

The Music Box - Modelling

CS 475/CS 675: Computer Graphics - Assignment 2, Part 1

Due Date: 11/10/2012

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=pdvR0zl_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 one dancer model and a box.

2 Modelling Overview

Modeling the box is easy. Create it with cuboids, or quads or any shape you want. The box must have at least on compartment, and lid that can open and close.

Modeling a human figure is difficult, so we will simplify the human figure. An example of such a model for a human is shown in Figure 1, minus the stand.

You can create variations of this and you body parts do not need to have this exact shape. At a minimum it must have a head, neck, torso, 2 upper arms, 2 lower arms, 2 thighs and 2 legs. Hands, feet, geometry for shoulders, elbow and knees are optional - though they do make your model look more realistic. You can also design your own hierarchical tree but name the parts as per convention shown on the figure.

You can add a second dancer to your music box if you want, either now, or later.

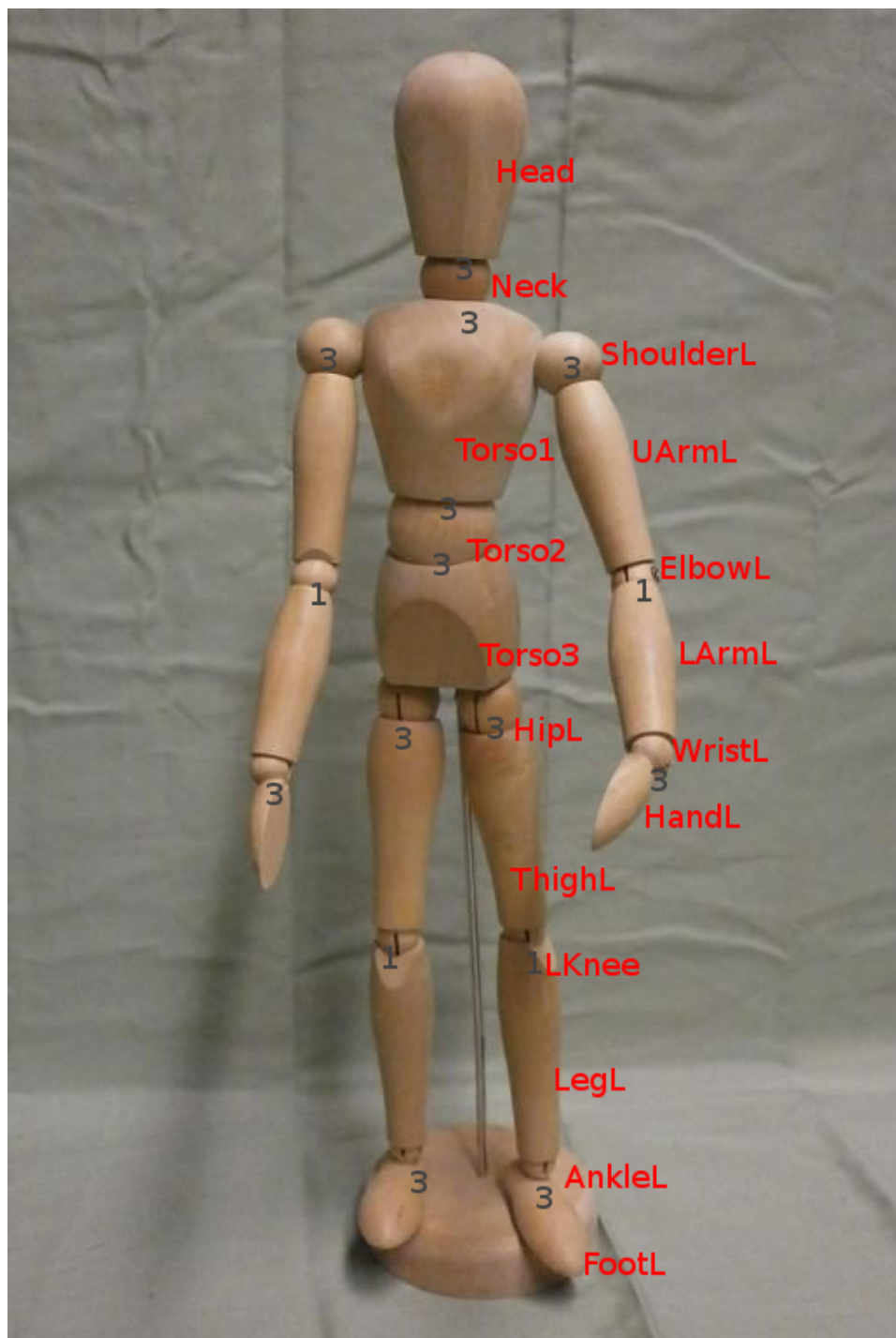


Figure 1: Articulated Human Doll

3 Modeling Requirements:

1. Create a hierarchical model of a box. The box must have at least one compartment and one lid. You must be able to open and close the lid, in increments, by pressing a key.
2. Create a hierarchical model of a human as explained above. You can choose the shape of each part to be made up of whatever primitives you want, but you have to model in OpenGL. No external tools like 3DS Max, or Maya or Blender are allowed. You will be allowed to improve your model later as you learn more things in the course, but you cannot change its basic hierarchy.
3. Learn how to use *display lists* in OpenGL for modelling. Try to encapsulate each rigid body part in a display list.
4. Your human model should at least have the following degrees of freedom (dof) - these are also marked by numbers on the figure:
 - (a) 1 dof at the knees and elbows
 - (b) 3 dof everywhere else.
5. BONUS: Adding more details to the model, like clothes, or shoes or props like sticks and hats.
6. Remember that finally you have to animate the model - so model the hierarchy carefully and choose your parameters and structural constraints wisely. You should not change the model hierarchy from this assignment to the subsequent parts. For this reason, you have to put down (draw the tree) what is your hierarchy in your report.
7. In preparation for the animation, assign keys to vary the values of each of your model parameters, for example, you can use the *uparrow* key to make the model bend backward at the joint between Torso2 and Torso3 and the *downarrowkey* to make the model bend forward. Similarly add keys to change all joint parameters.
8. Use these keys to pose the model in dancing pose.
9. NOTE: No Bonus marks will be given unless everything else is perfect - so complete the rest of the assignment before attempting the bonus!

4 Rendering

1. Texture the box model you built in the previous part of the assignment with a wooden texture. To figure out this out, read Chapter 9 of the OpenGL Programming Guide at <http://www.glprogramming.com/red/chapter09.html>

2. Similarly figure out how to texture the human model with an appropriate texture.
3. Render the box and the human side by side in the same window, against a black background. The user should now be able to use keys and make various parts of each model move as explained above.

5 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.

6 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 model 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 are different.

7 Marking

- Modelling the human properly as per the instructions above: 60 marks
- Modelling the box properly as per the instructions above: 20 marks
- Using Display Lists correctly: 15 marks
- Demonstrating that various parts can be moved using the keyboard: 40 marks
- Texturing the box and the human: 25 marks

- Rendering the two models side by side: 10 marks
- Report with the hierarchical model tree : 10 marks
- Bonus: Extra modelled items : 20 marks
- Total : 180 + 20
- 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.