Chris Piech
CS 106A

Section #7
May 23, 2018

Section #7: Lesson Plan

Written by Julia Daniel

This section is primarily about classes, although there's some problems that give practice with interactors as well. The goal for this section is to make sure students solidly understand what classes are for (serving as blueprints for custom variable types) and how to define and use them. Ideally, by the end of section they will have clarified and refined a lot of the concepts/skills they need for NameSurfer, which is due next Wednesday and relies heavily on getting classes to work together.

Key Concepts

- **`JComponents`** (Interactors)
  - Buttons, text fields, labels
  - Use of **`init()`** method *only* to set up the window
  - Asynchronous control flow, very similar to **`MouseEvents`**
- Classes
  - Templates or bluprints that allow us to define custom variable types
  - Need a constructor, instance variables (almost always), and methods
  - Promote program decomposition into multiple files with different jobs
  - Encapsulation: public vs. private and the wall of abstraction
  - Use of **`this`** when assigning values to instance variables, to disambiguate

Lecture Recap

Here's quick summaries of the relevant lectures for this week's section. Please check out the slides in full if you haven't been in lecture.

19. Interactors
    Buttons, text fields, labels, init() vs. run(), asynchronicity
20. Classes Part 1
    Templates/blueprints for custom variable types
    Require subvariables (ivars), methods, constructor
    Bank Account example
    Encapsulation: public vs. private variables & methods
21. Classes Part 2
    Inner workings of a GRect as example of a class
    Bouncing balls example
    Email client example
    Wall of abstraction
22. Data Visualization
    Very close parallels to NameSurfer's use of classes!

Classes Problems (3, 4)

Problems 3 and 4 are focused on the conceptual foundations of classes, which is the most critical takeaway for students this week. Problem 4 is the highest priority. It involves writing a program and a class that interact with each other. Feel free to be creative with this and get students standing up and interacting – one structure I've found works well is to split the group in half, have one group take a first stab at each problem, and then come back together and discuss/work out bugs or inconsistencies. Plus, since the premise of the problem is paper airplanes… feel free to make paper airplanes if you feel like your section needs some week 8 stress relief ☺

If you get through problem 4 and feel like they could use more practice with a slightly bulkier class, check out the Employee problem to talk more about some of the same themes.

Interactors Problems (1, 2)

Problems 1 and 2 give some more practice with interactors, which they haven't really focused on in a section yet. Because they build off of similar principles as MouseEvents, they're less of a priority, but feel free to explore one of these problems with students (I don't imagine it would be possible to get through more than one while covering classes sufficiently unless students are totally on it). Both problems involve graphics in addition to interactors.