Introduction to Modeling

CS1316: Representing Structure and Behavior

Today’s story
- What’s the point of this course?
- What’s a model?
- What are data structures?
- Why Java?
- Details on the course
- Getting set up for the course

The Point of this Course
- Real computer-based media developers rarely work in terms of pixels and samples
  - Computer musicians deal in terms of notes, instruments, patches, and other structures.
  - Computer animators deal in terms of characters, movement, scenes, and other structures
- Bottom-line: They structure their media.

(Some) Motivating Questions for the Course
- How did the wildebeest charge over the ridge in *The Lion King*?
- What is a piece of music?
- Who will win the World Series in 2007?
- Should we cover our mouths when we cough?

The Wildebeests in *The Lion King*

The Villagers in *The Hunchback of Notre Dame*
The answer: Modeling and Simulation

- Lion King scene does not use traditional drawn cel animation.
- Instead Disney:
  - Modeled the structure of wildebeest.
  - Modeled the behavior of wildebeest (how they stampede).
  - Then started a computer simulation that executed the model...and "filmed" the output.

What’s “modeling”?

- Describing things in the world in terms of their structure and behavior.
  - \( F=ma \) (Force=mass * acceleration) is part of a model of the world that describes what happens when one thing hits another.
  - Maps model physical spaces and their physical relationships
- On a computer, we can execute these models: Make them work, plug values into equations, move things in space, see what happens.
  - That’s simulation: Executing a model

What’s a data structure?

- A way of organizing information.
- Different physical structures organize space differently.
  - Skyscrapers vs. ranch homes.
  - Trees vs. snail shells
- Data structures organize the information we use in our programs in different ways.

Data structures you know

<table>
<thead>
<tr>
<th>Age</th>
<th>Eye Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matt</td>
<td>Brown</td>
</tr>
<tr>
<td>Jenny</td>
<td>Blue</td>
</tr>
</tbody>
</table>

Data structures that you’ll come to know

- Arrays and tables keep things organized right next to one another.
  - Makes it easy to find something in the array or table
  - But if you want to insert something new, you have to move everything over.
- Linked lists and trees keep track of relationships with links (or edges)
  - Easier to insert new things

Data structures have different properties
Thought experiment:
Adding a second of silence into a sound

- Assuming that there’s room for another second in the sound…
- We copy samples from the insertion point to the end of sound down one second:
  ```javascript
  sampleValueAt(sound, soundIndex + oneSec, getSampleValueAt(sound, soundIndex))
  ```
- Then we can insert oneSec’s worth of 0’s into the insertion point

How that looks visually

Insrerting into a table

<table>
<thead>
<tr>
<th></th>
<th>Age</th>
<th>Eye Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Matt</td>
<td>Brown</td>
</tr>
<tr>
<td></td>
<td>Jenny</td>
<td>Blue</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Age</th>
<th>Eye Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Matt</td>
<td>Brown</td>
</tr>
<tr>
<td></td>
<td>Jenny</td>
<td>Blue</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Age</th>
<th>Eye Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Matt</td>
<td>Brown</td>
</tr>
<tr>
<td></td>
<td>Jenny</td>
<td>Blue</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Age</th>
<th>Eye Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Katie</td>
<td>Brown</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Age</th>
<th>Eye Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Matt</td>
<td>Brown</td>
</tr>
<tr>
<td></td>
<td>Jenny</td>
<td>Blue</td>
</tr>
</tbody>
</table>

Inserting into a linked list

Note in Kitchen

- Note in Living Room
- Note in Study
- Note in Bedroom

Inserting into a linked list

Note in Kitchen

- Note in Living Room
- Note in Study
- Note in Bedroom

Modeling and Simulations are about data structures

- The visual structure of wildebeest is a tree or graph.
- Tracking which does something next is a queue.
- All of the wildebeest to stampede are stored in a list.
- The images to be used in making wildebeest run are usually stored in a list.
Learning objectives in the course

Computer Science Learning Objectives
- Students will be able to program Java classes and methods based on modification.
- Students will learn how to use and manipulate several core data structures: Arrays, linked lists, trees, stacks, and queues.

