The Music Box - Rendering

CS 475/CS 675: Computer Graphics - Assignment 2, Part 2

Due Date: 25/10/2013

1 The Music Box

A music box is a small wooden box that when opened plays a musical piece. The apparatus is mechanical. Coordinated to this music, sometimes in a music box there is a small dancer or a couple of dancers that move. The movement is very simple - usually just a rotation, or a few moving libs.

As an example, you can see these:

`http://www.youtube.com/watch?v=yjHFsAUaxMo`
`http://www.youtube.com/watch?v=pdvROzl_SLU`

The overall aim of this entire assignment is to create a short film with the music box lying closed in a room. The animation will show the music box open, and the figure dance to a tune and then close again. You will create all the models, create the scene with other elements, add lighting and texture, script the animation, add music and then generate the movie. In this part of the assignment we, create the room scene in which the music box will be kept, light the room and design a flythrough in the room by making the camera move on a Bezier path to start off our animation.

2 Creating the Room

You have to create a table to place the music box on. Create 3 more pieces of furniture in the room in addition to the table. The room must have four walls and at least 1 door. The room must have one wall mounted light and one lamp. Color and texture all these surfaces appropriately - it is not mandatory to texture every surface, but the idea it to make the scene look aesthetically good.

Create OpenGL lights at the position of the wall light and the position of the lamp light, so that they light up your scene. You must assign keys that can turn on/off both these light sources individually. Make sure that the lighting works with the texture.

3 Creating a Camera Flythrough

1. Implement an interface to create Bezier space curves by clicking control points using the mouse in your room. To make this feasible you will need a way to move around the room with arrow keys and you will need to visualize where in 3D space did you click. So drawing a small sphere where you click is a good idea.
2. Ability to add control points to create the curve is mandatory, but you may add the capability to delete the last clicked point or click and drag already clicked points if you want. After the control points are in place, join them with a single Bezier curve. Design the path to start at the door and end at the music box.
3. Now draw this curve by sampling points on this curve at small δt intervals and joining successive points with a GL_LINES primitive.
4. Add key that starts the camera animation. To animate the camera move the camera eye on the curve by sampling points on the curve densely. The density of sampling can be altered depending on how smoothly you want your camera to move. The camera lookat should be always toward your music box. The up vector can point along whatever direction is up for your scene. At the end of your camera animation, you will have the music box in focus and then the character animation you will make in the next part of your assignment will play.
5. The camera eye can be updated after a fixed interval automatically by using the glutTimerFunc callback, and calling glutPostRedisplay after each update.
6. BONUS: Adding a mirror to the room. Adding shadows to the room when appropriate lights are turned on.
7. NOTE: No Bonus marks will be given unless everything else is perfect - so complete the rest of the assignment before attempting the bonus!

4 Use of OpenGL and GLUT

You are free to use whatever OpenGL and GLUT functions you want. No external modeler like Maya or Blender should be used. All modeling is to be done in OpenGL.

5 Things to avoid:

1. Do not compile and produce an *a.out*. Learn how to use a Makefile.

2. Do not write code for non-inlined functions in header files.

3. Do not write untidy code - you will lose marks if you sprinkle your code with global variables, write code that is difficult to read and is unindented or write code that is not properly structured into objects, classes and files. Only the GLUT callbacks need not be encapsulated in any class and global variables are permitted in the main program file.

4. Do not make a scene that looks exactly similar to some other group's model from the class - both groups will then lose marks. This is an assignment where you have enough chance to show that all of you are original thinkers - please do not hesitate to be creative. So you are free to discuss solution strategies with your classmates but make sure that your code and your models and scene are different.

6 Marking

- Modelling the room with a door: 10 marks
- Modelling the 4 pieces of furniture: 40 marks
- Modelling the wall light and lamp: 20 marks
- Demonstrating that lights work correctly with texture and can be switched on/off: 30 marks
- Interface to create and editor Bezier space curve: 15 marks
- Rendering the camera path: 15 marks
- Camera animation : 50 marks
- Bonus: Mirrors : 10 marks
- Bonus: Shadows : 20 marks
- Total : 180 + 30
- Deduction - I am expecting everybody to write properly formatted, indented and structured code from now on. Untidy code will be penalized.
- Late submission will follow a policy of graceful degradation with a 25% penalty for each day's delay (i.e., you get zero marks if the assignment is more than three days late after the due date.)

TO SUBMIT:

1. A Tar-Gzipped archive of the complete source code (and only source code). It should compile using the given Makefile on any Ubuntu system.

2. A link to a html report page on the assignment that should contain some details about what you implemented and images of some the results that you generated. Put the link in a README file in the archive you submit. Also, include all the keyboard bindings in your code that move the various parts of the robot.

3. The submission will be through the submission portal.