Assignment 7.1

Animation – visualization with PyGame

While visualizing model output with a static plot can be useful, a dynamic view can be informative (and entertaining), particularly when tracking multiple objects.

In this exercise, you will build the basic tools needed to animate your orbit simulations.

1. All classes should conform with the class diagram on the next page.
2. In Model.py, create the SolarModel. In SolarModel.__init__(), hard code the ellipticities and semi-major axis for Mercury, Venus, Earth, Mars, and a comet. Give the comet a semi-major axis of 3.0 and an ellipticity of 0.9.
3. Test SolarModel by making a static plot of the five orbits. Plot the orbits as points and run the model for enough time so that all of the bodies complete one orbit.
4. Create Animate.py and Render.py in your Library directory.
5. In Animate.py implement the Animate class in accordance with the class diagram. You are free to add additional attributes and methods as you see fit.
6. In Render.py, implement the Render and RenderSolarSystem classes. Add whatever extra helper methods and attributes you deem useful.
7. In your Assignments directory, write a main function that constructs and runs an animation of your SolarModel.
8. Enjoy watching your tiny solar system. Fantasize about all you will do with the absolute power you have over its denizens.
9. Do the animated git.

Assignment 7.1