CISC106 Summer 2009 Project 2

July 27

Team rosters due. Project focus must be chosen. Team must submit by email (see email instructions below) to professor by 9 p.m.

Aug 3

Individual progress report due at midnight.

Aug 12

Complete Project 2 due midnight.

Project Overview

You have several options for Project 2. Please decide within your team which option you would like to pursue, put together a plan for how you will complete the project, and submit this to the professor by July 27. You may choose to do your own take on the example, but no more than one group may do exactly the same project.

Project Options

- 1. Database (Matlab or C)
 - Your group must create a database of 5-10 related structures, save and load the database from a file, and provide functions to insert new data items, modify current data items, and generate reports based on the data.
 - As an example, a zoo database will have many types of animals. It will also have zookeepers and visitors. The database would keep track of which zookeepers tended to which animals, and it would be able to change zookeeper assignments. It could also keep track of which animals are the most visited and generate a report that shows this.

2. Game

- An interactive graphical game. The game should take user input through mouse clicks or key presses. The game can be real-time or turn based.
- Example: simple minesweeper: a random board of hidden squares is displayed. Player can use the arrow keys to select a square and press the space bar to reveal what is underneath. Some squares are mines and the game ends. Other squares display the number of mines next to the current square (you would not need to implement the auto-expand on zero mines like it does in Windows). The game detects if all the non-mine squares are revealed.

3. Animation

• An animated movie. Using graphics (and sound if warranted), program an animation (which is just a series of individual frames drawn on the screen).

- Example: Fractal generation. Draws a specific fractal function at depth 1 in one color, and then deepens the image by drawing the same function at depth 2 in a different color. Iterates until the function is too small to display on the screen.
- 4. Text Editor
 - A simple text editor. Using gui components, lay out a simple File and Edit menu and a large text box to edit a document. The user should be able to open, edit, and save a text document. The editor should support cut, copy, and paste operations from the Edit menu.

5. Graphics Editing Functions (Matlab or C)

- A set of graphics editing functions. The graphics editing should load, edit, and save a graphics file that is displayable in a web browser. This can be done through command line arguments (C) or function calls (Matlab). A project should implement 2-3 different graphics manipulations such as resizing, recoloring, blurring, or distorting the image.
- Example: resizeImage takes three arguments, current image file and new width/height. The function then reads the image data from the file, calculates how to shrink or grow the image to the given width/height and saves a new image file.

Project Components

Different project options have individual requirements, however, all projects must follow good programming practice. All functions must have documentation (comments) indicating the types of the parameters and the purpose of the function. Functions should not exceed 20 lines of code; break larger functions into smaller helper/subfunctions. All functions should include examples in the comments of an appropriate call to the function, and if the function returns an output value these examples should be included in a test script with checks for the correct expected output. Functions and test scripts should have labels for which group member(s) contributed the test or function.

Your submitted project will include:

- 1. All code (.m files)
- 2. Test script (.m file) that tests your functions individually
- 3. Test script (.m file) that executes your overall project
- 4. Diary file showing a sample run of your overall project
- 5. Any additional documentation (such as user interface commands) needed to use your project

Grading

You will be part of a coding team (no more than 3 people per team). You will be individually responsible for certain specific parts of the project, and one third of your grade will be based on your achievement of those parts. One third will be a grade for the achievements of the team. The team grade will be apportioned among team members by the team, using a blind rating system where each team member rates the contributions of their peers. The final third will be based on project questions given in a quiz or exam.