Landon Crowther
CS 2420
Assignment 2: NBODY – Design Document

Purpose of this project: The purpose of this project is to enhance our knowledge of fundamental programming concepts, such as inheritance and polymorphism. These concepts will both be used in this assignment. For example, any object that will be in planetary motion can be considered a Satellite. Therefore, common methods must be created (abstractly) in the satellite class, and further refined in their unique classes. Then, when utilizing these classes, any objects extending from the Satellite class will have similar methods, and can be called as such. (polymorphism).
I think this assignment is also to help us figure out how write code and not have to start from scratch. Most of the framework is already here, and we don't need to understand exactly how everything works.

Classes:
--*Gemoetry_Vector:* Extends Point2D.double. Class will contain the "math" operations associated with vectors. For example, how to construct them, which direction they point, getting unit vectors, adding and subtracting vectors, multiplying & dividing, etc.
--*Main_Program:* This program simply sets up the GUI and initializes the program. This is straightforward.
Satellite: Extends JComponent. Class is listed as an abstract class. That is because any "moving component" is technically considered a satellite. This class will contain methods that need to be overridden by subclasses, but they are abstract because the user will never create a Satellite
--*Simulator_Loop_Thread:* Extends Thread. Interacts with the GUI and the Star_Field class. Keeps track of fps and positions
--*Solar_System_Facts:* Contains static final variables for the planets. This class exists to eliminate errors from having to hard-code the information about our solar system. In addition, it makes it so we don't have to memorize the information, we simply should call it properly.
--*Star_Field:* Extends JPanel and implements MouseMotion, MouseWheel, MouseListener, and ActionListener. This class is the "backbone" of the project, and manages of all the star objects, including planets. This also sets up the menu bar for the GUI, keeps track of mouse events, paints the planets, creates/paints flotsam, blackholes, etc.

Computer Speed: I'm guessing that if you were to double the number of objects in the sky, the processing time would also double. Additionally, a laptop that is half the speed would take twice as long to process. I would assume a relationship exists between number of object and processer speed, but I'm not sure if it's a 1:1 ratio. I'm sure there are many other factors that play into the speed.

Time estimation: Judging by the complexity of this assignment, I'm guessing that it'll take 8-10 hours to complete this project.

Interesting things: 1) Up until the lab today, I've never used an assignment.properties file. I think this is interesting and will be useful. 2) It makes sense to have a "facts" class for complex programs, rather than having to re hard-code them repeatedly. 3) It's interesting that the menu_bar is being constructed in the star field class, rather than the main program class. I would have assumed it would be created with the gui, but if it's interacting with the planets, it makes sense to have it there for the listener capabilities.

Additional thoughts: I'm hoping that I will be able to complete this assignment to the full specifications. Fulfilling each requirement helps me fully understand how these concepts work, and feel more comfortable with large programs.