

# CS82/182: Assignment

Out: Jan 15. Due: Jan 24.

## Goal

The goal of this assignment is to ensure that you have enough experience working with an interactive rendering codebase to pursue a project of your own later in the course. For this very reason, I will not give you too many instructions, nor I will indicate how you should achieve the requirements. You should be able to find out how by looking at the documentation and tutorials you'll find on the website below. This is designed to ensure that you can work on your own to achieve the project later on. Note that this assignment is specifically not relate to a particular topic taught in class (that is what the presentations are for).

During you project you are welcome to ask for help from any of your colleagues in the class. Especially if you did not have lots of experience, this may be crucial. When you hand in your homework, though, the work has to be just yours.

## Requirements

You will implement a simple interactive graphics program using the G3D library, that you can find at <http://g3d-cpp.sourceforge.net/>. G3D is a library build on top of OpenGL, one of the two main API for realtime graphics. Rather than building an entire engine, as the name may suggest, G3D provide convenient abstractions over lots of the low level routines found in the API, together with a powerful math library, GUI classes for interaction, and a couple of higher level algorithms (such as shadow mapping and skinned animation).

There are two requirements for this assignment. First, you are to create an interactive program with the following features:

- **Geometry:** Your program should be able to render at least a simple object (such as a sphere) on an “infinite” plane.

- **Shade:** You are to set the Phong parameters of your object and make the sphere glossy. Vertex shading is fine.
- **Lighting:** Your program will shade objects using a couple of lights. Vertex lighting is fine and no shadows are required.
- **Animation:** In your program, the object on the floor should exhibit some simple motion. For example, at the press of a key, it should move from left to right. Any movement is fine for this submission.
- **Interaction:** You should be able to move the camera with any control you like (G3D already has already a camera control similar to Quake or Doom). You should be able to move freely the viewer as the animation plays.

Now that we have the basic done, we will add one feature of your choosing from the following list (implement more for extra credit). Please note that G3D has many helper functions for each of these features.

- **Geometry:** Add textures to objects. For example make the plane look like wood and the object look like marble. G3D has support for this.
- **Shading:** Use pixel shaders to render your object with some interesting effect, such as per-pixel Phong or some procedural texture (you will find examples on the web).
- **Lighting:** Implement shadow mapping. G3D has built in helpers for this.
- **Animation:** Load a skinned mesh and let it run around. There are built characters in G3D (I think they are old Quake characters).
- **Interaction:** Add collision detection to build a tiny first person “shooter” game where your object will do something special when hit, by pressing a button and having the object in the center of view. Please do not add any violent content to the game (like blood splatters)! It is likely that this is the most complex of the tasks, but may be the most fun.

My advise is to start from the tutorials and work your way around the library, which provides most of what you need to do this quickly. My second advise is to start right right away!

While you will write your assignment in C++, you can choose whichever operating system you like to develop the code on. Typically, I suggest to use Windows for graphics, for your welcome to use OS X or Linux. G3D should compile on all those platforms and runs with C++ compilers you can grab for each of them.

Note: in looking through the web, I have not found any better library than G3D. If you know of one that uses a language you are more familiar with, like Java or C#, please let me know. If you prefer to use such library, please contact me asap to discuss it since I may allow it.

## Submission

You are to submit, via email to me, the following three things:

- **Code:** Your code, without excluding the library files, i.e. only the code you wrote.
- **Executable:** An executable of your program and the needed data files. The executable should run on OS X or Windows (I can only run those). I may run your code to test it out.
- **Readme:** A ReadMe file that describes which features you implemented and how to run and use your program.
- **Screenshots:** At least two different screenshots of your program running.

## Grading

During grading, I will mostly make sure that your program works and performs the needed tasks. Essentially, I will grade you by looking at how quickly you put that together, since I am mostly interested in making sure that you can successfully finish your final project. I will not look at code style at all.