12 Final Project

Project goals and logistics

The goal of this assignment is to give you the opportunity to use all the concepts we have learned in the design and implementation of a small project. The project will involve user interactions, but the core is the design of the functionality, following the design recipes, building abstractions, and using the existing libraries. Good design and a simple user interaction part is much more valuable than a flashy user interactions with poor design of the program that drives it.

You should finish the design and implementation of your project by Tuesday, April 20th. The final version is due on Wednesday, April 21st.

The project presentation will be done in the lab in 212 WVH during the regular lab times on Tuesday, April 20th, and Wednesday, April 20th during the regular lecture times.

Project details

Choose one of the four options for the project. They come from four different area of computing application and expertise. Choose the option you feel most competent to tackle, and that interests you the most.

Here is a brief overview of the four options:

1. Graph Traversals - Finding a Path.

   The background work for this project has been already included in several earlier homeworks. The goal is to help the user plan the route for traveling between the capitals of the 48 contiguous US states. The main work is in implementing the graph algorithms presented in Monday's lecture, and in adding a suitable user interface and display of the results.

2. Mars images.

   The provided library allows you to read data that comprise images of the planet Mars as seen by the Viking Explorer in the early 1990's.

   Your job its to process the information contained in these bytes of image data to enhance the image contrast, to look for hidden messages, and to generate a better visualization of the given image.

This project illustrates the kind of algorithms used in image processing.

3. Musical frogger.

   The goal of this project is to explore the possibilities provided by the new *isdraw* library that allows you to add musical accompaniment to your interactive games.

   Choose this project if you have some understanding of music and feel that you would like to engage your musical creativity. The project web site includes a couple of samples of the kind of musical effect one can create.

   Doing this project will give you a start on understanding how you can work with MIDI synthesizers from a Java program.

4. Java universe.

   The latest version of the DrScheme *world* teachpack allows the programmer to interact with the *universe* teachpack. The *universe* teachpack provides a high level abstractions for designing a server program and several client programs that communicate with the server over the network.

   Chris Souvey and Griffin Schneider designed a library similar to *idraw* that includes the *universe* functionality.

   If you choose this project, your goal will be to design an interactive game where two or more players communicate with the server that mediates their communications. There are several resources available for you to learn how to design this kind of programs.

For each of these projects we give you fairly detailed specifications for the model part of the program. The requirements for the view are minimal. They only describe the basic functionality — you are free to enhance your presentation as you wish — and will be given credit for the work.

One part of the credit for this assignment will be given for a design document that describes the data, the organization of the program, the key program components, and the design of tests. Imagine you want someone to keep improving your program — provide a road map that explains what your program does and how does it do it. This document should complement the Javadoc generated web pages. A separate document will give you a more detailed guidelines for what we expect.

One part of the credit for this assignment is for the model part.

You will also get credit is for the user interactions (view) — grading both the design of the user views and the design of the program that drives it. A small bonus may be earned for exceptionally well designed display or interactions. It is better if the user interaction is done only through the console, but is well designed and documented, than if a fancy GUI display is driven by a code that another programmer cannot understand and maintain.

12.1 Project Presentation

You will present your project (both partners together) during the lab on Tuesday, April 20th and the lecture on Wednesday, April 21st. Each partner should be able to describe any part of the code in the project, regardless of *who wrote it*, as we expect that both partners work on the project together. More information about the presentations will be provided shortly.

12.2 The Advice

The design part of each project typically takes the greatest amount of time. the more time you spend thinking things through, the easier it is to actually write the code.

Make sure you think the whole framework through before you start programming. Spend some time researching the Java libraries to see what tasks can be done using the existing tools. Write sample adapters to see how the existing class can be used in your setting.

Then design the key component by specifying their interfaces — the method headers, the interfaces that various classes must implement or use to get information from others.

For now, you have not learned about various tools and techniques to support such design process — other than class diagrams. Any description that you find helpful in clarifying the roles of the different classes and interfaces in your program is acceptable.

The design document you produce should include a brief user's guide, give a general overview of the project organization as well as describe all data definitions and the key methods. The Javadocs supplement this with detailed information about the actual implementation.