NBody Simulation

Creating an Animation

Everything you've done so far is leading up to this moment. With only a bit more code, we'll get something very cool.

To create our simulation, we will discretize time (please do not mention this to Stephen Hawking). The idea is that at every discrete interval, we will be doing our calculations and once we have done our calculations for that time step, we will then update the values of our Bodys and then redraw the universe.

Finish your main method by adding the following:

- Enable double buffering by calling enableDoubleBuffering(). This is a graphics technique to prevent flickering in the animation. This should be just a single method call, so you shouldn't do anything complicated here. You can see an example in StdDrawDemo.java. Here's the official documentation that explains it in a few sentences. You don't have to understand this for CS61B. Just know that if you don't call this function, any attempt at smooth animation will look bad and flickery (remove it and see what happens!).
    - When double buffering is enabled by calling enableDoubleBuffering(), all drawing takes place on the offscreen canvas. The offscreen canvas is not displayed. Only when you call show() does your drawing get copied from the offscreen canvas to the onscreen canvas, where it is displayed in the standard drawing

window. You can think of double buffering as collecting all of the lines, points, shapes, and text that you tell it to draw, and then drawing them all simultaneously, upon request.

- Create a variable that represents time. Set it to 0. Set up a loop to loop until this time variable reaches (and includes) the T from above.
- For each time through the loop, do the following:
  - Create an xForces array and yForces array.
  - Calculate the net x and y forces for each Body, storing these in the xForces and yForces arrays respectively.
  - After calculating the net forces for every Body, call update on each of the Bodys. This will update each body’s position, velocity, and acceleration.
  - Draw the background image.
  - Draw all of the Bodys.
  - Show the offscreen buffer (see the show method of StdDraw).
  - Pause the animation for 10 milliseconds (see the pause method of StdDraw). You may need to tweak this on your computer.
  - Increase your time variable by dt.

Important: For each time through the main loop, do not make any calls to update until all forces have been calculated and safely stored in xForces and yForces. For example, don’t call bodies[0].update() until after the entire xForces and yForces arrays are done! The difference is subtle, but the autograder will be upset if you call bodies[0].update before you calculate xForces[1] and yForces[1].

Compile and test your program:

```
javac NBody.java
java NBody 157788000.0 25000.0 data/planets.txt
```

Make sure to also try out some of the other simulations, which can all be found in the data directory. Some of them are very cool.