

S603: Introduction to Flex 3

SLIS Workshop Proposal Summer I and II, 1.5 Credits

Instructor: Shashikant Penumathy

PhD Student, School of Library and Information Science,
Indiana University Bloomington

Rationale

The recent Web 2.0 phenomenon has ushered in a new era of responsive, interactive and engaging user interfaces on the Web. Web applications today must not only perform well, but also provide the user with a fluid, dynamic and generally outstanding experience. Until recently, such experiences could be crafted only by those who had a deep understanding of JavaScript, XHTML, CSS and most importantly, how different browsers would interpret the same code. In other words, a programmer would have to spend days or weeks making sure that the code he/she wrote would look and work in the same manner in Firefox, Internet Explorer, Safari, Opera and so on. It also frequently required knowledge of multiple technologies such as PHP, XHTML, JSP, etc or intimidating technologies such as .Net.

With Flex, Adobe solved many problems with Web application development. Today the Flex 3 framework provides an easy to use set of components that can be used to create highly interactive user interfaces that provide a consistent, high-quality user experience across different browsers. Using the Adobe's Flash platform, designers and programmers can rapidly prototype and develop interfaces from simple Web forms to interactive data visualizations.

Companies such as Oracle, Intuit, NVidia and Yahoo have made Adobe Flex a part of their technology strategy because it can lead to great increases in productivity. As a result, a search for 'Flex' on sites such as monster.com or theladders.com today returns thousands of results. For anyone who wants to work in the area of Rich Internet Applications, a working knowledge of Flex is important. For those who wish to make a career developing such applications, expert knowledge is essential.

In addition, the simplicity of Flex technology can allow students taking courses such as Web Programming (S571), Human Computer Interaction (S516) and Information Visualization (S637) to put their classroom knowledge to the test immediately. Playing with different kinds of UIs, visual representations and interaction strategies is easier with Flex than with heavyweight UI technologies such as Java Swing or quirky ones such as DHTML.

The intended audience for this workshop consists of undergraduate or graduate students who are interested in learning about Rich Internet Applications, interactive user interfaces, or just want a gentle introduction to software development. This workshop can be a great way to overcome the fear of programming that many students have. Students that will do well in this course are likely to have taken a technical, hands-on course or workshop (see prerequisites).

Course Description

This course provides an introduction to creating Rich Internet Applications (RIAs) using Adobe's Flex 3 framework. At the end of this course, students will be able to prototype simple, interactive user interfaces using Flex Builder 3 and get a sense of whether a career in user interface development would be fun for them.

Scheduling

Dates: Both Summer I and Summer II.

Place: SLIS Mac Lab LI002

Credits: 1.5

This material is best taught as a workshop because it is a short introduction to a specific software technology and software development in general. A longer course would demand a more in-depth treatment of the material, which can be very challenging for people exposed to programming for the first-time. This course will provide a very gentle introduction to creating user interfaces with a little bit of programming at the end of the course that will help the student decide if a career in software development is interesting to him or her.

Prerequisites

A strong desire to learn combined with a resilient attitude is necessary to learn any technology. Other pre-requisites:

- S401 or equivalent skills.
- Sufficient proficiency with computers to be able to download, install and run the trial version of Adobe Flex Builder 3 from <http://www.adobe.com/products/flex/>
- **One** of the following:
 - Having taken a technical SLIS course such as Web Programming (S517) or Information Retrieval Theory (S534).
 - Consent of instructor. If the student has not taken **any** technical course, he/she may find this workshop a bit challenging. Interested students are encouraged to try the test (in the flyer at the end) and talk to the instructor in person.
- A basic understanding of XML or XHTML. Knowledge of XSLT or XLink is **not** required. Students that have taken the S603 XML workshop will find MXML very easy to understand.

Syllabus

Objectives

At the completion of this course, a student will:

- Understand how to use the Flex builder tool to quickly prototype a user interface.
- Know how to create moderately complex user interfaces without programming.
- Get a sense of whether a career in user interface design and development would be fun for him/her.
- **Note:** This course does **not** involve serious programming with ActionScript. It uses MXML, which is like HTML, but easier. Towards the end of the course the student will have the chance to try out some simple programming to see if he/she likes it.

