05433-B/05633-B User Interface Lab (Section B - GUI)

Final Project: BYOP

Proposal due 11:59pm Thursday, October 30th, 2014.
Rough Demo due 11:59pm Tuesday, November 11th 2014.
Presentation in class on Tuesday November 25th or Tuesday December 2nd, 2014.
Final Code due 11:59pm Thursday December 4th, 2014

Overview

For the Final Project, you will be implementing a project of your choice. You may select any of the project alternatives described below. You must let me know through email which alternative you choose by 11:59 pm, October 30th. If you choose alternative #4 (create your own), you must also describe with good detail what you plan on creating. In class on November 25 and December 2 we will be doing a product fair of final project demos for the rest of class. **Your grade for project 4 will be based partly on this presentation and live demo** of your project's features. Projects are intended to be individual, but we are open to proposals of large projects undertaken by two people. The final deliverable including the code of your complete project and your presentation materials, should be submitted via email by 11:59pm, December 4.

Project Alternatives

Alternative #1: A To-Do list (Base Score: 80)

Create a to-do list application. Your application should include these basic features:

- Allow users to enter new tasks
- Allow users to "check off" a particular task to signify that it's done. The task should become dim or have strikethrough, etc.
- Allow users to delete a task
- Allow users to edit a task

There are many free to-do list applications online that you can look for inspirations in UI design (just to name a few, Wunderlist, Remember the Milk, TeuxDeux, Google Tasks, etc.)

Ideas for more points:

- Group similar tasks using colors, folders or tags, etc.
- Drag and drop tasks into a trashcan
- Save/restore functionality – being able to save the tasks and open them again (involves learning how to save to and read from file).

Alternative #2: Photo browser (Base score: 80)

This project alternative is inspired by applications like Picasa, iPhoto and websites like Flicker. Your application should have at least two views: a "grid" view that lays all the photos out in a grid, and a "navigation" view that shows a particular photo and allows users to switch photos using next/previous controls. Users should be able to change the size of photos in the grid view.

Ideas for more points:

- Add animations to your user interface
- Add more views to the photo browser
- Allow users to zoom in and out of photos
- Allow users to group photos into folders

Alternative #3: Connect Four (Base Score: 80)

This project involves creating the familiar classic game: Connect Four. If you're not familiar with the rules, you can play it here: http://www.gamesgames.com/game/Connect-Four.html

Because implementing a computer player is outside of the scope of this class, you may implement a game for two human players. Your project should keep track of game variables like whose turn it is, whether the game is over (i.e. someone made connect-4 and won), and should allow users to start a new game after the game is over.

Ideas for more points:

- Add sound effects for chips falling and some one winning
- Add animation (the pieces fall down smoothly)
- More visual flare for the board and pieces

Alternative #4: Create your own

You may create your own final project as long as it is interactive, of sufficient complexity, and uses some of the tools you learned in this lab. Here are some ideas:

- Other board games (chess, go, etc.)
- A timer /reminder system
- An interactive real time game (breakout, tetris, your own game creation)

Product Fair

At the class wide project fair you should be able to:

- Explain your design process
  - Show early design steps such as initial sketches, storyboarding, pseudocode, FSM diagrams, etc.
- A live demonstration of your system that clearly shows off its features
- Give a high level overview of how you structured your code. The code structuring should show good software engineering principles such as:
  - Clear commenting
  - Modularization – using classes, components, etc to make the code more organized and modular
  - Use of MVC paradigm

- The exercise here is that you don't just jump directly to coding without proper design and logic planning first.

Remember that you will be graded partly on your project presentations, so make sure that your presentation includes every feature you implemented. Be sure to practice your presentation and make sure it's clear.

Grading

In the first three project alternatives, the "base score" represents the minimum score if all of the features mentioned are correctly implemented. Points will be deducted for missing features and bugs. Points will be added for adding cool features, polish, and enhancing usability. As mentioned, your grade for this project will be partly based on your presentation, so be sure to include every feature you implemented.