Toys - Modelling and Rendering

CS 475/CS 675: Computer Graphics - Assignment 3

Due Date: 31/10/2017

1 Toys

Toy Story (https://en.wikipedia.org/wiki/Toy_Story) was the first feature length animation film ever made. It was released in 1995 and was the first movie produced by Pixar. If you have not seen it, see it.

The overall aim of this entire assignment is to create a short film with two toys (at least). One of those toys must be humanoid (e.g., Buzz Lightyear, from the original films) and the other must be non-humanoid (e.g., Rex, who is a dinosaur toy, also from the original film). Your toy models do not have to be from the original film. You can, e.g., made models of your own toys, from your own childhood.

In short, be creative. The movie is just reference for what is possible.

2 Modelling Overview

The toys in the movie and even real toys, are sometimes very complex, so you do not have to recreate its virtual model in entirety - you can make you own approximations to its shape and degrees of freedom.

The toys can be composed of basic shapes like cuboids, cylinder, spheres, hemispheres, or generic polyhedrons. Divide all faces into triangles, like before.

Design the hierarchical tree of transformation matrices that represent this model. Now choose the parameters that must vary to help you move the toy. Assign keys to vary each of these parameters interactively.

Design and light an environment or scene in which you will place the two toys, for animation in the next assignment. Remember that these are toys - so if you place them in a room, make sure you scale things correctly.

3 Modelling Requirements

1. Create a hierarchical model of two toys 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.

2. Use your VBOs and VAOs judiciously - there are no restrictions on this, but keep in mind that you have to create a transformation hierarchy and the data structure you choose should make it easier for you to code complex sequence of transformations.

3. In addition to global translation and rotation, your humanoid robot model should at least have the following rotational degrees of freedom (dof):

   (a) 1 dof at the knees and elbows

   (b) 3 dof at all other joints

4. For the non-humanoid toy model, you are free to choose any 5 degrees of freedom.

5. Remember that finally you have to animate the model - so model the hierarchies 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 are your hierarchies for each model in your report.

6. 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 in the spine and the *downarrowkey* to make the model bend forward. Similarly add keys to change all joint parameters. Exact key assignment can be of your own design.

7. Create a overall environment to put your toys in - the environment can be indoors i.e., a room. Decide the scale of the room with respect to the toys. The environment can also be outdoors - at minimum you must have a floor or ground plane, and a surrounding environment like walls or sky. Put appropriate textures on the sky, ground, walls - as necessary to make the environment realistic.

4 Rendering Requirements

1. Colour, texture and assign materials to the parts of the toy models appropriately to make them look as desired.

2. Add at least two general directional lights to your scene to light your scene. You can add more if you want but at least two is compulsory.

Add keys that can switch these lights on and off. Make sure the textures are lit properly and that the lighting works with the texture.

3. Add one spotlight to the scene in a location such that it can cover both the toy models, when turned on - again assign keys to turn them on/off. Create geometry for this light that also becomes appropriately brighter when the light is turned on.

4. Render them under a perspective camera inside the created environment. The user should now be able to use keys and make various parts of each model move as explained above.

5. Assign keys to move the camera closer to the toys so that they can be seen from near.

5 Use of OpenGL and GLFW

You are free to use whatever OpenGL and GLFW functions you want. No external modeller like Maya or Blender should be used. All modelling 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.

4. Do not make models that look 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 humanoid toy (transformations and dof): 70 marks

- Modelling the non-humanoid toy (transformations and dof): 50 marks

- Texturing, materials and colours for the toys: 40 marks

- Demonstrating that various parts can be moved using the keyboard: 40 marks
- Scene/Environment model with texture: 20 marks
- 2 Directional lights with texture lighting: 20+20 marks
- Spotlight : 10 marks
- Perspective camera that can be moved: 20 marks
- Report with the hierarchical model trees : 10 marks
- Total : 300
- 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.)

8 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 toys.
3. The submission will be through the submission portal.