CSE165/ENGR140: Final Project

Your final project will be a group project. Each group is supposed to have 3 members and there will be a presentation of the project on **Tuesday May 10** during the scheduled final exam time. The presentation will be very short (about 6min) and you will simply present the results of your project, with a demo running on a laptop computer, slides, a video, or any other media as applicable to your project. Exceptions for a lower number of members per group are possible provided there is a given reason for it. You may bring your executable in a USB drive and use my computer. Be sure to schedule a time with me to test your program with my computer before your presentation.

Each group should choose one topic among the topics given below. A different topic (with the use of the freeglut support code as a platform) is possible and can be proposed by any group, but a detailed proposal is required to describe the intended project. We will then let you know if your proposed topic is fine or if it needs adjustments etc.

What to submit

In addition to the final presentation you are also required to submit the following materials:

1. Report – written group report of your project, at least 2 pages long (PDF format).
2. Source code – the source code you developed for the project (archived directory).

Create a single archived file (zip,7z) that contains all of the above. Add it to CatCourses under the Final Project assignment entry. Only one member of the group needs to submit the material.

Content of presentation and report

The format of the presentation and of the final report must discuss the following parts:

1. Project description – a short description of the project topic.
2. Members – description of how each member contributed to the project.
3. Implementation – short description of your source code.
4. Results – what are the results you obtained from your implementation?
5. Lessons/Conclusions – what are the lessons you learned and you want to share with your colleagues.

Important Dates:

- **Thursday Mar 31 11:59PM** – submission of groups and topics. We will have one exercise entry in **LA8** where each group member will need to submit his/her project information (list of members and topic), and as well a short description of what will be his/her part in the project.
- **Monday May 9 11:59PM** – deadline for uploading to CatCourses your project materials.
- **Tuesday May 10 3:00PM** – project presentations (Official final exam schedule).

Grading:

50% - minimum requirements of the specific project

20% - overall quality of the results

20% - quality of the final presentation

10% - quality of report

(Grades may be different per member if no sufficient contribution is seen for a given group member.)

Project topics:

1. Video Game

Use the freeglut support code as a platform to develop a video game. We do not expect you to learn/explore advanced graphics concepts or additional freeglut functionality; you will be doing something very interesting with the 2D primitives in your mini-projects from the labs. Here are some examples of games you can implement: space invaders, pac man, brick out, tetris, etc. As a group project, one member will have to be the "integrator" who will implement the main game logic, and the other members can be divided to work on many other tasks: to write classes for drawing the various elements of the game, for drawing the environment of several levels, designing classes that will compute the behavior of the needed game entities, etc. Even the simplest 2D game can be as complex as you'd like in terms of features and art.

Minimum Requirements:

a) The game has to have at least one object controllable by the user (multiple players ok).

b) At least one object will move autonomously according to the game logic.

c) The user must be able to interact with the autonomous object(s) in order to play the game.

2. Algorithm Animation

Use the freeglut support code as a platform to animate algorithms. You can select any suitable algorithm, for example: sorting, tree traversals, shortest paths in graphs, etc. Of course, select an algorithm that you know well about it. What it means to animate an algorithm? Let's use sorting algorithms as an example. Each group member can select a different sorting algorithm to animate. The input to the algorithm is a vector of numbers to be sorted. You can represent each number as the height of a rectangle and then animate how the rectangles get re-arranged during the iterations of the sorting algorithm. You can search the web to take a look at several examples of animated sorting algorithms. Below are some useful links to get started:

http://en.wikipedia.org/wiki/Sorting_algorithm

http://www.sorting-algorithms.com/

Minimum Requirements:

a) Each group member has to implement at least one animated algorithm.

b) Each animation has to show the evolution of the algorithm in a graphical way.

c) All the animated algorithms have to be integrated in a single freeglut application.

3. Vector Graphics Editor

Use the freeglut support code as a platform to draw a collection of graphical objects similar to the ones in several vector graphics drawing tools. For example, you could design a collection of object shapes that is normally needed to draw flowcharts: each different type of box and arrow would represent one graphical object and you can then include the ability to connect the objects to each other in meaningful ways so that the boxes of the flowchart can be re-arranged without breaking the arrow connections, etc. You can also work with a selection of standard primitives (circles, rectangles, etc) and properties (filled, hollow, etc) in order to achieve a generic drawing tool. It will make sense to have each group member responsible for preparing a certain number of graphical objects, and one member responsible for integrating everything.

Minimum Requirements:

a) Each graphical object has to integrate meaningful resizing capabilities.

b) The collection of objects has to well work together in order to achieve something interesting (and not just be a random collection of shapes).

c) All the objects have to be integrated in a single freeglut application.