§ COMP 365 Computer Graphics §

MW *Lecture/Lab (SC 1315)* – 12:30-1:50

My favorite class by far has been Computer Graphics... Although doing graphics projects was like having a full time job, it was the first time I could show my computer science projects off to non-CS people.

– Sam Von Ehren '10

Who:	Michael Gousie
Where:	Science Center 1325
When:	Mon 2-3:30, Tue 3-4:30, Wed 2-3
	and by appointment
E-mail:	mgousie(at)wheatoncollege(dot)edu
Web:	http://cs.wheatoncollege.edu/mgousie

Content:

This course covers practical and theoretical aspects of computer graphics. Programming projects in C++ with the OpenGL API allow you to create and display complex scenes. These projects will incorporate many techniques from other courses, and some higher mathematics will be involved as well. Homework assignments and exams probe your theoretical knowledge. Some class time will be used for hands-on practice.

Computer graphics is a vast subject area, covering 3D concepts, object modeling, curves, surfaces, lighting, shading, texturing, animation, and on and on. This course covers the basics along with a few specific techniques, such as creating fractals. The last topic covered in this course is the OpenGL Shading Language, which is the new graphics standard API. From this point, you are well positioned for future work in an Independent Study course, graduate work, or programming computer graphics in the real world.

In particular, some of the goals of the course are as follows:

1. Theory: develop facility with relevant mathematics of computer graphics:
   (a) splines using basic Calculus
   (b) transformations and projections using homogeneous coordinates
   (c) 3D rotations using both vector algebra and quaternions
   (d) surface and vertex normals using linear algebra techniques
2. Theory/Practice: learn principles and commonly used computer graphics paradigms and techniques:
   (a) the graphics pipeline
   (b) splines
   (c) ray tracing
   (d) visibility algorithms
3. Programming: gain proficiency with OpenGL, the most widely used platform-independent API. It is used in applications from games to movies to virtual reality, and implemented on platforms that include mobile phones on one end and supercomputers on the other.

Required Texts:

- Guha. *Computer Graphics Through OpenGL: From Theory to Experiments*, 3^{rd} Edition. CRC Press, 2019.
- Prusinkiewicz and Lindenmayer. *The Algorithmic Beauty of Plants*. Springer, 2004. This is available on the course web page and in the Wheaton library. You will need this for one of the projects.

Recommended Texts:

- Sellers, Wright, and Haemel. *OpenGL SuperBible: Comprehensive Tutorial and Reference*, 7^{th} Edition. Addison-Wesley, 2015. This edition covers OpenGL 4.5, much of which is beyond what we will cover in class. You might want to check out the 4^{th} Edition, which more closely matches what we will cover.
- Stroustrup, Bjarne. *The C++ Programming Language*, 4^{th} Edition. Addison-Wesley, 2013. C++ never goes out of style.

Grading:

There will be a midterm and a final exam, both open book/notes, accounting for 40% of your grade. In addition, there will be several homework assignments. Note that a homework problem may include a small amount of programming. These assignments account for 8% of the grade. Of course, there will be lots of programming to do, each project about two weeks in duration. Because graphics programming is very code intensive, you can't afford to wait to start your projects. These projects encompass the remaining 52% of the grade.

Grades will be assigned according to the following scale:

A = 93-100, A- = 90-92, B+ = 87-89, B = 83-86, B- = 80-82, C+ = 77-79, etc.

Exam Schedule:

Exam	**Weight**	**Date**
Midterm	20%	March 17
Final	20%	May 10 @ 9:00 AM

Assignment Schedule:

Program	**Weight**	**Topic** (Subject to change)	**Due Date**
G1	4%	Compiling, fun maps	February 19
G2	8%	Splines	March 5
G3	10%	Fractal plants	March 21
G4	10%	3D transformations and animation	April 2
G5	12%	3D modeling and shading	April 23
G6	8%	Ray tracing or GLSL	May 7

Course Policies:

