

COS126 Precept

Will Clarkson
February 18, 2009

Outline

- ▶ Questions about class
- ▶ Standard Draw
- ▶ Activity
- ▶ Nbody Outline/Discussion



Standard Draw (StdDraw.java)

- ▶ **Allows us to draw things on the screen**
 - ▶ API (Application Programming Interface) on pg 718
 - ▶ See Booksite 1.5 for FULL API
- ▶ **Main functions you will need for Nbody**
 - ▶ `StdDraw.setXscale(xmin,xmax);`
 - ▶ `StdDraw.setYscale(ymin,ymax);`
 - ▶ `StdDraw.clear();`
 - ▶ `StdDraw.picture(x,y,filename);`
 - ▶ `StdDraw.show(int t);`



Bouncing Ball (as seen in class)

- ▶ **DeluxeBouncingBall.java**
 - ▶ Uses some of the StdDraw API functions we saw earlier
 - ▶ Looks much more impressive
 - ▶ Outline is similar to what your Nbody.java program will do



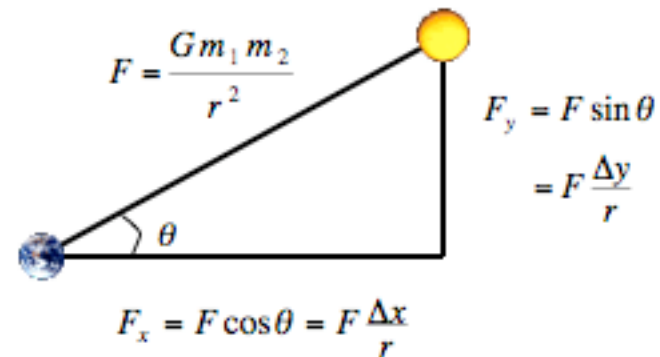
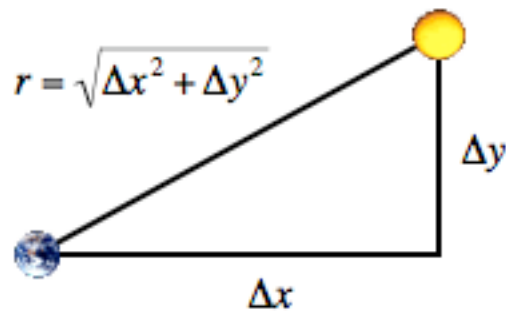
Activity??

- ▶ `/**`
- ▶ `* Draw a filled square of side length $2r$, centered on (x, y) ;
degenerate to pixel if small`
- ▶ `* @param x the x-coordinate of the center of the square`
- ▶ `* @param y the y-coordinate of the center of the square`
- ▶ `* @param r radius is half the length of any side of the
square`
- ▶ `* @throws RuntimeException if r is negative`
- ▶ `*/`
- ▶ `public static void filledSquare(double x, double y, double r)`



NBody

- ▶ We are going to model the physics of a universe using Newton's law of gravitation



- ▶ This assignment is harder to debug than previous ones
 - ▶ Please follow our outline of how best to complete this
-



3 Stages

- ▶ **Initialize Data Structures**

- ▶ Arrays for position (x and y), velocity (x and y), force (x and y), mass (x and y), and picture
- ▶ How many planets do you have?

- ▶ **Read in the data from standard input, into arrays**

- ▶ Similar to Students.java on Tuesday

- ▶ **Print the data back out**

- ▶ Verifies you did the reading correctly
- ▶ Will be helpful for later debugging after you change position/velocities



Planets governed by physics

▶ How do you know what to draw?

▶ File format:

N

R

xpos ypos xvel yvel mass image_name

... (there are N of these)

xpos ypos xvel yvel mass image_name

Comments can go below the data



3 Stages

- ▶ **Initialize Data Structures**

- ▶ How many planets do you have?
- ▶ Arrays for position (x and y), velocity (x and y), force (x and y), mass (x and y), and picture

- ▶ **Read in the data from standard input, into arrays**

- ▶ This is similar to Students.java on Tuesday

- ▶ **Print the data back out**

- ▶ Verifies you did the reading correctly
- ▶ Will be helpful for later debugging



Remaining Stages to Complete

- ▶ Read in and Initialize arrays
 - ▶ Remember to set $X/Yscale(-R,+R)$
- ▶ Loop forever [while(true)]
 - ▶ 3 Add in force of attraction between planets
 - ▶ 2 Move planets based on velocities only
 - ▶ 1 Draw background and planets on screen in fixed positions (no movement)



Stage 1: Drawing Planets on Screen

- ▶ We will use `StdDraw.picture(x,y, picture_name)` to draw planets
- ▶ Put background (starfield) at 0,0
- ▶ For each planet
 - ▶ Display the picture at planets position



Stage 2: Move planets based on velocities

- ▶ Inside while(true) loop:
 - ▶ For each planet
 - ▶ we can update velocity
 - $v = v + dt*a$
 - ▶ Then we can update position
 - $p = p + dt*v$
 - ▶ Then draw planets on screen at new position



Stage 3: Use Force Formulas

- ▶ Calculate force between each pair of planets
 - ▶ Think nested for 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
 - ▶ $p = p + dt*v$
- ▶ Forces and accelerations are calculated fresh during each 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 forces acting on each planet)
 - ▶ Update velocity
 - ▶ Update position
 - ▶ (1) For each planet
 - ▶ Draw planet at new position



FAQ

- ▶ Stages 1 and 2 work, but during stage 3 my planets fly off the screen?
 - ▶ Check to make sure your delta x and delta y are computed correctly. (i.e. switch the order during the subtraction)
- ▶ Screen Flickers?
 - ▶ Use `StdDraw.show(30);`
- ▶ How do I play audio?
 - ▶ Put the line: `StdAudio.play("file.mid");` just before you start your while loop.

