CSCI 4239/5239

Advanced Computer Graphics

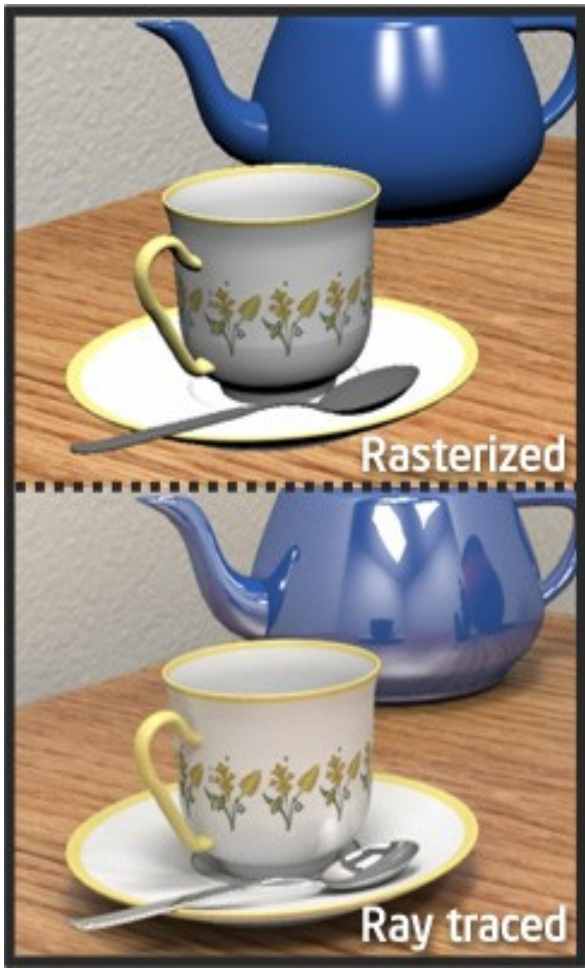
Spring 2025

Instructor

- Willem A (Vlakkies) Schreüder
- Email: vlakkies@colorado.edu
  - Begin subject with 4239 or 5239
  - Resend email not answered promptly
- Office Hours:
  - Monday 3-4pm by Zoom
  - Thursday 2-3pm by Zoom or in ECOT 732
  - Other times by appointment
- Weekday Contact Hours: 6:30am - 9:00pm

Course Objectives

- Explore advanced topics in Computer Graphics
  - Pipeline Programming (Shaders)
  - Embedded System (OpenGL ES)
  - GPU Programming (CUDA&OpenCL)
  - Ray Tracing
  - Special topics
- Assignments: Practical OpenGL
  - Building useful applications
  - Use GLFW to build programs



Course Organization

- Tuesday: Introduction of next topic
  - Lecture
  - Example programs
- Thursday:
  - Discussion of previous homework
  - Presentations

Ungrading

- You self-assign the final grade
  - I may adjust it if I disagree
- Before each Thursday class period
  - Set SMART goals for the next week
  - Evaluate your goals from the pervious week
  - Keep a record of these weekly evaluations that I will review with you as part of the final project during the last week of class
  - This should be a 10 minute activity

S.M.A.R.T. Goals

- Summarize your goals each week
  - **S**pecific - what you want to achieve
  - **M**easureable - evaluating success
  - **A**chievable - doable in a week/semester
  - **R**elevant - somewhat on topic
  - **T**imely - plan your week
- You weekly journal should follow this pattern

Assumptions

- You need to be fluent in C/C++
  - Examples are in C or C++
  - You can do assignments in any language
    - I may need help getting it to work on my system
- You need to be comfortable with OpenGL
  - CSCI 4229/5229 or equivalent
  - You need a working OpenGL environment

Class Attendance

- **Attendance is expected**
  - I don’t typically take attendance
- More of a seminar than a lecture
  - Participation is important
- If you are legitimately sick, use Zoom
  - Email me well before class
  - I will record the lecture if there is a compelling reason

