

CMPT 485/829: Assignment 2 Specification

Due – Sunday February 26, 2012: 11:59pm

1 Introduction

This assignment is the second part of a multi-assignment mini-project that will span the term. At the very end you will have a single OpenGL program that:

1. Renders an environment in real-time;
2. Has simple camera controls to control where you are looking;
3. Renders shadows;
4. Allows you to ray trace a view with the press of a button; and
5. Incorporates some very basic animation.

In this assignment you are going to focus on adding support for shadows to the program.

2 Framework

As with the first assignment, you are being provided with a C++ framework for a basic OpenGL renderer. There are a few differences between this framework and the solution provided for assignment #1:

- The Camera class has been retooled to provide a camera with the same handedness as OpenGL, has had a bug fixed with respect to setting the depth planes, and has had a function added for setting the position of the camera.
- Shaders for depth-only, and shadow mapping have been provided.
- Addition of `src/assign2.{cpp,h}`, and appropriate changes to `src/main.cpp` to add a skeleton for this assignment.
- The UI support code has gained an approximate frame-limiter. By default, it will aim to call your `repaint()` function at approximately 30fps. This can be adjusted with `UI::setTargetFPS()`.

3 Assignment 2

3.1 Programming Component

For this assignment, you have one main programming goal: To implement omni-directional shadow mapping for a single omni-directional point light.

All of the code that you will have to write for this will be located in `src/ShadowMapping/shadowmap.cpp`, and is described in great detail in the comments. All of the steps that you require are explained.

The main learning objectives for this assignment are for you to learn how to read and understand OpenGL API documentation, and to see what is involved in implementing omnidirectional shadow mapping.

You may modify any code present in the framework, and add any new source files that you require. The places where you will need to add code can be found by grepping the source tree for "TODO".

Note: This implementation is not 100% bug-free. You may encounter odd behaviour if you alter the scene setup – shadows will appear where they should not. As long as your assignment behaves the same as the reference solution it will be considered correct.

3.1.1 CMPT 829 Students

Students enrolled in CMPT 829 have one additional goal: Animate the light source. This will involve adding code to `Assignment2::idle()`, and is described in the comments.

3.2 Written Component

Answer these questions in a separate document entitled `assign2.pdf` that you submit along with your assignment.

1) [5 marks] Change the value of `m_shadowmapSize` in the `Assignment2` constructor to other powers of 2: 32, 64, 128, 256, 512, 1024, and 2048. Describe any differences you observe. Explain the behaviour you observe.

2) [5 marks] Change the min and mag filter for your cube map to `GL_LINEAR` from `GL_NEAREST`, and repeat (1). Describe and explain any differences you observe.

3) [5 marks] Change the value of the "bias" constant in the fragment shader source of the Depth shader to: -0.05, 0.0, and 0.05. Repeat (1) and (2). Describe and explain any differences you observe.

4) [5 marks] Support for shadow mapping was not added to any of the Gouraud shaders in the framework. Why do you think they were not altered to support shadows?

Reset your source tree to 1024x1024 shadow maps, `GL_NEAREST` filter, and a 0.0125 bias before handing in your source tree.

4 Deliverables

1. Upload your completed programming as a zip or tgz file to "Assignment 2" on Moodle. Your source code must compile and run on the Linux machines in S360, and include a Makefile that builds the project.
2. Your written component document should be submitted as a separate file, and must be submitted as a pdf.

5 Evaluation

The assignment will be graded subjectively by the marker with the following breakdown:

1. 25% – Code readability/clarity

2. 75% – Implemented to spec

Each component of the assignment is worth the following:

1. Shadow map – 80 marks
2. Light animation – 20 marks
3. Written component – 20 marks

The maximum grade for CMPT 485 students is 110 of 100 marks; the 829-only component of the assignment may be completed for bonus marks.

The maximum grade for CMPT 829 students is 120 of 120 marks.

Note: Moodle has no way to differentiate students by class in a split class. So, please look at the absolute mark value instead of the percentage grade when looking at your grade.