Carleton University, Computer Science, Winter 2011

**Comp 4002 Real-time 3D Game Engines**

**Times/Instructor**: Wed 6pm (3 hours) Room 213 Tory **W.R. Lalonde**

**Workshop (Assignment help)**: Thursday 6pm (3 hours) Gaming Lab. Help starting the assignment. Watch others work. See status of other people’s work…I will also be there the night before an assignment is due.

**Course Description**: The theory and implementation of real-time 3D game engines giving you substantial experience programming “in the large; i.e., developing a large piece of software” with OpenGL and C++.

**Course Goals**: A project-oriented course to develop a complete C++ game engine with at least one NEAT feature per person. Students can work in teams of 2 for the assignments though it is best to work alone. The neat feature must be done individually but is optional (see **Grading** below). There is NO exam.

**Course Work**: The course will require you to develop software for building games including a game engine, a number of game levels to illustrate its capability, and perhaps support tools for building the game. This will be done through **3 assignments plus a neat feature** equivalent in complexity to 1 assignment. Assignments alone contribute at most B+ to the final mark, the neat features extends that mark to A+. Assignments are a heavy workload that must be done on time to quality for B+. An assignment can be delayed but not missed since every task requires the previous task in working order for it to work too.

A NEAT feature is an effect that is considered cool, novel, interesting, never been done before, or challenging. Start looking for neat effects by looking at recent games such as “Oblivion”, “FarCry2”, “Assassin’s Creed” or “Resistance2”, by reading new game technology books, by trying to think of what would be nice to have in a new game, by imagining what it might be like if our games were technologically equivalent to interactive movies, and by looking on the internet. It can but need not make use of advanced OpenGL facilities. It can either be technologically interesting but not look good artistically speaking, or look really good, or both.

**What DOES NOT constitute a neat feature**: Facilities that though useful but does not require 3D knowledge; e.g., texture readers, model readers, Direct X sound… Also, some things that have been done to death are also not suitable; e.g., a simple particle system. Note that you might need to develop facilities like this. Just don’t count it as part of your neat feature.

**Main Book**: **OpenGL Programming Guide**, Addison-Wesley (Don Mills, Ontario or Reading, Massachusetts), ISBN 0-321-17348-1. If you buy it, get the newest version… There exists an online equivalent (it’s not the same) on the internet at <http://fly.cc.fer.hr/~unreal/theredbook>.

**Game Books**: **Game Programming Gems, II, III, IV, V, VI, and VII**, Charles River Media, Inc.

**3D Games: Real-time Rendering and Software Technology**, Alan Watt, Fabio Policarpo, Addison-Wesley.

**GPU Gems, II, and III**, Addison Wesley.

**AI Game Programming Wisdom, II, and III**, Charles River Media, Inc.

**Mathematics for 3D Game Programming & Computer Graphics**, Eric Lengyel, Charles River Media, Inc.

**Software**: Microsoft Visual Studio C++ (standard or .NET version) with OpenGL (it’s built into Microsoft Windows). Make sure you have HELP installed since it provides documentation for all the OpenGL routines that we use. It is also possible to develop on Unix machines and Macs though the details would be your responsibility. Existing software was designed to run in Visual Studio C++ 6.0 but can be upgraded.

**Laboratory**: Most students have their own computers even though they may use Computer Science laboratories. The speed of your game engine is dictated by the speed of the game card it contains (not the processor). The fastest cards are typically NVIDIA or ATI based.

**Grading**: Doing only assignments on time gives you a mark in the C- to B+ range. Doing assignments plus a neat feature extends the range up to A+. Assignments can be done in groups; a neat feature must be done individually. Assignment marking relies only on checklist items listed in the assignment.

**Assignment marking**: A (failed at most 1 item), B (failed 3), C. Late drops one grade.

**Final marking**: Complete result handed in whether or not NEAT feature done. Assigment marks reviewed and converted to a more exact mark based on the handed in state; e.g. B stays in the B range but becomes either B+, B, or B-.

**Due Date**: **Due** **on Friday April 15 at 6 pm in my personal mailbox** in the office of the School of Computer Science (I will also be physically present at the door of the School’s office at 6pm).

Must be submitted on a CD which includes all the software, maps, and textures needed to rebuild the worlds that demonstrate the capabilities of your game engine including a pre-compiled version. Should include a Microsoft WORD or TEXT file of the form “Jim Smith and Willy Phong.docx” which allows the authors to be determined without opening the file. The file itself should at least contain details about the important keys for running the game, and which neat features are illustrated (provide details about how it works and make sure it’s clear what we need to look for when we run the game) and who is to get credit for which feature.

PROVIDE an e-mail address and phone number on the CD where you can be reached if there is a problem (e.g., if the CD is unreadable or missing files prevent the software from being compiled or linked). Recommendation: copy your CD onto an arbitrary University computer and try it there to make sure you are NOT referencing some file that is only on YOUR computer (it happens a lot)…

**Web**: <http://www.opengl.org>, <http://www.gdmag.com>, www.scs.carleton.ca/~courses, [ati](http://developer.amd.com), [nvidia](http://developer.nvidia.com), <http://www.scs.carleton.ca/~lalonde> (my page), and <http://www.scs.carleton.ca/~lalonde/comp4002> (course page).