Python Project

1911700

1 Introduction and Theory

The aim of this project was to create a Python script that produced a three dimensional plot of any number of planets orbiting about a point based off Newtons Gravitational law, using the Euler-Cromer method. Furthermore this was to be imported this into a graphical user interface (GUI) to allow for straightforward use. Newtons gravitational law states that two bodies will attract each other with a force that is proportional to their masses and inversely proportional the distance between their centers squared. This has been expressed as a vector equation below.

$$\hat{F} = G\frac{M_1 m_2}{r^2}\hat{r} \tag{1}$$

In the above equation $\hat{F}$ is the force between the bodies in Newtons, G is the gravitational constant with a value of $6.674 \times 10^{-11}\ m^3\ kg^{-1}\ s^{-2}$, M_1 and m_2 are the masses in kilograms of the bodies, r is the distance between the centers in metres and $\hat{r}$ is a unit vector. This can be used for any number of bodies, with the total force being a sum of all forces acting on a body. Equating this to Newtons second law an equation for acceleration for multiple bodies can be determined.

$$\hat{a_r} = \sum_{i \neq j}^{j} \frac{Gm_j}{{r_{ij}}^{\frac{3}{2}}}\hat{r} \tag{2}$$

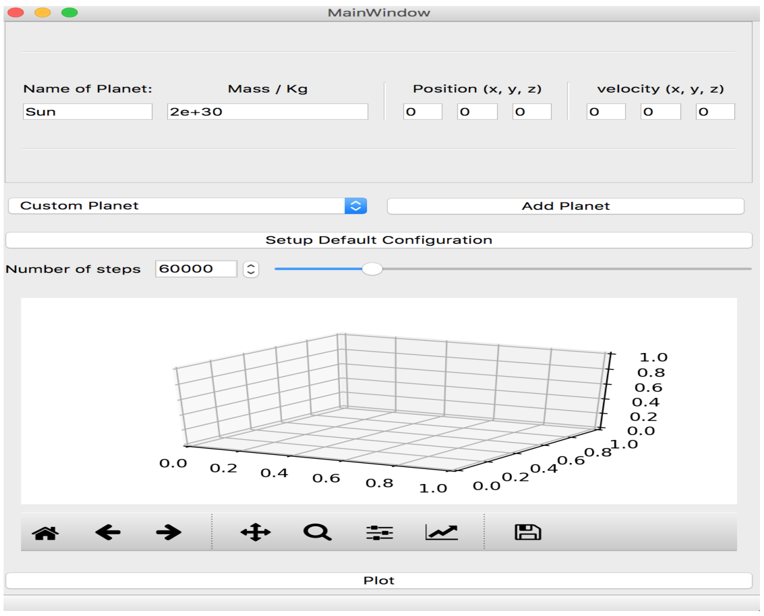
Where a_{ir} is the acceleration of a body i in the direction of r, m_j is the mass of a body and r_{ij} is the distance between the bodies. Using the acceleration the velocity and position can then be determined from their respective definitions. Because this was in three dimensions the distance r was expressed as seen below.

$$r = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{3}$$

The acceleration was resolved independently for each direction, velocity and position were also to be solved for in each direction. The method described above was used to produce plots of planetary orbits about a focal point, this being the Sun. Euler-Cromer is a method to determine approximate values of each parameter a step forward in time (dt), the above parameters were used to find these approximate values.

2 Instructions of Use

To run the program open the file main.py. Running this file will open the main window with an empty three dimensional plot as seen below.