Media Learning Objectives
- Students will be able to explain the role of data structures in structuring and manipulating data, especially multimedia.
- Students will be able to explain key issues of modern animations, such as sound synchronization and moving objects in layers.
- Students will be able to discuss the properties, strengths, and weaknesses of the different structuring approaches for media.
- Students will be able to design, define, and implement some simulations.
- Students will be able to explain the value of computation for modeling and simulation.

Why are we using Java? (Why aren’t we using Python?)
- Java is faster than Python
  - We can do more operations in less time, so we can do more complicated media in less time.
- Java is more well-known than Python.
  - So there’s more “resume value” than Python.
- If you take more CS, it’ll probably be in Java.
  - More CS classes are being taught now in Java than in other programming languages.

General flow of course

- Introduction to Java
  - Manipulation of pictures and sounds (as in CS1315)
  - Manipulating music and turtles
- Using arrays, linked lists, and trees
  - With music, pictures, and sounds
  - Creating animations using arrays, lists, and trees
  - Generalized linked lists and trees
- Creating simulations
  - Predator/prey, disease propagation models, movement of people
  - Different kinds of random
  - Sorting our events
  - Simulations with resources
  - Resource queues
  - Creating animations with simulations

Class website (Class CoWeb)

- http://coweb.cc.gatech.edu/cs1316
  - Links to other on-line materials you’ll need
  - Syllabus and all slides
  - Homework assignments
  - We’ll use Sakai (Tsquare) for turnin and grades

Reaching me (Colin Potts)

- potts@cc.gatech.edu
- TSRB 339 (85 Fifth Street)
- Office phone: (404) 894-5551
- AIM/Yahoo: colin2sheds
- Office hours: t.b.a.

Teaching Assistants

- Dawn Finney
  - Head T.A.
  - dawn.finney@gatech.edu
  - Office hours: t.b.a.
- Kristin Caldwell
  - Section A1
  - kristinc@gatech.edu
  - Recitation: Mondays
  - Office hours: t.b.a.
- Rory Murray
  - Section A2
  - LordOfFace@gatech.edu
  - Recitation: Wednesdays
  - Office hours: t.b.a.
Recitations
- Mon/Wed 6:00-7.45pm (CoC Bldg 101)
- WE WILL HAVE (shortened) RECITATIONS THIS WEEK!
  - It'll be help on installing DrJava and other files.
  - Always feel free to bring your laptop computer to recitation for help!

Course text
- Course notes
- Available in Engineer’s Bookstore and also on website
  - Reading the text is required.
    - You get to choose your medium.
    - BUT for the most part: Your text is the course slides.

Grading policy
- 20% for four in-class quizzes.
  - Pre-quiz on-line, not graded nor turned in.
  - Top three grades will count for you
- 30% for nine homeworks
- 50% for three exams
  - Two midterms worth 25% (12.5% each)
  - One final worth 25%
- No curve. 90, 80, 70, 60 cutoffs

Homework
- Collaborative, but you should do it on-your-own as much as you can.
  - (1) You’ll want to create your own media.
  - (2) You’ll learn the coding better on your own, so you’ll do better on exams.
  - You must acknowledge collaborators!
- First homework is due soon: Building a picture function.
- Last four homeworks are required to be pair-programming exercises
- No credit for late homeworks, due 7pm EDT

Homeworks for Summer 2007 (Tentative)
- HW1: Implement one new image method in Picture
- HW2: Create a Picture collage
- HW3: Create four-part music
- HW4: Use Weaving and Repeating to create music
- HW5: Create new picture list manipulations
- HW6: Create an animation with sound effects
  - HW 6.5 (extra credit) Create a walking character
- HW7: Build a GUI for changing a picture
- HW8: Simulate an ecosystem
- HW9: Simulate a stampeding crowd scene

What you need to do to get started
- Install Java SDK.
  - http://www.java.sun.com
  - Java 1.4.2 or 1.5 for Windows. (You should already have it for Macintosh.)
- Install DrJava
  - http://www.drjava.org
- Install JMusic
- Install Java code for class from website or CD
- (Probably want to download text and slides.)