CIS 410/510: User Interface Programming

Winter 2007
Final Project

Due March 20, 2007 by 5pm in the CIS Office in Deschutes Hall.

Description: The goal of this assignment is to produce a *prototype* of software that has excellent usability. This is worth 30% of your overall grade. This could be team project with 2-3 people on a team. If you choose to work as a team, I will expect a more challenging application. You can produce the software using any GUI builder or programming language you want…..as long as I can run it to grade it. If you choose to do a Web project, please be aware that it must have substantial usability challenges.

Ideas for Final Projects:

- UI development tool (such as an AI system to design UI layouts or visual widget editor)
- Novel widget (such as a visualization tool or tiled windows)
- Software with embedded tutoring
  - Tangrams game with an embedded tutor
- Groupware application that is distributed and synchronous
  - Group editing or drawing system
  - Distributed, multi-player Tangrams game

Grading:
Usability of Final Artifact (10%)
Programming: Effort and quality of final code (70%)
Written Report: Completeness and quality of (20%)
NOTE: You will be graded on (1) completing all the parts of the assignment, (2) correctly applying the methods and techniques, (3) having the content make sense and be representative of the real world, and (4) the quality of your presentation and writing—communicating ideas clearly, concisely, completely, and correctly (spelling and grammar).

Hand-in:

1. *How to run the system* and any other needed instructions.

2. *Interactive System Design Report* (See below.)

3. *CD-ROM* containing the project. This should contain *source code* and *application* version ready to run.

4. *Paper version of source code* - well commented. IMPORTANT: Please highlight the code you have added to any GUI builder generated code.

5. *Group Member Evaluation (GME)* form rating each of your teammates and documenting any major problems you had with your group. (You should send this to me by email or put it in my CIS mailbox.)

The Interactive System Design Report

This document must COMMUNICATE your design. Keep that goal in mind. Another programmer should be able to imagine an interactive system and how it will work, or possibly implement from your description.

Use the following headings and contents for your report. This font is an explanation.

PART I: INTRODUCTION

A. Introduction

What kind of interactive program is it? Multi-player game, programming tool, groupware graphics editor?

B. Problem statement:

What is the program's purpose in general?

C. Context

1. Description of target hardware/software.
2. Integration: Other application software
   How does this application integrate into other software available on the computer? Have you accommodated that in the design.
3. Products
   What products will it produce and at what level of quality?
4. Outstanding constraints on design (standards, laws, etc.)

D. Description of program's functionality

What are the typical user tasks or activities? Describe briefly as a scenario.

E. Description of potential users

Who are the users? What is their experience with other similar devices? What are their skills? How often will they use this system, and thus will they become expert users? Is there anything special about them that affects usability? BE VERY SPECIFIC. This is where universal usability begins!!!!!

F. Brief description of the user studies you conducted, if any:

What kind of study did you do? (Task Analysis, Interviews, Observation, Questionnaire, etc.) Include example forms.

PART II. DETAILED USER INTERFACE DESCRIPTION

In this part of the document, you describe how the functional requirements are achieved through specific tasks/activities of the user with the user interface. This is the "blueprint" for the implemented user interface design: What functions the user can perform are tied to what the screens look like and what user actions occur.

A. Overall interface style chosen and why

B. Conceptual model or metaphor (if any) used

C. Description of the UI

Describe menus and other important elements of the GUI.

D. How the user interface works: Core activities

Provide at least three interaction scenarios or storyboards as narrative descriptions of the core user activities. These should illustrate typical use. In these narratives, coordinate screen states to represent what the user sees. Responsive actions are described in the narrative. This will provide a description of how the user interface works for core activities without having to

run the program. These also can be used for training purposes in the user manual. Be sure that labels and screens are large enough that we can read the text!

E. How the user interface works: Handling trouble

Describe at least one scenario of a problem encountered doing core activities. These should illustrate error dialogs, undo and using on-line help.

PART III. PROGRAMMING and TESTING

A. Programming

Reflect on conceptual elements that we have learned in class.

1. Programming language chosen and how well it allowed you to implement your ideas

   What did you have problems with? What was easy?

2. Widget sets, event management and geometry managers

   What did you have problems with? What was easy?

3. Programming environments (UIDE) for developing GUIs

   What did you have problems with? What was easy? How would you change it?

B. Testing

1. Program Testing and Debugging

   Describe the program testing that you did. Was it adequate for discovering UI bugs? Explain. How easily did your language choice lend itself to developing prototypes?

2. Usability Testing

   Did you attempt any usability testing with real users? What were the results? How easily did your programming language choice allow you to run usability tests on very early designs?

PART IV. CONCLUSIONS

A. Summary

Briefly describe what you accomplished with your development effort and what you didn't. Is the program working? What problems were you able to fix? What problems still need to be fixed? Explain why.

B. Future Design

What improvements, extensions, etc. does it need? Where should this design go in the future?