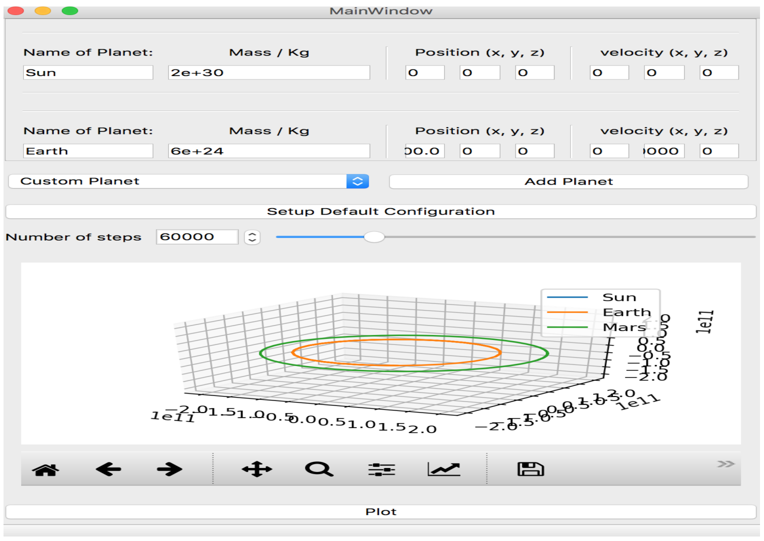


```
Name of Planet:  Mass / Kg   Position (x, y, z)   velocity (x, y, z)
Sun              2e+30       0   0   0            0   0   0

Number of steps  60000
```

Figure 1: Initial empty main window.

As can be seen above the Sun is automatically input to be plotted as without this focal point orbital trajectories are not produced. Using the labelled boxes, the label, mass, velocity and position can be changed. If the set up default configuration is clicked, the planets Mars and Earth will be input to be plotted. To view the orbital trajectories click the plot button.



```
Name of Planet:  Mass / Kg   Position (x, y, z)   velocity (x, y, z)
Sun              2e+30       0      0   0         0   0      0

Name of Planet:  Mass / Kg   Position (x, y, z)   velocity (x, y, z)
Earth            6e+24       00.0   0   0         0   000    0

Number of steps  60000
```

Figure 2: Default configuration input.

Planets can also be input using the combination box, custom planet information can be added by selecting an input and clicking the add planet button. The required data can be input into the labelled boxes above, along with this known planets such Venus and Jupiter can be quickly added with the same method. By selecting a planet the required information is input automatically into the correct fields, however these values can be changed freely.

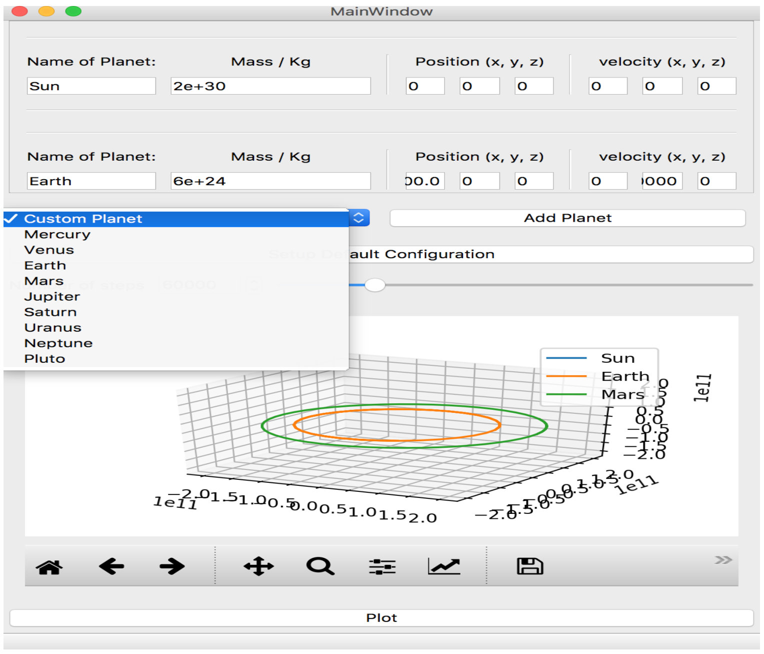


Figure 3: Changing planets position.

An example of the values being changed can be seen below. The Position of the Earth was changed to be in the z direction.

