

# The Music Box - Animation

CS 475/CS 675: Computer Graphics - Assignment 2, Part 3

Due Date: 8/11/2013

## 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=pdvR0zl\\_SLU](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 the animation for the dancer.

## 2 Creating the Keyframes

You can assume that the camera does not move when the figure is animated, for simplicity.

Modify your code so that you save a vector of everything that changes its value in your scene from one frame to the next. E.g., if the degrees of freedom in your scene are the state of a light bulb, angle of the box lid, the root translation and rotation of the dancer, the rotation of the right elbow of the dancer, then one keyframe of the animation will look as:

Bulb On/Off	BoxAngle	RTx	RTy	RTz	RRx	RRy	RRz	ERx	ERy	ERz
0	30	23.4	45.3	-34.9	0.0	10.0	0.0	10.0	0.0	10.5

These frames, which are authored by you, will form the key frames of your animation. Assign a key on the keyboard so that pressing it will save the current frame as a keyframe to a file. Each successive keyframe gets

appended to this file, which will ultimately contains a list of such vectors. This will be done in a “record” mode for your animation code. Save the frames to an ASCII text file called “keyframes.txt”

### 3 Creating the Animation

Creating the animation is now simply a matter of interpolating between keyframes. Note interpolation functions will vary depending upon the kind of variable - e.g., you can linearly interpolate the translation and rotation angles, but the light bulb state may only be 1 or 0.

1. For simplicity you can assume a constant number of frames between each pair of keyframes or you can encode different number of frames between pairs of keyframes manually.
2. Use the `glutTimerFunc` to drive the animation at a particular fps (frames per second) rate. Assign a key that changes to “playback” mode and activates the timer callback, and plays back the animation by reading the keyframes from “keyframes.txt”
3. Look at the notes page on the course page to figure out how you can dump the frames of the animation to a sequence of image files. Join these files into a video and upload the video to YouTube and link the video to your report for the assignment.
4. BONUS: Adding a music track to your animation. You can do this in the program or you can add a track separately to your video.
5. 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 an animation 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

- Making the recording interface and correctly saving the keyframes file: 30 marks
- Correct interpolation: 10 marks
- Correct use of glutTimerFunc for playback: 20 marks
- Making the video: 20 marks
- Bonus: Music : 10 marks
- 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 and the embedded video uploaded to youtube. 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.