Tony Diep
u0934661
01/26/2017
2420
Design Document For N_Body Program
Ed Stephen Ancajas
I pledge that the work done here was my own and that I have learned how to write this program, such that I could throw it out and restart and finish it in a timely manner. I am not turning in any work that I cannot understand, describe, or recreate. I further acknowledge that I
contributed substantially to all code handed in and vouch for it's authenticity. (Tony Diep)

The purpose of this project is to reinforce the essential concepts of object-oriented programming (OOP): data abstraction, inheritance, encapsulation, and polymorphism. OOP is an essential and efficient tool because it eliminates redundancies in code and provides concise meaning in terms of program building. Although the skeleton of this project is provided, there are key classes left out and thus we must implement them on our own. We will need to implement the “Planet”, “Star”, and “Flotsam” classes because they are all types of Satellites which will represent the planets, sun, and floating debris respectively. They will interact with each other by making them relate to the superclass Satellite and thus will have similar characteristics such as moving in space. If a desktop that has a 2 gigahertz processor takes 1/10 of a second to process 1000 bodies, then I would say that the desktop would take at least 2/5 of a second to 2000 bodies. I believe there may be a relationship between the number of objects in the simulation and the run time it takes to construct that amount of bodies. Depending on the gigahertz and processor speed of any desktop/laptop used to run the simulation, the seconds may vary but they will have a similar pattern. Each body constructed into the program allocates a memory spot in the desktop/laptop so imagining that 1000 of these bodies takes a lot of memory. Similar to the Amoebas assignment, the number of circles does depend on the number of frames to even draw a single amoeba. As the number of gigahertz in a computer increases, the amount of seconds to process N amount of bodies can increase.

Ever since I skimmed through the project from all of the different classes, it seemed pretty intimidating at first because of the complexity of how the code is constructed. But after spending a little over 30 minutes, it seemed okay. But I will say I struggle to understand the concept behind gravitational forces and the physics concepts but the vectors I do understand. I understand there is a lot of mathematics involved but I will conceptually struggle behind it. Thus, I will expect to spend 20+ hours on this assignment trying to put the pieces together. From past projects in CS1410, I have spent at most 20 hours overall to work on similar GUI projects but that reinforce me to have good background and experience with GUI programs.

When looking at the code, I found three interesting things along the way. First, the project introduces a new interface I have not dealt before, which is the MouseWheelListener interface. But I will say it is quite similar to MouseListener and MouseMotionListener, so it’s not too far from completely new. Second, I noticed there are a lot of constants to keep track of so I can see why math works the way it does and that is a key “workhorse” to this project. If there is a slight miscalculation in the mathematics, then there will be some errors in the program since the satellites do depend on those calculations to have the correct movements. In the Solar_System_Facts class is a library full of public final constants. It is not a class that has a constructor nor is it an interface whatsoever, so the only usage for that class is the constants.

Tony Diep
u0934661
01/26/2017
2420
Design Document For N_Body Program
Ed Stephen Ancajas
I pledge that the work done here was my own and that I have learned how to write this program, such that I could throw it out and restart and finish it in a timely manner. I am not turning in any work that I cannot understand, describe, or recreate. I further acknowledge that I
contributed substantially to all code handed in and vouch for it's authenticity. (Tony Diep)

And finally, I found that some of the classes/objects take in more than 3 parameters. Some of them have 6 or more parameters which will be a lot to keep track of.

I think the key to succeeding the assignment is basically knowing what each class does and how to make them communicate to each other. It is also wise to keep track of the correct computations because mathematics is one of the key forces to getting the program to work.