- You are responsible for all material covered in class. Reading and *understanding* the text(s) is also crucial.
- While some mathematical topics may be covered quickly in class, reasonable proficiency in basic Calculus (functions, derivatives) and Linear Algebra (matrices, vectors) is assumed. Appendix C and D in the text may be of some help if a bit of review is needed.
- If you must miss a quiz or exam for any reason, you must inform me **before** the test. Except in the case of emergency, illness (almost death), or you've fallen into the drain in the Dimple (which is really a portal to another dimension), makeup exams will not be given.
- All programs must be written in ANSI standard C++ and OpenGL. Although development may be done on any computer, the final version must work in Linux. (More on this in class.)
- Written homework should be neat and done on loose-leaf or plain paper, or typed in **plain** text or LaTeX. If done on paper, scan the page(s) and submit one PDF file.
- Assignment due dates are **firm**.
  - Assignments must be submitted electronically by 11:59:59 PM on the due date unless stated otherwise. Use `tar` or `zip` to bundle programs into one neat package for submissions. Hard copy must be handed in only if stated as required on the assignment sheet. Assignments one day late will receive a 15% penalty; anything later will receive a 0.
  - Written homework must be submitted electronically by the date/time specified on the assignment sheet. There are **no** provisions for late homework.
  - There may be some additional digital portions of homework. These must be turned in before the date/time specified on the assignment sheet. There are **no** provisions for such homework turned in late.
  - Due dates are firm, **but** I realize we are in a pandemic! Let me know if you can not turn something in on time because of unforeseen circumstances.
  - A computer crash is not an excuse for late work. It is important that you **back up all of your work!**
  - There will not be any individual "extra credit" work. However, some assignments may have provisions for doing fancier graphics that can earn extra points.
- You are expected to adhere to the Honor Code.
  - Although *discussion* of assignments is encouraged, the *implementation* of programs is to be the result of your own (or your project group's) work.
  - Collaboration on exams is prohibited.
  - You will be required to write and **sign** the pledge on all work turned in: *I have abided by the Wheaton Honor Code in this work.*
  - Any violation of the above guidelines will result in a 0 for that assignment or exam, and/or a failing grade for the course.

- Unless instructed to do so during hands-on sessions, **the use of laptops or other computers/pads is not allowed during lecture.** Arrangements can be made in special circumstances.
- The use of cell phones, iPods, alienPods, iPads, lillyPads, and other personal electronic devices is prohibited during class and exams.
- Please do not disrupt class by leaving/returning, unless there is an emergency. **A phone call or text does not constitute an emergency.**
- Accommodations for disabilities:

> *Wheaton is committed to ensuring equitable access to programs and services and to prohibit discrimination in the recruitment, admission, and education of students with disabilities. Individuals with disabilities requiring accommodations or information on accessibility should contact Autumn Grant, Associate Director for Accessibility Services, at the Filene Center for Academic Advising and Career Services.*
> *∼ accessibility@wheatoncollege.edu or (508) 286-8215 ∼*

Course Schedule (Subject to change, especially weeks 12 & 13):

Week #	Date	Topic	Reading (Guha text)
1		**Introduction**	
	Feb 3	Fun & movies, systems, compilation, event-driven programming	Chapter 1
2		**OpenGL**	
	Feb 8 Feb 10	Basic OpenGL and OOP OpenGL menus, mouse, text	Chapter 2 Chapter 3
3		**Viewing**	
	Feb 15 Feb 17 Feb 19	3D Viewing, windows & viewports, clipping Hermite splines G1 due	Chapters 2 & 3 Sections 19.1–19.2
4		**Moving Stuff**	
	Feb 22 Feb 24	Applied transformations, CTM, stack Viewing transformations, animation	Chapters 4 & 5 Still Chapter 4
5		**Most Important!**	
	Mar 1 Mar 3 Mar 5	Theory of transformations: Basic 3D Theory of transformations: Viewing G2 due	Chapter 5 More Chapter 5; Section 20.1
6		**Fractals**	
	Mar 8 Mar 10	Fractals I, growing plants **No class! Break Day!**	Section 10.5, P & L text
7		**Midterm**	
	Mar 15 Mar 17 Mar 21	Still viewing, catch up Midterm exam G3 due	
8		**Qua-what?**	
	Mar 22 Mar 24	Quaternions Triangulation	Section 6.5 Chapter 8

Course Schedule, continued:

Week #	Date	Topic	Reading (Guha text)
9		**Modeling**	
	Mar 29	Orientation	Chapter 9
	Mar 31	Modeling, Fractals II	Selections, Chapter 10
	Apr 2	G4 due	
10		**Lighting**	
	Apr 5	Lighting theory and practice, shading	Chapter 11
	Apr 7	Computing surface normals	More Chapter 11
11		**More Realism**	
	Apr 12	Computing vertex normals	Still more Chapter 11
	Apr 14	Texture mapping	Chapter 12
12		**Ray Tracing (?)**	
	Apr 19	Basics of ray tracing	Section 21.1
	Apr 21	Ray tracing, POVray	Section 21.2
	Apr 23	G5 due	
13		**OpenGL Shading Language (?)**	
	Apr 26	Basics of GLSL	Chapter 15
	Apr 28	More GLSL	Chapter 16
14		**Behind the Scenes**	
	May 3	Bresenham's line algorithm	Chapter 14
	May 5	Clipping and scanline algorithms	More Chapter 14
	May 7	G6 due	
15		**Final Exam Week**	
	May 10	Final Exam @ 9:00 AM	Happy Summer!