Student Name: Mai Ngo

Course Name and Number: DSC 430 Python Programming

Assignment Name and Number: DSC430_Assignment0901-ClosestPlanet

Date: 3/14/2023

Honor Statement: “I have not given or received any unauthorized assistance on this assignment."

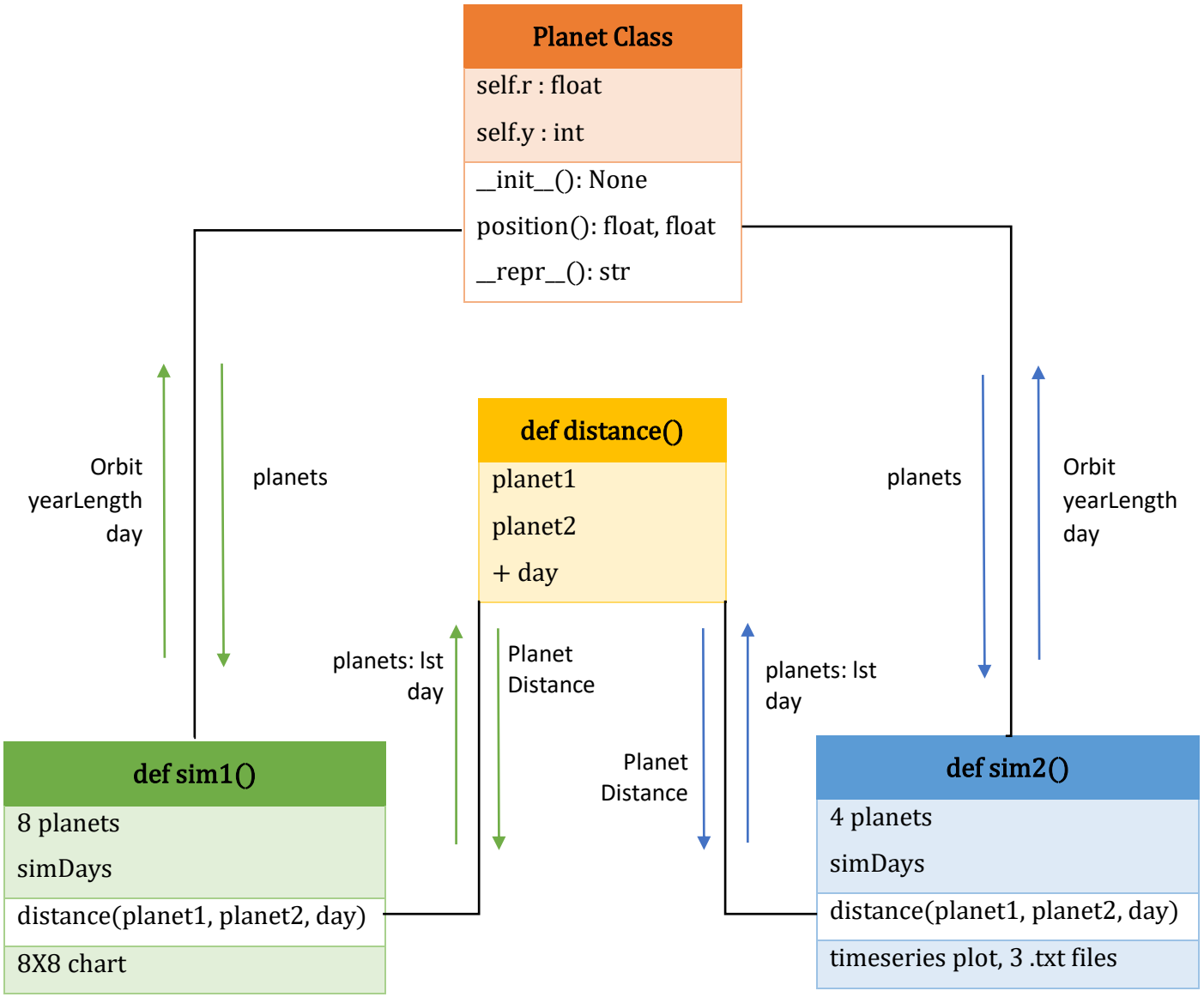
Final Report

Part A:

1) A well-documented top-down structure chart showing how both the planets and the simulation work and interact. Include any assumptions you made about your design.