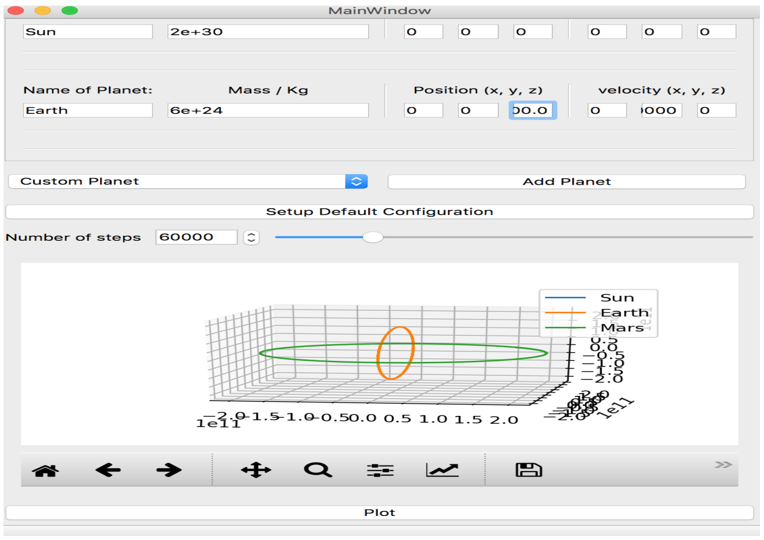
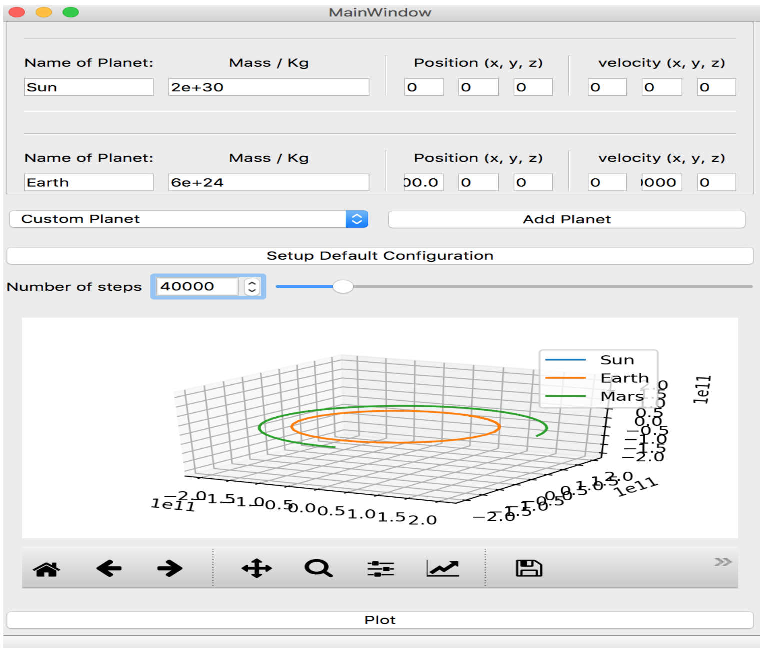


Figure 4: Combination box options.

The slider and spin box can be used to change the number of steps in time the program will use, one time step is equal to 100 seconds. These are connected, therefore changing the value in the spin box will change the value of the slider. The initial value has been set as 60000 steps, this was selected for the default configuration. The minimum amount of steps was set to 2500 steps, this was because any lower and the plot could not be seen. the maximum value was set to 300000. Below shows the default configuration with a smaller number of steps.



MainWindow

Name of Planet:	Mass / Kg	Position (x, y, z)			velocity (x, y, z)		
Sun	2e+30	0	0	0	0	0	0

Name of Planet:	Mass / Kg	Position (x, y, z)			velocity (x, y, z)		
Earth	6e+24	00.0	0	0	0	000	0

Number of steps: 40000

Figure 5: Changed number of steps.

As can be seen in the above image, when the time step is reduced the orbit of Mars is in incomplete. This shows that the number of time steps used is less that the period of one revolution.

3 Conclusion

In conclusion the program is able to plot the trajectories of any number of planets about a Focal point. Although this was the objective of this code, improvements could be made. One issue with the program is the limiting maximum number of steps. Although this value can be changed an issue is caused with the slider and the maximum value. Because of this limit the revolution of planets at great distances from the Sun cannot be seen. Another improvement that could be made is using a scaling factor as to make the values on the graph easier to read. Along with this, if the mass of the planet could be linked to the size rather than a constant point, the program would become easier to understand. Lastly the accuracy could be improved using the relativistic equation instead of the classical equation. This was the intended equation to be used, however this caused disc space issues that could not be resolved.