Grading

The student's final course grade will be computed on the basis of points assigned for class participation, exercises and a project. The final grade will be computed according to the following distribution:

Class Participation	10 points
Assignments	60 points
Final Project	30 points
Total	100 points

Grades: A: 95+, A-: 90-95, B+: 85-90, B: 80-85, B-: Below 80. Hard work and dedication will be rewarded with an increase in grade (so A- + high level of dedication = A). SLIS letter grade definitions follow.

A	4.0	Outstanding achievement. Student performance demonstrates full command of the course materials and evinces a high level of originality and/or creativity that far surpasses course expectations.
A-	3.7	Excellent achievement. Student performance demonstrates thorough knowledge of the course materials and exceeds course expectations by completing all requirements in a superior manner.
B+	3.3	Very good work. Student performance demonstrates above-average comprehension of the course materials and exceeds course expectations on all tasks as defined in the course syllabus.
B	3.0	Student performance meets designated course expectations and demonstrates understanding of the course materials at an acceptable level.
B-	2.7	Marginal work. Student performance demonstrates incomplete understanding of course materials.
C+	2.3	Unsatisfactory work. Student performance demonstrates incomplete and inadequate understanding of course materials.
C	2.0	
C-	1.7	Unacceptable work. Coursework performed at this level will not count
D+	1.3	toward the MLS or MIS degree. For the course to count toward the
D	1.0	degree, the student must repeat the course with a passing grade.
D-	0.7	
F	0.0	Failing. Student may continue in program only with permission of the Dean.

Assignments, Projects & Late Submissions

In this class students will get a second chance to finish their assignments because sometimes it is easy to get stuck with a problem. Here are the rules:

- Assignments will be handed out on Tuesdays and will be due by 5 PM next Monday.
- Late submissions will be accepted only if the student informs the instructor within 2 days of the assignment being handed out that he/she will not meet the deadline. A student that does so will be given 2 extra days to finish the assignment, i.e. he/she can turn it in the next Wednesday instead of Monday. A student may use this buffer only once during the entire workshop and may not use this for finishing the final project.
- A student that turns in an assignment late without informing the instructor will receive no points.
- If a student faces extraordinary circumstances, he/she should talk to the instructor and negotiate an alternative. The objective is to learn, not just get a grade, so the student is expected to be as honest and direct as possible regarding any issues he/she may be facing and try his/her best to finish the assignments.

Incompletes

This is a short hands-on workshop so a grade of incomplete ("I") is not appropriate.

Academic dishonesty

Policies on academic dishonesty have been established by Indiana University and the School of Library and Information Science. These policies, which have been set out in Indiana University's Code of Student Rights Responsibilities and Student Conduct will be adhered to in this class. The Code is available at <http://campuslife.indiana.edu/Code/index1.html> [See Section III.A Student Misconduct--Academic Misconduct which is available online at <http://campuslife.indiana.edu/Code/toc.html#sec3>].

An assignment or project that indicates any form of academic dishonesty will receive no points.

Special Needs

Students with special needs should discuss them with the instructor prior to beginning this workshop.

Texts & References

Software technology moves rapidly and texts can become obsolete. The following books are recommended **only** if the student is serious about using Flex beyond this course and has programming experience:

- Flex 3 Cookbook: Code-recipes, Tips and Tricks for RIA Developers by Joshua Noble and Todd Anderson.
- Essential ActionScript 3.0 by Colin Moock.

This course will use parts of the following reference:

Flex 3 Developer's Guide: <http://livedocs.adobe.com/flex/3/html/help.html>

If a student uses any resources beyond these two such as online forums, friends, classmates, etc. he/she **must** acknowledge this in their assignments. Working in groups is encouraged.

Resources

This workshop needs Adobe's Flex Builder 3 integrated development environment (IDE) to be installed in the SLIS mac lab. The trial version of this tool lasts 60 days. Students, faculty and staff can obtain the professional version for free by sending Adobe proof that the tool will be used in an academic setting.

The license permits installation of the IDE on multiple machines in a lab.

See <https://freeriatools.adobe.com/flex/>

Office Hours

By appointment. (Regular office hours can be allocated based on student preferences.)