- Design a solar system simulation by using object-oriented programming. Using the defs() required from homework PDF. By this time I have class Planet with def position(); additionally def distance()
- Create def sim1() for the first simulation. Input 8 planets, using Planet class to generate these planets into a list. Then iterate through every planet pair, every day within simulation’s time frame, calling def distance() to generate distances. Average distance is calculated within the function. Put result into a 8x8 chart using Numpy array(to generate) and Pandas (to set up data frame).
- Create def sim2() for the second simulation. Input 4 planets, using Planet class to generate these planets into a list. Then iterate through every day within simulation’s time frame. Getting the distance between Earth to those 3 planets, calling def distance() to generate the distances. Put them 3 different text files, convert to data frame and generate plot.
- I was trying to combine def sim1() and sim2() together into a Test class. But it got messy and time consuming, so I stopped.

Top-down structure chart



2) Create an 8x8 chart showing the average distance between all the planets. Show the chart in the documentation.

	Mercury	Venus	Earth	Mars	Jupiter	Saturn	Uranus	Neptune
Mercury	0.000000	7.165479	9.632488	14.416431	48.463277	88.934577	179.017224	288.010741
Venus	7.165479	0.000000	10.550754	15.002459	48.632216	89.025632	179.062106	288.038410
Earth	9.632488	10.550754	0.000000	15.771544	48.849381	89.142677	179.119881	288.075670
Mars	14.416431	15.002459	15.771544	0.000000	49.445326	89.464711	179.279206	288.170766
Jupiter	48.463277	48.632216	48.849381	49.445326	0.000000	95.490784	182.202017	289.948424
Saturn	88.934577	89.025632	89.142677	89.464711	95.490784	0.000000	190.081366	295.306532
Uranus	179.017224	179.062106	179.119881	179.279206	182.202017	190.081366	0.000000	320.885379
Neptune	288.010741	288.038410	288.075670	288.170766	289.948424	295.306532	320.885379	0.000000

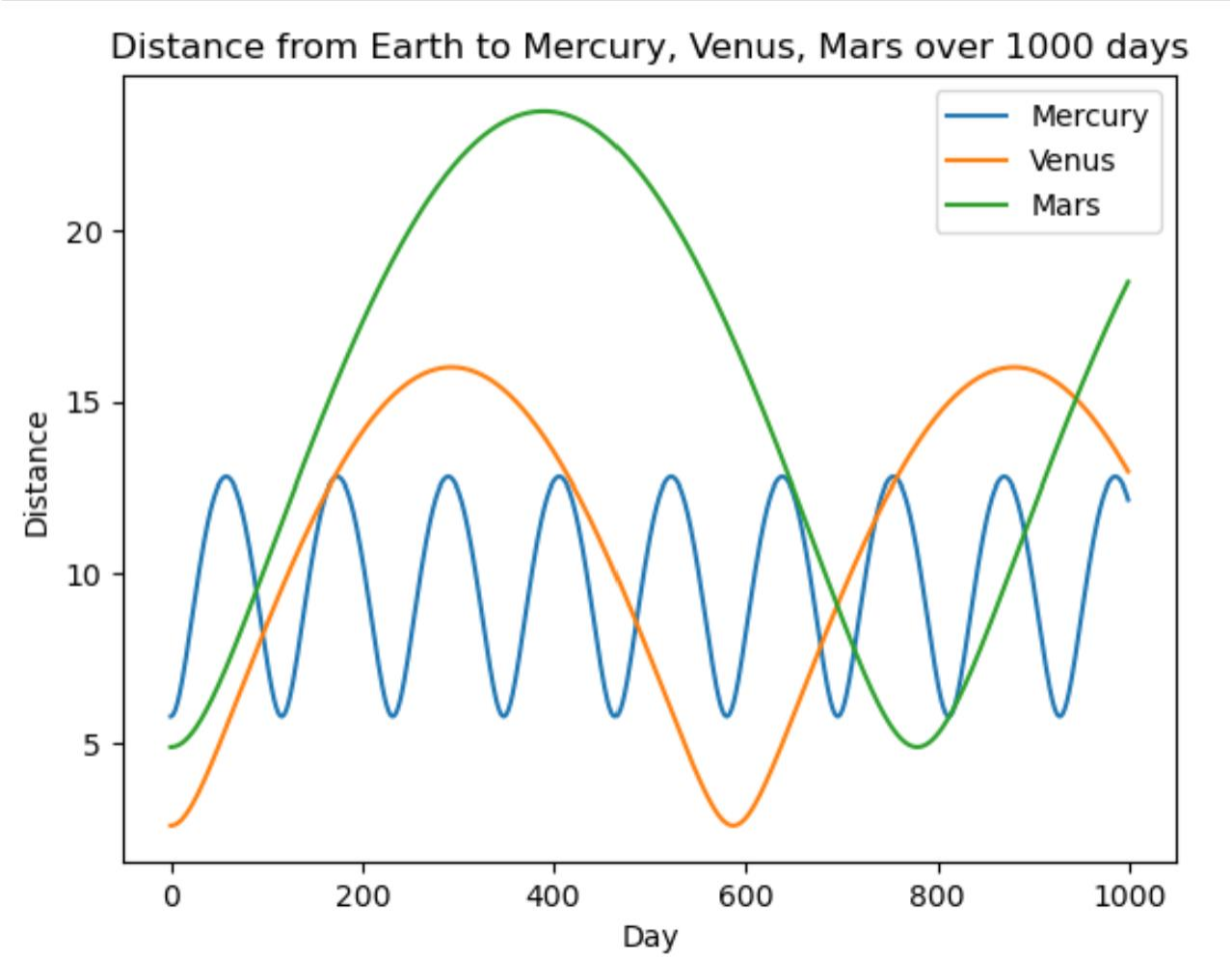
3) Which planet is on average closest to Earth? Did that result match your expectations? Explain.

