Coding Instructions

Abstract

These are the instructions for coding the game at home.

Curriculum

The curriculum presented in the Python class at KCC2018 followed this agenda

1. Download, install and configure software, including the game source code.
2. A brief presentation on the Python language, which is included in this repository as "KCC2018 Slideshow.pdf".
3. A coding session where students can write Python code to fill out the nearly-empty globaldefense.py file. This document describes this process.

Instructions

The Coding Steps.pdf document shows step-by-step examples of the code that must be entered into the globaldefense.py file to build the game. There are 11 steps.

Each step is runnable on its own and every step adds a tangible and complete aspect to the game.

Students should start with Step 1 (which is the base game with a blank window and no other functionality, then add the code from steps 2 through 11 one step at a time.

It is in the nature of this exercise that students (and teachers alike) will make spelling mistakes, typing errors, and even become confused about their progress. Moreover, mistakes may go unnoticed. When this happens, the game will likely be broken and it will be very hard for a beginner to figure out what is wrong and fix it before moving on to the next step. For this reason, the student's source code can be rescued and reset to the proper state at any given step. To do this, I've included a Python program that they can run which makes it easy to jump to a given step. If the student is following along with the Coding Steps.pdf document, they should know what step they want to go to, but there is a listing of the steps at the end of this document for additional reference.

To fix, reset or jump your code to any given step, use these command at the command console. This assumes that you downloaded the software to %userprofile%\globaldefense (Windows) or ~/globaldefense (Mac / Linux):

(Windows)
cd %userprofile%\globaldefense\src
python reset.py

(Mac)
cd ~/globaldefense/src
python3 reset.py

When prompted, enter a number for the step you want to jump to. If you enter 0, it will reset you to the finished game. This is the default state of the source code when you download it. So before beginning the exercise of going through the steps to build the game, you should run Python reset.py and enter 1 to go to the first step.

Steps

0 - The finished game. Go to this step if you just want to play it. And you will. It's awesome.
1 - An empty game loop and blank screen. The game does nothing except it watches when you click the X to close the window. This is your starting point.
2 - Sky. This adds a sky image to the background.
3 - Asteroids. This adds the asteroid field and asteroids will start to float around in space.
4 - Earth. This adds an Earth to the scene, but it doesn't do anything yet.
5 - Mouse Control. This makes the earth watch the mouse movement and points it at the mouse cursor.
6 - Keyboard States. This watches for the F key and fires continuously while it is held down. This means it also introduces the first weapon, the artillery gun. Bullets do not destroy anything yet.
7 - Keyboard Keypresses. This watches for the A, S and D buttons for full keypresses. It fires 3 more weapons, the direct missile, the garbage dump multigun and the missile array. These weapons are fired once for each keypress and they each have a cooldown period. Weapons still do not destroy anything.
8 - Asteroid Collisions. Weapons can now damage and destroy asteroids. A collision between a weapon and an asteroid causes damage. If the damage is severe enough, the large asteroids will break up into shards. If a shard is destroyed, there is nothing left.
9 - Earth Collisions. Asteroids can now damage and destroy the earth. A collision between an asteroid and the Earth will damage the earth. If the earth takes too much damage, the game is over.
10 - Heads-Up Display (HUD). This adds a heads-up display with the score on the left, Earth's remaining endurance in the middle, and a row of cooldowns for your weapons on the right.
11 - This adds background music.