Solar System Orbit Simulator

This software allows the generation of a virtual solar system (SS) containing any desired amount of celestial physics bodies representing stars and planets. Currently, it enables one moon per planet. The software stores position and velocity data of the SS's consituent bodies and performs force calculations to approximate the influence of gravity on each body. Body-states are updated via numerical calculations (Euler-Cromer method). This facilitates the generation of accurate orbital paths, which the software then animates over time for the user's view.

Instructions for Build and Use

Steps to build and/or run the software:

1. Ensure the following python files are in the same folder:

- main.py
- solar_system.py
- physics_body.py
- star.py
- planet.py
- moon.py
- asteroid.py (Not functional)

2. Ensure that the following public libraries are installed onto your computer: Numpy (version 1.26.4) and MatPlotLib (version 3.7.0).

Instructions for using the software:

1. In main.py, look for the `planets` dictionary. Activate (via uncommenting the line) which planets you wish to see simulated in the animation.

2. Next, look for the `moons` dictionary. Activate (via uncommenting the line) which moons you wish to see simulated in the animation. Ensure host planets of ALL activated moons are also activated.

3. Run Main.py (Running other files will not perform any meaningful action). Animation of orbit simulations will play.

Development Environment

To recreate the development environment, you need the following software and/or libraries with the specified versions:

- Python 3.13
- Public Python libraries: MatPlotLib (version 3.7.0), Numpy (version 1.26.4)
- Local Python libraries: main.py, solar_system.py, physics_body.py, star.py, planet.py, moon.py, asteroid.py

Useful Websites to Learn More

I found these websites useful in developing this software:

- Article on Object Oriented Programming
- Free (limited) course on OOP concepts and applications in Python

Future Work

The following items I plan to fix, improve, and/or add to this project in the future:

- □ Generate physically accurate moon-orbits
- □ Improve superclass-subclass relationships (reduce copy-pasting of methods, etc.)
- □ Implement asteroid orbit physics
- □ Develop user-interface for simplified control of software
- □ Add additional orbit scenarios for user to simulate (three-body simulation, binary star system, etc.)