Tentative Schedule Assuming Class Meets on Tuesdays

Date	Topic & Readings	Assignments
May 12, 2009	Introduction to Flex 3 Watch Animated Overview Explore Flex Showcase Read Technology comparisons , Getting started with Flex Builder 3 , Flex Builder basics , About MXML , Code editing in Flex Builder	Assignment 1 10 points
May 18, 2009@5PM	Assignment 1 Due	
May 19, 2009	Layout, UI Components and Images Read: Introducing containers (excluding Flex Coordinates and Managing Components at Run Time), Sizing and positioning components , Using layout containers , Controls	Assignment 2 15 points
May 25, 2009@5PM	Assignment 2 Due	
May 26, 2009	Data Providers Read: About collections and Data provider components	Assignment 3 15 points
Jun 1, 2009@5PM	Assignment 3 Due	
Jun 2, 2009	View States and Transitions Read: Using View States , About transitions , Defining transitions	Assignment 4 20 points
Jun 8, 2009@5PM	Assignment 4 Due	
Jun 9, 2009	A Peek into Events and Functions Read: Using Events , Programming fundamentals , Common program elements , Working with objects	Project 30 points
Jun 11, 2009@5PM	Project Design Due	
Jun 15, 2009@5PM	Project Checkpoint	
Jun 16, 2009@5PM	Work on Projects in Lab (optional)	
Jun 18, 2009@5PM	Project Due	

Assignments

If a question requires a descriptive answer, write it as a comment within your MXML file. Assignments will be turned in as an archive of your entire “project” file. To find interesting images, use the creative commons search engine at <http://search.creativecommons.org/>. To generate random text, use <http://www.lipsum.com>.

Assignment 1

1. Create a new Flex project called ‘<your username>_assignment1’. It should contain one MXML application file called ‘HelloFlex.mxml’.
2. Set the background color of your application to white. If you run the application, how does the background look?
3. What do you think is the difference between the backgroundColor and backgroundGradientColors properties? If you set both properties, what happens?
4. Using Panel, Form, ControlBar and any other components you want, create a course registration form with at least 10 elements, a title, a submit and a cancel button. The form won’t actually do anything.
5. Change colors, fonts, backgrounds, spacing, alignment etc. to make the form look as hideous and revolting as you can (you read that right). Results will be shared with the entire class so go nuts!
6. The main challenge here will be to work in a methodical manner. Its easy to make too many changes at once such that you lose track of what you did and the application fails to run! Develop the habit of making small changes and repeatedly running the application to see if it works.

Assignment 2

1. Pick a news website of your choice (nytimes.com, idsnews.com).
2. Take a screenshot of part of the page (at least 1280x1024). It must contain the banner.
3. Recreate it in Flex as closely as possible without using absolute layout, i.e. you must use boxes, grids, etc.
4. You must use at least: 2 images, 2 buttons, 2 text inputs, 5 text components and one navigator component (TabBar, Accordion, etc). You’ll get full points if you meet these requirements. Do not use images to simulate the appearance of text.
5. Don’t worry about getting it perfect in terms of fonts, colors and behaviors like rollover effects. Pay attention to positioning and sizing. The objective is to gain an understanding of how layout works, an appreciation for how hard it is to do right and to have fun!
6. Include the screenshot in your project archive when you turn in your assignment.

Assignment 3

1. Use the form you created in assignment one and modify it so that it includes at least 2 combo boxes, 2 lists and 1 data grid with at least 3 columns. Modify the form so that it uses styles to control appearance.
2. Populate the above components using data providers with some random data of your choice.
3. Coming up with random data can be surprisingly hard so plan for it and give yourself enough time.

Assignment 4

1. Your application should contain 3 states. The first state should show a login form with a submit button. When the button is clicked, your application should transition (using an effect of your choice) to the second state.
2. The second state should contain the UI from the assignment 3. When the submit button is clicked, your application should transition (using an effect of your choice) to the third state.
3. The third state should contain the UI from Assignment 2.
4. Once you are in the third state, what's an easy way to go back to the first state? What do you think is happening? Hint: Use a browser to *refresh* your memory.
5. The challenge here will be dealing with complexity. You should split up the UI for the three states into three (or more) files so that your application is manageable.

Project

On the first day of your new job, your boss asks you to create a UI mockup for a cool new Web application. Your boss knows you don't have programming experience but wants the mockup to respond to mouse clicks so that it feels real to some extent.

