CSSIE 450: Final Project Specification

Assigning Date: November 7, 2001

Due Times:

Proposal - November 14, 2001 at 5:00pm

Hero Object and UI Prototype Demo - November 26, 2001 at 2:00pm

Project Progress Demo - December 3, 2001 at 2:00pm

Final Project Report and Demonstration - December 17, 2001 at 2:00pm

NOTICE: The Due times for the email submission are at 2pm (instead of the usual 5pm!!!)

Objective

You will combine the 2D interactive knowledge and techniques learned in this class to design and implement a real-time 2D interactive graphics application. Your 2D application must contain a state. The purpose of your application is to reflect this state graphically. Through a friendly user interface, you would allow your user control the state of your application. In return, your application will react to the state change by updating the graphical representation in real-time. For example, at this point we have acquired sufficient knowledge to implement a 2D driving simulation program, where the user could navigate a vehicle in a maze. In this case, the car and the surrounding environment will be the state of the application. Through a friendly user interface the user would be able to control the direction and speed of the vehicle. The application would update the position and velocity of the vehicle continuously in real time. You are free to specify and design any application you like, given the technical constraints as specified.

Technical Constraints

1. **One Hero Object.** Your must support at least one Hero object as defined in the MP4 specification. The transformation of at least one of the components of one of the hero object must be completely under user’s control.

2. **Supporting Objects.** Your must support at least two different types of supporting objects. These can be any objects (including OpenGL primitives). Different types mean these two objects must be of different geometric shape. For example, a blue rectangle and a green square have the same geometric shapes and thus do not qualify as two different supporting objects.

3. **Object Positioning.** Your must provide the user with the direct control of the positioning of at least one type of object at all time. Another way of saying this is that the position of one Hero or one Supporting object must be directly under user’s control at all time.

4. **Object Visibility.** The object that is under user’s control can never move out-of-sight. As the object that is under user control approaches the boundary of a viewport, you must support two ways to react. First way to react, you must support scrolling of the world window displayed in the viewport to convey to your user that the view is following the user’s action. Second way to react, when the object directed by the user approaches the bound of your world, you would override user’s command and stop the motion of the object. You must support both of these functionalities.
5. **Object Motion.** You must support non-zero velocity on at least one type of object. This is saying, at least one *Hero* or one *Supporting* object must have non-zero velocity (must be moving).

6. **Object Interaction.** Your *Hero Objects* and the *Supporting Objects* must interact with one another. You are free to design how these objects interact with the minimum requirement that no objects can occupy the same position at any time in your application (no objects can overlap).

7. **Displays for user.** Your drawable must contain at least two window/viewport pairs. You must allow your user zoom in/out, and panning of the WC window in one of the window/viewport pairs. You must support resizing of your drawable in a *graceful* manner. For example, a square will not become a rectangle after drawable re-sizing.

8. **Suspend/Resume/Reset.** Your application must allow your user to suspend and resume the real-time interaction. In addition, at anytime, if the user wishes to do so, they must have the option of re-initializing (reset) the state of the application.

9. **User friendliness.** You must design a friendly graphical user interface (GUI) to interact with your user. Please spend sometime to make sure that information presented in your GUI are organized in a meaningful fashion and only necessary information is presented. It is ok to map direct interaction functionality to the mouse buttons, and map time critical movement commands to the keyboard. Your GUI must reflect the state of the application at all time.

### Time Table and Credit Distributions

<table>
<thead>
<tr>
<th>Date</th>
<th>What</th>
<th>Credit:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nov 14</td>
<td>Final Project Proposal</td>
<td>3%</td>
</tr>
<tr>
<td>Nov 26</td>
<td>Hero Object and UI Prototype Demo</td>
<td>2%</td>
</tr>
<tr>
<td>Dec 3</td>
<td>Progress Demo</td>
<td>1%</td>
</tr>
<tr>
<td>Dec 17</td>
<td>Final Project Report</td>
<td>3%</td>
</tr>
<tr>
<td></td>
<td>Final Project Demo</td>
<td>3%</td>
</tr>
<tr>
<td></td>
<td>Final Project Correctness</td>
<td>18%</td>
</tr>
</tbody>
</table>

**Final Project Proposal (3%):**

1. (1%) - Complete functional specification of your application that satisfies the given technical constraints.
2. (2%) – Complete layout of your graphical user interface design. Including diagrams showing the location of viewports in your drawable, and the design and how to interact with your hero object.

