

# Assignment 3: Project (50 Marks)

CSC 418/2504: Computer Graphics, Fall 2011

Due online by 11:59 PM on Wednesday, November 30, 2011

## Overview

For this project, you will build either animation software, based on keyframe animation of 3D hierarchical models, or rendering software based on the ray tracing algorithm. Each option has a set of required criteria as well as a set of optional criteria; some optional criteria must be completed to get full credit.

We recommend, but do not require, that you work with a partner on this project. Both partners will receive the same mark.

## Animation Option

For this option, your task is to create keyframe animation software that animates 3D geometric models with transformation hierarchies and to use this software to create an interesting animation. When run, your software should play back your animation. Animation data can be stored in data files that your program reads or hard coded into your program. The starter code will include a simple user interface for setting variable values and for controlling playback. Documentation will be included with the starter code that describes how to save individual animation frames to image files and how to create an animation file from the image files. Completion of all requirements should be clearly demonstrated in the animation you create.

### Animation Requirements:

- [10 marks] Create a hierarchical 3D scene with several control variables; include animation of these control variables in your final animation.
- [10 marks] Implement piecewise cubic interpolation (any form is acceptable) to interpolate between adjacent key frame poses or key frame values on individual degrees of freedom, where the keyframe poses or values are specified in a sequence with associated times. Include smooth,  $G^1$  continuous motion in a significant portion of your animation.
- [5 marks]: Report.
- [5 Marks]: Creativity, originality, and special effects not covered in the optional requirements.
- [Up to 30 Marks]: Any subset of the optional requirements. Note that you can get 10 bonus marks on the assignment for completing extra optional requirements.

### Optional Animation Requirements:

- [5 Marks] Motion paths: objects should follow geometric paths in space and stay oriented upright relative to the path.

- [5 Marks] Particle system animation with gravity and randomness.
- [5 Marks] Add collisions to your particle system.
- [10 Marks] Physics-based mass-spring animation; for example a flag in wind.
- [5 Marks] Load and play back hierarchical motion capture data.
- [5 Marks each, up to 2] Animated nonlinear deformations such as bending, twisting, or tapering.

## Rendering Option

This option requires you to create a ray tracer and to use this ray tracer to create interesting images. You may use a file format that describes the scene, or you may hard code the ray tracer to behave differently to generate the necessary images; i.e. by changing the command line arguments. Provide a shell script called *makeImages* that will run your ray tracer to generate all the images needed to demonstrate the requirements. You should also include these images in your report. All requirements should be clearly demonstrated in the images created when we run your shell script.

We recommend you test with very small images (i.e. 200 x 200) and low recursion depths (2-3), as ray tracing can be slow. Please use similar settings wherever possible in your shell script. Feel free to use larger images for your report and submitted images.

### Rendering Requirements:

- [10 marks] Implement ray casting to determine visibility of 3D spheres and triangles with affine transformations.
- [5 Marks] Implement the Phong lighting model to compute approximate local illumination from point light sources
- [5 Marks] Recursive ray tracing to determine shadows from point light sources and specular reflections.
- [5 marks]: Report.
- [5 Marks]: Creativity, originality, special effects not covered in the optional requirements.
- [Up to 30 Marks]: Any subset of the optional requirements. Note that you can get 10 bonus marks on the assignment for completing extra optional requirements.

### Optional Rendering Requirements:

- [10 Marks] Implement path tracing to compute global illumination.
- [10 Marks] Implement intersections with general quadratic surfaces. Include a variety of examples in your results.
- [5 Marks] Extend your triangle intersection tests to handle arbitrary triangle meshes.
- [5 Marks each, up to 3] Depth of field, motion blur, anti-aliasing, or area light sources.
- [5 Marks] Refraction.
- [10 Marks] Texture mapping.
- [5 marks each, up to 3] Implement non-trivial BRDF models such as the Lafortune model and make appropriate comparison images.

## Report

Your report should be a concise, well-written report that describes which technical requirements you met and how you met them. In particular, for each requirement, describe 1) where in your results the requirement is demonstrated, 2) how you achieved the requirement, and 3) where in your source code the requirement has been implemented. Include images in your report (i.e. frames from your animation or ray traced images) to show where the requirement has been met. The report can be submitted in pdf format or as a webpage (a directory containing an index.html file and associated image files). Include your name and your partner's name, if you have a partner, at the start of the report. If you have a partner, document who completed which requirements and who did what creative work. Document any outside software or code you used and how you used it (see the section on this for what is ok and what is not). Document the sources of any data you did not create by hand (i.e. models from the web, motion capture data, textures). Cite any sources you used for looking up algorithms beyond the course texts, notes, and slides. Your report does not need to be excessively detailed or very long, but we want to know how you met the technical requirements, where in the code they were met, and in what images or parts of your animation we can see the results.

