

COS 126 Precept

Will Clarkson
February 17, 2009

Outline

- ▶ Questions
- ▶ Standard Input (StdIn.java)
- ▶ Activity
- ▶ Redirection
- ▶ Nbody Overview



Standard Input

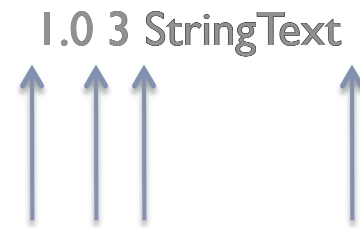
- ▶ Easy way to read in lots of data
- ▶ Traditionally read from text file
- ▶ Automatically can give us next int, double ,String,
 - ▶ Must use StdIn.java
- ▶ **Examples**
 - ▶ `int x = StdIn.readInt();`
 - ▶ `double x = StdIn.readDouble();`
 - ▶ `String x = StdIn.readString();`



Reading in Standard Input

- ▶ Read in until nothing left to read
 - ▶ `StdIn.isEmpty()`
 - ▶ True if no more input
 - ▶ False as long as we could read something else
 - ▶ Reads until End Of File marker (EOF)

```
double x = StdIn.readDouble();  
int y = StdIn.readInt();  
String z = StdIn.readString();  
  
boolean EOF = StdIn.isEmpty();
```



Example: Average Another Program's Output

- ▶ **RandomIntegers.java**
 - ▶ Takes in arguments N,M.
 - ▶ Prints out M numbers uniformly chosen between [1,N]
- ▶ **Average.java**
 - ▶ Reads in some number of integers
 - ▶ Calculates running sum
 - ▶ After all read in, prints out the average of inputs



Redirection of Standard I/O

- ▶ **Does not work in Dr. Java**
 - ▶ Don't use Dr. Java to run/test your Nbody assignment
- ▶ **Must use Terminal/Command Window**
 - ▶ See Assignment 2 page for Windows/Mac/Unix details
- ▶ **To save output of program into file**
 - ▶ `java OutputGeneratingProgram > all_output.txt`
 - ▶ `Java RandomIntegers 5 10000 > random_numbers.txt`
- ▶ **To read in contents of file using Standard Input**
 - ▶ `java ProgramNeedingInput < input_file.txt`
 - ▶ `java Average < random_numbers.txt`



Activity

- ▶ **Students.java**



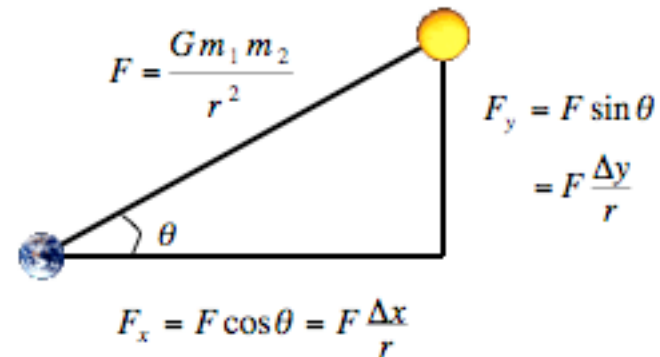
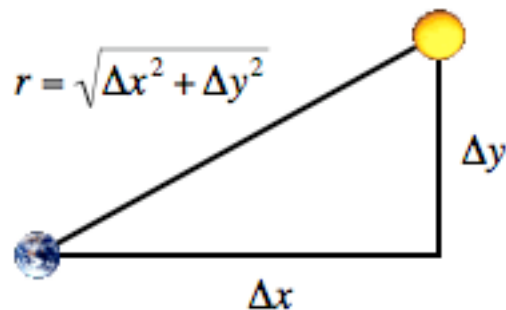
Formatted Output

- ▶ **See Section 1.5**
 - ▶ Not needed for Nbody assignment
 - ▶ May be on test, so look it over



NBody

- ▶ Please Please Please read this for Thursday
- ▶ Make sure you can download the files
- ▶ We are going to model the physics of a universe using Newton's law of gravitation



Nbody Example



Stages to completing NBody

- ▶ Read in input information
 - ▶ And make sure you can print it back out
 - ▶ Useful for debugging: see checklist for details
- ▶ Draw planets on screen
- ▶ Implement position updating
- ▶ Implement Force/Acceleration calculation



Formulas

- ▶ Calculate force between each pair of planets
 - ▶ Think nested loops
 - ▶ Careful of divide by zero (when would this happen)
- ▶ Once we know sum of all forces acting on an object
 - ▶ Can calculate acceleration
 - ▶ $a = F/m$;
 - ▶ Can update velocity
 - ▶ $v = v + dt*a$
 - ▶ Can update position
 - ▶ $p = p + dt*v$
- ▶ Remember to maintain velocities (x,y) and positions(x,y)
- ▶ Accelerations are calculated fresh during repeat of while loop



Nbody.java Outline

- ▶ Read in planet data from Standard input
- ▶ Set Scaling (ignore for now)
- ▶ While(true)
 - ▶ (3) Calculate net force on each planet
 - ▶ Nested for loops
 - ▶ (2) For each planet
 - ▶ Calculate acceleration (from force)
 - ▶ Update velocity
 - ▶ Update position
 - ▶ (1) For each planet
 - ▶ Draw planet at new position



Good Luck