**Hero Object and User Interface Prototype Demo (2%):**

We will dedicate one lecture, where each of you will have 5 minutes to describe your application and demonstrate your hero object and the user interface you have implemented.

1. (1%) – Your presentation. Your presentation should include a combination of slides and demo of the prototype user interface implementation. Half of the credit will come from your peers and half will come from me. You will be graded based on:
   a. Organization of the presentation.
   b. Clarity of describing your application.
2. (1%) – Meeting the progress requirements: completeness of the Hero Object Model and the User Interface that conform to your submitted proposal.

At this stage of software system implementation, you should have basic objects and UI layout completely defined. Your objects may not interact correctly, and your UI may not be *hooked* into your application. It is perfectly acceptable if your prototype user interface implementation is slightly different from your proposal, just describe and justify the differences.
Progress Demo (1%):  
We will dedicate half of a lecture, where each of you will have 3 minutes to demo your progress. At this point, you should support at least:  
1. (0.5%) – Complete Hero Object the correctly supports all user interactions.  
2. (0.5%) – Correct support of multiple Viewport/Window pairs.  
It is very important that during this demo, you compare your current status with that from the first Demo and be able to show actual progress.

Final Project Report (3%):  
This is the final documentation of the system you have implemented. This document should include:  
1. (1%) User manual describing how to use your system, including mouse buttons and keyboard commands.  
2. (2%) A description of the design of your system, including the important data structures/classes and the basic event-driven state transition tables/diagrams.  
You must describe the limitations of your system, and what are the possible features to implement in the next version.

Final Project Demo (3%):  
Each of you will have 5 minutes to describe your application and demonstrate all the functionality of your system. One percent of the credit will come from your peers and the rest will come from me. You will be graded based on:  
1. Organization of the presentation.  
2. Clarity of describing your application.  
3. Completeness of showing all functionality.

Final Project Correctness (18%):  
Finally, I will test run your application and grade it based on completeness and correctness of your system. Here is how credit will be distributed:  

1. Hero and Supporting Objects.  
   Including complete control of at least one of the components.  
   1%  

2. Object Positioning.  
   At least one object’s position is completely under user control.  
   1%  

3. Object Motion.  
   At least one type of object is in constant motion.  
   4%  

4. Object Visibility.  
   Scrolling of world (pushed by object), and override user command.  
   2.5%  

5. Object Interaction.  
   Hero object and supporting object interacts and no object overlaps at any time.  
   5%  

6. Displays for user.  
   At least two viewports, zoom in/out, panning, resizing.  
   3%  

7. Suspend/Resume/Reset.  
   0.5%  

8. User friendliness.  
   1%
**Extra Credit**

You can get up to 2% of extra credit by being *creative* and *resourceful*. Notice, this is 2% of the final score in this class!! Examples of what you can do include:

a. **Saving/Retrival. (1%)** Your application must allow the users to save and re-load their work to files.

b. **Selection. (0.6%)** Your user can *activate/deactivate* any of the visible objects.

c. **Texture Map. (0.2%)** Use of two more texture maps in meaningful ways.

d. **Sound. (0.2%)** Include more than two distinct sound in your application.

There are numerous possibilities for extra credits. Please come talk to me if you have other ideas. I will tell how much your idea worth 😊.

In all cases, you are expected practice what you learned in the Technical Writing class and submit documents with complete and correct English sentences. I will not, and am not qualified, to correct your grammar or presentation style. If I cannot understand what your writing, your will not receive any credit.

Please send me an electronic version of your submissions (either a word-doc or pdf document, and a zipped program source code) on the due date before 5pm. I will make sure your system is built for your in-class demo. As in all cases, if your system does not build, or if there are compilier errors, you may loose up to 5% of the corresponding credit (e.g. 5% of the 2% in User Interface Demo).

Your final project will count 30% towards your final grade for this class.