## Starter Code

Starter code will be available on the course web site. You are not required to use it, but it will likely make the assignment easier.

## Use of Outside Software or Code

As this project is partially open-ended, we will allow you, within reason, to use outside software or code you have written prior to this term *to assist with the creative work*. The caveat is that you may *not* do so to assist you with any of the programming requirements. For example, you may *not* use such software or code to compute transformations, interpolation results, or intersection tests, but you may use it to create static polygon mesh data that you load into your software to make interesting images. In addition, your overall program must be of your own design and made up of code that you write or based on the starter code. You may not use open source software as a development framework. Be careful—*use of outside software or code to assist you with programming requirements is academic misconduct*. Feel free to ask permission on the course bulletin board if you are not certain whether or not something is ok. Your report must document all use of additional software or code.

## Compilation

Your animation software or ray tracer must compile on CDF Linux using a Makefile, and without requiring the TA to install any additional software. The starter code will include Makefiles that you can use and modify as needed.

## Submission

Place your entire project inside a directory called *a3*. Within the *a3* directory, place your images and/or animation(s) in the directory *a3/results*. Place your source code, Makefile, and any additional data files in the directory *a3/code*. Place your report and any ancillary files in the directory *a3/report*. Use the following commands to submit your project:

```
tar cvfz a3Solution.tgz a3
submit -c csc418h -f -a a3 a3Solution.tgz
submit -c csc2504h -f -a a3 a3Solution.tgz
```

depending on if you are registered for CSC 418 or CSC 2504 respectively.

## Wooden Monkey Competition

The best student work from projects in CSC 418/2504 is documented in the Wooden Monkey Hall of Fame. While your submission for your mark has a deadline of November 30, you may submit additional images or animations *for the Hall of Fame only* until 11:59 PM on Sunday December 4. Only work included in your original submission will be considered for your mark. Additional results can be used to better show off your software's capabilities and are for your own creative benefit. Results included in both your Assignment 3 submission and your Wooden Monkey submission will be considered for the competition, but results included in the Wooden Monkey submission will be ignored during the marking of Assignment 3. To submit additional results, use the following submission commands, assuming your results are contained in the directory *results*:

```
tar cvfz wm.tgz results
submit -c csc418h -f -a wm wm.tgz
submit -c csc2504h -f -a wm wm.tgz
```

depending on if you are registered for CSC 418 or CSC 2504 respectively.

We emphasize that these additional results are for the Wooden Monkey competition only and will not reflect on your Assignment 3 mark.

If you wish to submit higher resolution results than CDF will allow, submit a low-resolution version by the Wooden Monkey deadline and get the higher resolution version to your instructor via website, cd, dvd, or usb drive by the end of the day on Monday, December 5. You may use higher-quality video codecs such as divx or h.264 for Wooden Monkey submissions. Such codecs must be available (and installed) on CDF Linux for the submission to be considered.

## Compatibility

All of your assignments must run on CDF Linux. You are welcome, however, to develop on other platforms and then port to CDF for the final submission. The sample code is designed to work on Linux and Mac OS X machines, but should be portable to other Unix platforms (including Windows with Visual C++). If you are running your own machine, it almost certainly has OpenGL installed; if not, you can search for an rpm or go to <http://www.opengl.org/> (see their getting started FAQ). Mac OS X users might need to install the glui library; see the bulletin board for discussions on how to do this. If you develop on a platform other than CDF's Linux machines, be sure you know how to compile and test your code on CDF well before the deadline. *Your assignment must run on the CDF Linux machines to receive any credit.* Several marks will be deducted if your code does not compile and/or run without modification. If the marker cannot easily figure out how to compile and execute the code, it will most likely receive zero marks. If you choose to develop the assignment on Mac OS X, test the porting of your code to CDF Linux long before the deadline, perhaps even before you have finished the assignment. Often bugs that are "hidden" when compiling on one platform make their presence known by crashing the application on a different platform. There will not be any special conditions allowed for problems encountered while porting at the last minute.