

Class 01: Introduction

Held: Thursday, 24 January 2013

Summary: We begin with an overview of the expectations for the course and of how research is conducted in Glimmer labs.

Related Pages:

- EBoard.

Notes:

- I'm offering two sessions of this seminar. One meets on Wednesdays at 11 a.m. The other meets Thursdays at 1:15 pm. You are welcome to attend either. (If you can't make either, let me know, and I'll see if I can find another time.)
- I encourage you to attend this week's CS extra Thursday at 4:15, since it's on an architecture for Media Scripting. You'll see a more detailed version of what we're talking about today.
- I encourage you to attend Monday's Microsoft session.
- I encourage you to attend next Thursday's session on summer research opportunities in the CS department.
- I've been asked to remind students that academic honesty cases have been rising at the College and you should be careful to understand your faculty members' expectations.
- I've been asked to remind students who use MathLAN workstations to log out of the workstations.

Overview:

- About this course
- Media Scripting research in Glimmer labs
- The typical architecture of a Media Scripting application
- Other important practices
- Looking ahead

About this course

- As most of you know, this course exists because my summer research students suggested I provide prospective research students with the opportunity to learn some of the tools and techniques we use during summer research in the spring before they do the research.
- This is a one-credit course. That means that I expect about three hours of your time each week: One hour in class and two hours outside of class.
 - If you have time, you may find some benefit in spending more than two hours outside of class, but it's up to you.
- Although we will work in C, I will do my best to assume no more C than you'll know at the point we hit a topic. (Well, I may assume C that you won't get in 161, but then I'll teach it to you.) We won't

use C until

- This course is graded on an S/D/F scale. If you miss no more than two classes, attempt the homework, and are active in class, you will get an S. If you fail to meet this criteria, you will get an F.
- I will do my best to put outlines for each class and an EBoard for each class on the Web.
 - The rest of the Web site was created with copy-and-paste. I make no guarantees about accuracy or relevance.
- I am going to try to put my lectures in more narrative form. When I create those narratives, you can find them under the Glimmer Guide link at the top of each page.
- I don't plan to cover much in detail about C and general Unix tools. If you've taken CSC 161 and have the time, I would recommend you consider sitting in on my C/*nix seminar on Tuesdays at 1:15.

Media Scripting research in Glimmer labs

- My present research emphasizes intersections between computing and the arts.
- My work (and research questions) emphasize scripting of media applications, approaching scripting from three directions.
 - **Education:** *Can the ability to script media applications encourage people (undergraduates, middle-school students, artists and designers, etc.) to learn basic programming concepts?*
 - **Art:** *How might the ability to script media applications affect artistic practice?*
 - **CS:** *Can we successfully apply the pure functional model to scripting media applications?*
- I ask these questions, in part, by building applications that support various forms of Media Scripting.
- I tend to conduct my research with groups of student assistants. Most work in the summer, but a few work during the academic year.
- My work started with 151 but is reaching beyond it.

The typical architecture of a Media Scripting application

Scriptable applications form the center of my work. How do we build such applications? We've tried a variety of techniques over the years, and have settled on a technique that seems to be promising. (If all goes well, it will be used for the core Scheme/Gimp interface starting in Fall 2013.)

- We extend the media application (e.g., GIMP, Inkscape, Audacity) to support inter-application communication, using the D-Bus framework.
 - We need to choose which actions we support (e.g., "Select a brush")
 - We need to choose how we model actions with functions.
- We extend a programming language (e.g., Scheme, Python) to support the D-Bus framework.
 - We need to design usable and clear frameworks.
- We build a library of simplified wrapper functions to make it easier to interact with the application.
- We write exercises that help people learn.
- Increasingly, we look for ways to incorporate "self disclosure", so that the application can let the programmer know the code versions of things done by hand.

So, what do you need to do all of this?

- Going back down the chain from the language to the application
 - We need to know how to send function calls on D-Bus
 - We need to convert from the language's representation of data to the common representation of data.
 - We need to know how to accept function calls on D-Bus (at the application side).
 - We need to know how to convert data from general form to the application's representation.
- Going back up the chain
 - We need to know how to call functions within the application. That means we need to understand the architecture of the application.
 - We need to know how to publish information on function to D-Bus
 - We need to be able to translate the internal representation of data to a common representation of data.
 - We need to know the internals of the programming language to support it acting as a client. In particular, we need to translate from the common representation of data to the language's representation.

Other important practices

- Working with other folks. (Not discussed in this course.)
- Managing code that's shared with other folks. (Next week.)
- Building large projects with make and automake.
- Other good programming practices, including formatting conventions. (Implicit - Read the GNU guidelines.)

Looking ahead

See the schedule.

Copyright © 2012 Samuel A. Rebelsky. This work is licensed under a Creative Commons Attribution-NonCommercial 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.