

CS770/870 Spring 2017

Course Overview

<http://cs.unh.edu/~cs770>

R. Daniel Bergeron

Kingsbury W320

rdb@unh.edu

862-3780 (voice mailbox only)

01/24/17

1

Course Goals

3D computer graphics techniques/algorithms

Interaction: to the extent that it facilitates testing

Course is “implementation-intensive”

Good object-oriented design

Good programming style

Portable code – across platforms

Thorough testing

OpenGL and GLSL: we’ll use OpenGL and GLSL, but they are means to our goals, not goals in and of themselves.

01/24/17

CS770/870 Spring 2017 Bergeron

2

Main Course Tasks

Probably, 6-7 Programs (about 2 weeks each)

First 2-3 will be OpenGL/GLSL learning vehicles

Rest: in-depth exposure to particular algorithms

Exams: 2 exams plus the final

Focused on concepts, algorithms, techniques

01/24/17

CS770/870 Spring 2017 Bergeron

3

Grading

Programs (about 65%)

Correctness

does it implement the specification?

is it robust?

Object-oriented software design

Style (see CS770 Style guidelines)

Tests (about 35%)

Did you answer the questions correctly?

01/24/17

CS770/870 Spring 2017 Bergeron

4

Plagiarism

- Department policy on plagiarism will be strictly enforced
- The design, implementation and debugging of programming assignments should represent each individual's own efforts
- Collaboration should be strictly limited to high-level discussion of concepts and features.
- Use of work from other students or downloaded from the web should be documented with appropriate acknowledgement

Acknowledging Other Work

It is not *per se* wrong to build your software upon other work; in fact, it's often a good idea (as long as the work is in the public domain)

It is critical to acknowledge your sources

- If your algorithm was learned or derived from a book or article, cite that source
- If you make use of utilities or packages, tell us about it
- If your program is a modification or extension of pre-existing code, document that -- and how you improved it

Your grade will be based on what you contributed

Talk to me if you are not certain.

Graphics Software

OpenGL and its associated C/C++ libraries:

GL: basic graphics features

GLSL: gpu programs loaded from cpu application

GLFW: a portable window management tool for GL

LWJGL: Java interface to OpenGL/GLSL/GLFW

OpenGL/GLSL Issues

Traditional OpenGL has a fixed graphics processing pipeline

inappropriate for modern GPU architectures

is nearly all now “deprecated”

NOT O-O, but you need to code in OO style

The new OpenGL allows programmer to determine computation at each stage of pipeline

CPU-based component is significantly smaller

GPU-based component written in GLSL; compiled and loaded at execution time

Much “auxiliary” functionality from the old OpenGL is no longer available; you need to provide it

Java Issues

Two viable options for Java interfaces to OpenGL

JOGL

LWJGL (Lightweight Java Game Library)

JOGL is a complete interface to OpenGL

LWJGL is not a complete implementation

what it provides is OpenGL-compatible

appears to be more efficient

this is our choice; better packaged; easier to install

GLSL/OS Issues

GLSL implementation has been very uneven across major platforms; it's getting better.

Windows supports OpenGL/GLSL 4.3+

Mac OS X supports 4.1+

Fedora Linux (via the Mesa) supports 4.1+

However, it's unpredictable how any OS will handle a low-end gpu.

CS770/870 Framework

This is a computer graphics course

it is NOT an OpenGL or GLSL course

as long as the tool is good enough to learn graphics, that's all we need

As an implementation-intensive course

good software development principles are key

we will only write portable software

Textbook

- There is no specific text for the course.
- Lots of good texts and web sources for conceptual material:
 - basic linear algebra: vector and matrix math
 - object modeling and transformation
 - three dimensional viewing
 - realistic image generation
 - lighting and shading models
 - color theory
 - ray tracing
 - texture mapping
 - anti-aliasing
 - and more

OpenGL/GLSL Resources

- Baker and Cunningham, *Graphics Shaders — Theory and Practice*, 2nd edition. CRC Press. I like this book.
- Angel and Shreiner, *Interactive Computer Graphics, A Top-down approach with WebGL*, 7th edition (2014) uses WebGL and Javascript. Graphics topics in their earlier books were handled reasonably well. This is the only text I know of that combines the graphics content with WebGL.
- There are several recent books based on GLSL, but I haven't looked at them carefully. Most focus only on the package not on graphics in general.
- Tons of web resources.