Droids - Animation

CS 475/CS 675: Computer Graphics - Assignment 4

Due Date: 6/11/2015

1 Droids

The Star Wars Wikia (www.starwars.wikia.com/wiki/Droid) says, "Droids, short for androids, or also called robots, were mechanical beings, often possessing artificial intelligence. They were used in a variety of roles and environments, often those considered too menial or too dangerous for humans and other species. Droids were also used in fields that required extensive specialization and knowledge, such as medical droids and astromech droids. Droids designed for combat were battle droids."

For an example of the two most famous droids from the Star Wars Universe see the clip below:

`https://youtu.be/R2Mo9LHYPlA?t=151`

The overall aim of this entire assignment is to create a short film with two kinds of droids. One of those droids must be a humanoid droid (e.g., the golden C3PO Protocol Droid in the above clip), the other has to be non-humanoid (e.g., the short, rolling R2D2 Astromech Droid.

2 Animation Requirements

In the previous parts of the assignment you created the droids, an environment/scene for them and added key controls to move the droids. In this part, you create the animation for the droids.

Ideally, you would make a small story that you want to tell through your animation. Then program the keyframe recorder and player as explained below. Save the frames of your animation to images and add a soundtrack. Finally, you must upload your animations to YouTube and submit the YouTube link as explained.

3 Creating the Keyframes

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 the lights, the root translation and rotation of a droid, the rotation of the right elbow of the humanoid droids, then one keyframe of the animation will look as:

```
Light1 On/Off Light2 On/Off RTx   RTy  RTz RRx  RRy RRz  ERx  ERy  ERz
0             0             45.3 -34.9 0.0 10.0 0.0 10.0 0.0  10.5 1.0
```

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”

4 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 for droid parts or camera locations, but the light states 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 glfw timer functions 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.

4. Add a music track to your animation. You can do this in the program or you can add a track separately to your video.

5. Upload the finished video to YouTube. The link to this video must be submitted using the form link given below, **27/11/2015** (copy the link carefully, especially mind the underscore in the middle of the characters):

   https://docs.google.com/forms/d/1jqlyIORTl43FZXbG9R1sp2_zZsm4lRqDEcxphefWX1Q

6. A bonus of upto 20 marks will be awarded if the camera position is animated to move along a Bèzier curve during a portion of the animation. To claim the bonus, you must implement a way to author this curve, then sample and store the samples in a file and then animate the camera position on it. The bonus marks will be awarded only if the rest of the assignment is complete, so please complete the regular assignment before attempting the bonus.

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. Use of GLUT is also not allowed.

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 GLFW callbacks need not be encapsulated in any class and global variables are permitted only 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

- Making the recording interface and correctly saving the keyframes file: 30 marks

- Correct interpolation in playback mode: 10 marks

- Correct use of glfw timers for playback: 20 marks

- Making the video and uploading to Youtube: 20 marks

- HTML Report: 10 marks
- Bonus: Camera Move: 20 marks
- Total: 90+20
- Deduction - I am expecting everybody to write properly formatted, indented and structured code from now on. Untidy code will be penalised.
- 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 droids/change the lights/change the camera.
3. The submission will be through the submission portal.