PyGame Primer

Due April 21st, 2017
CSE332 Programming Paradigms, Spring 2017

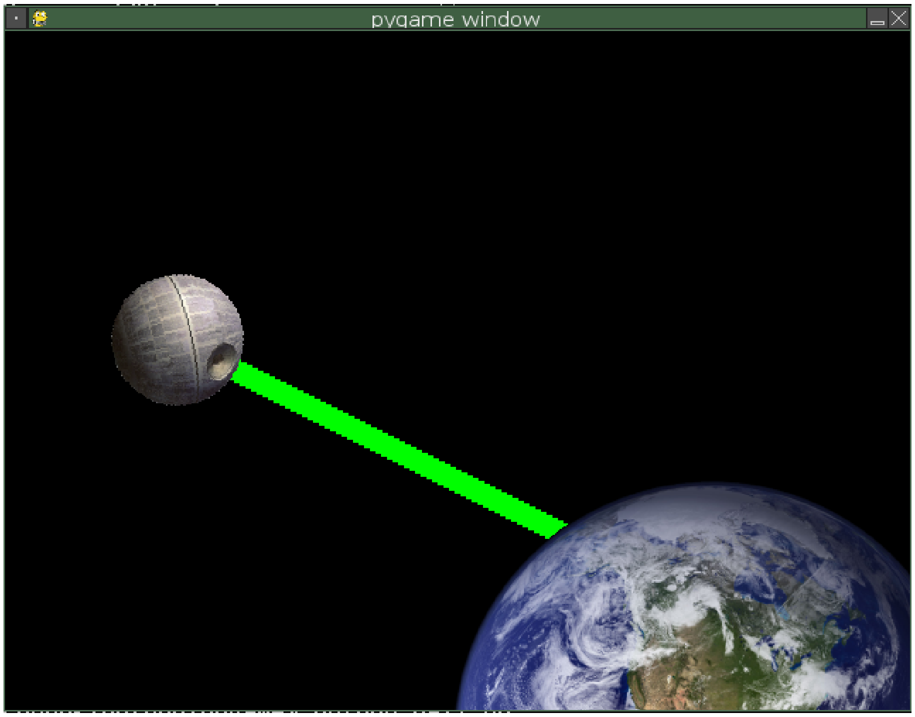
Overview Write a very simple game using PyGame. Your game may differ from the one discussed in class, but should include at least the following elements:

1. (4 points) A "player" avatar that is movable with the keyboard and rotates to face the mouse.
2. (4 points) An "enemy" avatar that either is static or moves on a fixed path.
3. (4 points) "Battle" functionality that fires a projectile, laser, etc.
4. (4 points) Enemy hitpoint reduction on projectile collision.
5. (4 points) Animated "disappear" (e.g., explosion) of enemy when hitpoints exhausted.

That's 20 points total! You may be creative with your design, or you may duplicate the "Deathstar Simulator" game from class because after all what could be more creative? The media files for that simple game are in the shared folder:

`/home/scratch/paradigms/deathstar/`

The "simulator":



Learning Objectives What you're really practicing here is the "clock tick" design paradigm, where the behavior of your program is controlled by a combination of events and clock ticks. Technically, this happens via a `tick()` function in each game object. The clock tick design is common not only in games, but also embedded systems and distributed programs. A major research area in computer science deals with how to ensure that the objects are properly synchronized on each tick, among other problems.

Turn in Put your code into a folder in your directory on `ash` as you have for earlier assignments: `turnin/deathstar/`

Include a `README` file that briefly (3-4 sentences) describes your game and how to play it, if it is different from the one demonstrated in class.

Resources

`http://pygame.org/docs/`
`http://pygame.org/docs/tut/intro/intro.html`
`http://pygame.org/docs/tut/chimp/ChimpLineByLine.html`
Trig. reminder for rotations:
`http://www.mathsisfun.com/algebra/trig-inverse-sin-cos-tan.html`