Zach Pomper

15-112

Dr. David Kosbie, Mentor Joyce (Xinyi Wang)

November-20-2016

Project Proposal: SolarSim

In as few words as possible, my term project boils down to: Space in Python (or Python in Space). If I were to more precisely describe my project, however, I’d say something like: A homebrewed 3D simulation of the solar system with adjustable timescales, semi-realistic modeling of gravitation, minimal use of exterior modules (other than, say PIL and Numpy, which are both essentially Q.O.L. modules for me, not the user), support for n-body simulation, and a baseline 3D framework that could potentially be portable to other Python projects.

Some background:

In my senior year, I took an awesome class on linear algebra, during which I did a presentation on the basic matrix principles of a 3D engine. At the same time, I was looking into integration techniques for n-body simulations and I stumbled upon an old paper describing Runge-Kutta integration (specifically RK4). I essentially had no idea what it was talking about, but it piqued my interest enough to motivate later attempts to integrate such a technique in Python. Naturally, for my term project, I wanted to combine step integration of O.D.E.’s and 3D principles somehow, so I figured a gravitational simulation would be a good application.

The Problem:

From a coding perspective a simulation of this kind presents a few hurdles:

* Needs an accurate integration method that approximates continuous-time integration
* Needs a basic framework for rotation/translation of a variable number of polygons
* Needs a way to project a 3D space onto a tkinter canvas for the user
* Needs a way to specify initial conditions through some kind of GUI

Tricks for implementation:

* Using NumPy arrays to accelerate matrix operations
* Use OOP techniques to store not-so-global-but-sorta-global-variables between many functions at once
* Use the 112 ™ tkinter wrapper to simplify data structures
* Use a library like PIL to implement resizable/rotatable image graphics (useful for, say, skyboxes)
* Loop through actual matrices (i.e. looping over row vectors) very sparingly--This operation is known to be expensive