

Final Project

Due Date: Wednesday, 20 April 2011 at 10:00 pm

14 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 most of the implementation of your project by the morning of Tuesday, April 19th. The project presentation will be done during the regular lab times on Tuesday, April 19th (from 9:00 am till 4:10 pm) and during the regular lecture times on Wednesday, April 20th (i.e. 9:15 am - 10:20 am and 1:35 pm - 2:40 pm). All presentations will be done in the lab in 212 WVH.

You can then use the rest of the time on Tuesday and Wednesday to finish the project, incorporate the suggestions given during the project review, and submit the final version by 10:00 pm on Wednesday, April 20th.

Project details

Choose one of the three options for the project. Choose the option you feel most competent to tackle, and that interests you the most.

Here is a brief overview of the three options:

1. Interactive game with sound effects.

Assignment: Design an interactive graphics-based game with sound effects and/or background music.

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

The focus is on the design of the game and on making sure the background music and the special effects integrate with the game actions. You do not need to have a serious musical background to tackle this project.

There are two variants of libraries for designing *worlds* that use the same *tunes* library. One is the `JavaWorld.jar` we have used in the course already. The other one is *isdraw* with extensive documentation on the *java-lib* web pages:

<http://www.ccs.neu.edu/java-lib/SoundLib>

The project web site for the *SoundLib* library includes a couple of samples of the kind of musical effects one can create and provides explanation of how the sounds are added to the game behavior. All but one sample include the source code for that project.

The drawing methods in the *isdraw* are different from *JavaWorld*, but the way the sound is handled is the same for both.

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

2. Musical explorations.

Assignment: Design a program that will allow the user to experiment with music, either learning about the notes, chords, instruments, musical structures, or musical instruments; or will allow the user to interactively compose music from several components or musical tunes.

Choose this project if you have some understanding of music and feel that you would like to engage your musical creativity.

The goal of this project is to give you a chance to explore how a computer program can be used in music composition, or in learning to understand music notation, musical instruments, etc.

3. Algorithm visualization and animation.

Assignment: Design a program that will allow the user to visualize and explore the behavior of an algorithm. Include sound effects, if possible.

The goal of this project is to provide the user with the opportunity to explore the behavior of the selected algorithm, see the effect of every step of the algorithm.

Choose this project if you think you would like to help the user to understand the details of how the algorithm you select behaves, and are not really interested in exploring musical effects in depth.

Your program should demonstrate visually steps in one of the algorithms we have studied (or you may choose a new one that you have learned about somewhere else). It should let the user control the pace, or step through using some keyboard input.

The project grade will cover the following:

- The design document
- The design of the program model
- The user's guide
- The design of the user's interactions
- The project presentation

Of course, the design of appropriate tests is an integral part of the design of any program (*just in case you have not yet realized this*).

The design document

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.

The design of the program model

One part of the credit for this assignment is for the model part: the design of the behavior of the program apart from the user interfaces.

The user's guide

The user of your program needs to know the rules of the game, how to run the algorithm animation, or what are the features of your musical exploration program. You need to create a user's guide that will explain concisely what to expect of your program and how to interact with it.

The design of the user's interactions

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 simple, but is well designed and documented, than if a fancy GUI display is driven by a code that another programmer cannot understand and maintain.

Project presentation

You will present your project (both partners together) during the lecture on Thursday. 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.

14.1 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.