- Based on the chart, on average Mercury is closest to Earth at 9.632488091502875.
- The result doesn't match my expectation because I look for fact from NASA website, Venus is the second planet from the Sun and is Earth's closest planetary neighbors.

4) Run another simulation, this time for only 1000 days, each day writing to a file the distance from Earth to Mercury, Venus, and Mars (i.e., one file for Earth to Mercury, another for Earth to Venus, and another for Earth to Mars). In the end, you should have a dataset with 1000 rows and 3 columns. Using Pandas and Matplotlib, create three timeseries. Show the timeseries in a plot in the documentation. Also describe the time-series and discuss them considering your findings in #3. '

- This simulation resulting a timeseries plot shows the distance between Earth and Mercury, Venus, and Mars over 1000 days.
- From the first simulation, it shown that Mercury is the closest planet to Earth.

- The timeseries plot shows that Mercury is fluctuating the most, more than the other two planets due to its small orbital radius. Thus, in this simulation Mercury completes the orbit fastest than other planets.
- Just for this stimulation. For example, Venus took almost 600 days to get close to Earth again, while Mercury already closest to Earth 5 times.



5) Describe three ways you could extend the simulation.

- Adding parameters: planet weight, velocity.
- Longer simDays, change the simulation time frame. To see if Mercury is still the closest planet to Earth. Change planet group: instead of Mercury, Venus and Mars. Should try Saturn, Uranus, Neptune.
- Adding an object/the moon: I do not have great knowledge about astronomy. But I know if there an object interferes with the solar system. Something would happen to the orbit.

Part B:

a) How efficient is your simulation? Can you do better?

- I think it is efficient to the requirements and scope of this homework. I know that if I account astronomical and physics variables, this simulation is like a baby demo.
- I think the simulation and code are efficient enough to help me understanding the concept of solar system and how planet orbits around the Sun.
- Obviously, there are many other ways to improve the simulation, use of physics, object interference, mass, velocity, etc. The only achievable improvement to the simulation I can think of now is adding velocity to each planet.

b) When computing the average distance between planets, would it be better to sample random days rather than iterating over every day for 1000 years?

- In general, using random is good with large data and studying/understanding the pattern. In this simulation, using much longer time period.

- In my opinion, iterating over every day is good to study past event. Which we can see the change day by day precisely. Analyzing the behavior and rebuild the model. Then use the new model with random to predict future pattern.

c) What was your original assumption regarding the closest planet to Earth? Did the results match your expectation? Does the definition of "closest" matter?

- My original assumption is Venus because that is what we all know. The result was showing Mercury.
- The definition of "closest" matters or not depends on the variables and how the simulation sets up. In this simulation, the 'closest' planet is defined as the one nearest to Earth on a given day. I did some Google search, and there are much more factors that concludes Venus is the closest planet to Earth: mass, speed, ellipse orbit, etc.