create the **Space Invaders** game, based on the Alien Invasions! Code from project 2. The image resources you will need (ship, ship animation destruction, bunker, different types of aliens, alien animation destruction, and ufo destruction) will all have to be created using an Image editing tool such as Inkscape or Gimp. The audio resources you will need can be captured using an audio editor such as Audacity from an online version of Space Invaders.

Classic Space Invaders has several differences from the Alien Invasions! Game. You will need to complete:



1. You will need PyCharm, Pygame, and Python 3 installed.
2. Using classic Space Invaders as a guide, and using an Image editor such as Inkscape/Gimp, create the four types of aliens shown, the traditional Space Invaders ship, and bunkers to hide behind.
3. The aliens must include a simple, slow, two-state animation while they are moving (it looks better if alternating aliens are synchronized). Aliens must have a different image when they explode (could show a simple, fast animation as well).



4. A UFO should move across the screen at random intervals. It makes a continuous oscillating sound as it moves. If it is destroyed, it shows its (random) value instead of an explosion.

5. The ship must have a fast. animated explosion (8-12 frames) when it is destroyed. Be sure to move the pixels of the exploding parts around from frame to frame. (Note: the ship we used in Alien Invasions! Is not the same as that used in Classic Space Invaders.) The ship makes an exploding sound when hit.
6. Create a start screen, that shows the name of the game (in white and green), the aliens and their values, and the menus for Play Game and High Scores. The start screen should show at the beginning of each game, including if you have just lost a game.

7. Create a high scores screen, that shows the scores of the top ten players.
8. Add lasers to the aliens, so they can shoot back at the ship. Use a random number generator and a timer (pygame.time.get_ticks()) so they don't shoot too often.
9. Add bunkers to the game that the ship can hide behind. The bunker can be damaged by both the ship's and aliens' lasers. Use a random number generator to set the bunker's pixels to transparent when a laser strikes a part of the bunker to avoid a bite-out-of-a-sandwich look. Use the Python Imaging Library (PIL) to set the pixels.
10. Push the contents of your project to a new GitHub repository using a git client(e.g.,the git command-line client, GitHub Desktop, or GitHub for Atom). Do not submit files using drag-and- drop onto the repository web page, and do not push this assignment to the same repository as your previous homework assignments.

Submission

Turn in the code for this homework by uploading all of the Python source files you created, the images directory, and the sounds directory to a single public repository on GitHub. While you may discuss this homework assignment with other students. Work you submit must have been completed on your own.

To complete your submission, print the following sheet, fill out the spaces below, and submit it to the professor in class by the deadline. Failure to follow the instructions exactly will incur a 10% penalty on the grade for this assignment.

CPSC 386 Project Three: Space Invaders: due 2 Oct (-01), 4 Oct (-02) at beginning of class		
Your name:		
Repository: https://github.com/_____/ _____		
Finished	**Not Finished**	**Verify each of the following items and place a checkmark in the correct column. Each item incorrectly marked will incur a 5% penalty on the assignment's grade.**
❑	❑	The game has a startup screen that shows the name of the game, the values and images of the aliens, and has a Play Game and High Scores menu.
❑	❑	The high scores of the top ten players are stored on disk, and are displayed on a separate screen when the high scores menu item is selected.
❑	❑	The game has three types of movable aliens. Each type of alien has a different point value and animation. Aliens move back and forth, dropping lower when wall is hit.
❑	❑	A UFO alien moves across the screen at random, infrequent intervals. It was created using a pixel editor, and it shows its value when it is destroyed.
❑	❑	The aliens have simple, two-frame animations for movement. They also have a simple (3-4 frame) animation for destruction.
❑	❑	The ship at the bottom can be moved left or right, and can fire lasers at the aliens. Each time a laser hits an alien, it disappears and the player's score is incremented.
❑	❑	Ship created using a pixel editor, & has an animated (8-12 pixel frame) for destruction.
❑	❑	If all of the aliens are destroyed, a new level begins, with the aliens moving faster, and their point value is increased.
❑	❑	Bunkers allow the ship to hide from the aliens' lasers, but they are damaged by aliens' or ship's lasers (use PIL). Lasers can go through partially damaged bunkers.
❑	❑	Ominous bg music plays faster as the number of aliens left in a level decreases.
❑	❑	A Play button begins/restarts the game (when the mouse clicks on it).
❑	❑	After each group of aliens is destroyed, the player advances to a higher level, where the aliens move faster, their lasers are fired more frequently, and point values increase.
❑	❑	The Python code is object-oriented, with classes being created for the Ship, the Laser, the Alien, the Scoreboard, and the Settings. No issues are shown in PyCharm.
❑	❑	Project directory has been pushed using a GitHub client, not by manually dragging-and-dropping files onto the GitHub web page.
Comments:		