Requirements

1. At least two view states with one transition between them.
2. At least two components that use data providers. One of the data providers should react to events. For example, if you use a set of buttons and a data grid, clicking each button might highlight or select a different element in the data grid. The objective is to demonstrate that you know how to do this so keep it simple.
3. The use of at least two images.
4. Use a color scheme of your choice (try <http://kuler.adobe.com>) and apply it to a part of your application.

Process

1. Create a sketch showing what your application looks like (both states) on paper. The sketch should show what components you are going to use, how they are laid out and provide an idea of which elements are going to react to events and how. You can use the applications on the Flex showcase for design inspiration but don't spend more than a couple of hours designing. As you know by now, even the simplest designs can pose serious challenges so keep it very simple.
2. Annotate your sketch with names of specific containers and components you will use to implement it (HBox, TabBar, Button etc). You should be able to translate your design into MXML with very little effort.
3. Discuss your design with the instructor to make sure you can finish the project on time, especially if you intend to write ActionScript code.
4. **Project Checkpoint:** This is an insurance policy for you. On this date you will turn in an archive of whatever state your project is in that day. If your final project works on the final due date, the

checkpoint won't be used. If your final version does not work, you will get a chance to explain what you tried and use the checkpoint as proof. If you tried hard, you won't lose many points. If you didn't do anything by the checkpoint and your project does not work (meaning you rushed at the last minute), you will not be graded leniently.

5. Turn in your project archive electronically by email. You can turn in your sketch as paper, photocopy, digital file, scanned image, etc. You can also choose to present your final project to the instructor in person (in addition to emailing the archive). If you are not fully confident of your work, this latter option is recommended because you can get a chance to explain what you tried and get more points!

Sample Flyer

SLIS WORKSHOP S603: INTRODUCTION TO FLEX 3

MARCH 9, 2009

<http://ella.slis.indiana.edu/~sprao/flex3>

Flex 3 Workshop

Create interactive user interfaces using Adobe's cool new framework

Why are hundreds of companies such as Oracle, Intuit and Yahoo looking for people who know Flex? Because Flex makes creating rich user interfaces (UIs) a snap. In this workshop, you will learn how to rapidly prototype rich UIs using Adobe Flex Builder 3.

Maybe you always liked cool UIs but heavyweight technologies such as Java or quirky ones like JavaScript +XHTML+CSS put you off. Or maybe you always wanted a more gentle segway into software technology than what the CS* dept could offer. Yes? Then this course may be for you.



Anyone can ^{code} cook
Programming experience not necessary

If you find the snippets on the right interesting, you will do just fine in this workshop. If you get them right, you will do *very* well.

Who's talking?

Shashikant Penumarthy is a PhD student in SLIS who creates advanced visual interfaces for a living. He is tired of geeks scaring people away from programming and is hell-bent on fixing it.

S603 Summer I & II, 1.5 credits
School of Library & Information Science
<http://ella.slis.indiana.edu/~sprao/flex3>

Try these snippets

1. In XML, every opening <tag> should have a corresponding close </tag> and tags should be nested such that elements that are opened last are closed first (e.g.: a b c /c /b /a). Can you fix the snippet below?

```
<person>
<address>
<street>10th</street>
<city>bloomington</city>
<state>in</address>
</person>
```

2. If a starts out with value 3 and b starts out with value 4, what does the following snippet say?

```
x = a + 1
y = a - 1
z = x times y
if z > 10 say "I think I'll do fine"
otherwise say "Maybe I should talk to the instructor"
```

3. How many inches wide are box1 and box2? (Box2 is inside Box1)

```
<box width="100 inches">
<box1 width="80%">
<box2 width="20%" />
</box1 >
</box>
```

4. A jar contains 30 Red, 30 Green and 30 Blue balls. With your eyes closed, how many balls do you have to pick from the jar so you can be sure two of them are exactly the same color?

Answers

2. $x = 4$; $y = 3$; $z = 3 * 4 = 12$
3. box1: $300 * 0.8 = 80$ inches, box2 = 20% of box1 = $80 * 0.2 = 16$ inches.
4. 4 balls (there are only 3 colors)

* Instructor is a former CS geek that has recently seen the light.