CIS 110 Recitation NBody and Parallel Arrays

Tuesday, July 11th

NBody: Overview

• Simulate the motion of n number of bodies or particles in a plane undergoing gravitational forces from one another.



- Each particle has four important properties for simulating its motion:
 - Position -> point in space
 - Velocity -> change in position over time
 - Acceleration -> change in velocity over time
 - Net force applied to it -> gravitational pull from other planets
- The last three properties are split up into x and y components.

- Each frame of your simulation is separated by an amount of time called the timeStep.
- Given each particle's properties at one frame, you can calculate each particle's x and y velocity at the next frame.
- Assume velocity is constant during the timeStep in between frames:
 - $px_{n+1} = px_n + \Delta t * vx_{n+1}$ (where n denotes the frame)
 - What is py_{n+1} ?

- Remember that each particle interacts with each other particle
- Outer loop: For each particle, use the net force exerted on it to calculate velocity.

$$\circ a_x = F_x / m; a_y = F_y / m$$

$$\circ vx_{n+1} = vx_n + \Delta t * ax_n; vy_{n+1} = vy_n + \Delta t * ay_n \text{ (assuming constant a)}$$

• Inner loop: For each particle add up every other particle's contribution to the net force.

• Calculating each particle's contribution to net force:



NBody: Command-Line Arguments

- NBody takes three command-line arguments to run:
 - A double, simulationTime, that represents the time at which the simulation should end.
 - A double, timeStep, that represents the time quantum.
 - A String, filename, that contains the filename of the universe information.

NBody: Particle Specifications

• Universe specifications are given by specially formatted text files that look like so:

5 num	Particles				
2.50e+11 rad	ius				
5.97400e+24	1.49600e+11	0.00000e+00	0.00000e+00	2.98000e+04	earth.gif
6.41900e+23	2.27900e+11	0.00000e+00	0.00000e+00	2.41000e+04	mars.gif
3.30200e+23	5.79000e+10	0.00000e+00	0.00000e+00	4.79000e+04	mercury.gif
1.98900e+30	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	sun.gif
4.86900e+24	1.08200e+11	0.00000e+00	0.00000e+00	3.50000e+04	venus.gif
m[]	px[]	ру[]	v×[]	vy[]	img[]

Note: the blue text is not actually in the document and is just for labeling.

NBody: Particle Specifications

- Put numParticles and radius in an int and double variable respectively.
- Declare the six arrays labeled and then use a loop to populate them with the values in the document.
- How large should each array be and of what type?

NBody: Particle Specifications

• To pull information from the file you must initialize file reader you will call inStream:

In inStream = new In(filename); // creates a variable inStream to read from the file

• There are a set of methods you can use with this variable:

boolean b = inStream.isEmpty(); // boolean value that is true if there are no more values, false otherwise int i = inStream.readInt(); // reads in an int from inStream double d = inStream.readDouble(); // reads in a double from inStream boolean b = inStream.readBoolean(); // reads in a boolean from inStream String s = inStream.readString(); // reads in a string from inStream String s = inStream.readLine(); // reads in an entire line from inStream String s = inStream.readLine(); // reads in the entire file from inStream

Parallel Arrays: Overview

- An implicit data-structure used to represent a set of records, where a record is a collection of fields
 - Example: A student's name, Pennkey, and Penn ID #.
- Parallel arrays are a set of arrays with equal length whose indices correspond to one another.
 - Example: One array for names, one for PennKey's, and one lastly for Penn ID #'s.

Parallel Arrays: Example

```
<sup>1</sup> public class Students {
      public static void main(String[] args) {
2
3
4
           String[] names = new String[10];
           String[] pennKeys = new String[10];
5
           int[] pennIDs = new int[10];
6
7
           //ideally populate the arrays with values
8
9
10
           for (int i = 0; i < names.length; i++) {</pre>
11
               System.out.println("Name: " + names[i]);
12
               System.out.println(" PennKey: " + pennKeys[i]);
13
               System.out.println(" Penn ID: " + pennIDs[i]);
14
15
16 }
```

Parallel Arrays: Exercise

- Initialize three arrays of equal length
 - Price
 - Quantity
 - Revenue
- Populate the price and quantity arrays will values of your choosing.
- Populate the revenue array using the formula:
 - Price X Quantity = Revenue
- Lastly, print out each calculation.