Assignment Expectations

- The goal is to impress your friends
- Assignments **must** be submitted on time unless prior arrangements are made
  - Due by 23:59 Wednesday
  - Grace period until Thursday at 8:00am
- Assignments must be completed individually
  - Stealing ideas are encouraged
  - Code reuse with attribution is permitted
- I will review your submissions before class

Code Reuse

- Code from the internet or class may be used
  - You take responsibility for any bugs in the code
    - That includes bugs in my code
  - Make the code your own
    - Understand it
    - Format it consistently
  - **Improve upon what you found**
    - **I may ask what improvements you made**
  - **Submitting code without crediting the source is violation of the CU honor code**
- The assignment is a minimum requirement

Code Expectations

- I expect professional standards in coding
  - Informative comments
  - Consistent formatting
    - Expand tabs
  - Clean code
    - Clean out unused code
- Good code organization
- Appropriate to the problem at hand
- See ***Expectations*** on Canvas
- You need to understand ***every line***

Text

- OpenGL Programming Guide (9ed)
  - Kessenich, Sellers & Schreiner
  - “OpenGL Vermillion Book”
  - Implementing Shaders using GLSL
  - Don't get an older edition
- Ray Tracing from the Ground Up
  - Kevin Suffern
  - Theory and practice of ray tracing
- Recommended by not required

Other Texts

- OpenGL SuperBible: Comprehensive Tutorial and Reference (7ed)
  - Sellers, Wright & Haemel
  - Good all-round theory and applications
- Graphics Shaders: Theory and Practice (2ed)
  - Bailey & Cunningham
  - Great shader examples

Other Texts

- OpenGL ES 3.0 Programming Guide
  - Ginsburg & Purnomo
  - “OpenGL Purple Book”
  - Has a chapter specific to the iPhone
- WebGL Programming Guide
  - Matsuda & Lea

Other Texts

- Programming Massively Parallel Processors
  - Kirk & Hwu
  - Explains GPU programming using CUDA
  - Shows how to adopt OpenCL
- CUDA by Example
  - Sanders and Kandrot
  - Great introduction using examples

Other Texts

- Advanced Graphics Programming Using OpenGL
  - Tom McReynolds and David Blythe
  - Great reference for miscellaneous advanced topics
- Physically Based Rendering
  - Pharr, Jakob and Humpfreys
  - Only for PBRT homework
  - 3rd edition for PBRTv3

OpenGL Resources

- www.google.com
  - Need I say more?
- www.opengl.org
  - Code and tutorials
- nehe.gamedev.net
  www.lighthouse3d.com
  - Excellent tutorials
- www.mesa3d.org
  - Code of “internals”
- www.prinmath.com/csci5229
  - Example programs from CSCI 4229/5229

Assignment 0

- Due: **Today Jan 14** by 23:59
- Check your Canvas notification settings
  - Set notifications to immediate
- Submit
  - Current picture
  - Your study area
  - Platform (Hardware, Graphics, OS, ...)
  - Why are you taking this class?
  - Does office hours work for you?

My information

- Mathematical modeling and data analysis
  - PhD Computational Fluid Dynamics [1986]
  - PhD Parallel Systems (*CU Boulder*) [2005]
  - President of *Principia Mathematica*
- Use graphics for scientific visualization
- Open source bigot
- Program in C, C++, Fortran, Perl & Python
- Outside interests
  - Aviation
  - Amateur radio

Hardware Requirements

- You need hardware that will run shaders well
  - Integrated graphics may be marginal
  - Graphics cards from the last 5 years should be OK
  - GPU computing needs high end hardware
  - A VM is probably not going to cut it
- Try on different hardware
  - AMD/nVidia/Intel sometimes behave differently
  - I have nVidia hardware

Examples use glfw

- Why drop GLUT?
  - Apple support for GLUT is waning
  - It is easy to use, but limited capabilities
- Why glfw
  - It is cross platform: Linux/WinX/OSX/iOS/...
  - Very light weight wrapper to OpenGL
  - Does not do sound, load images, etc
  - Actively being developed (Vulcan is coming...)
- Can I use SDL or another wrapper?
  - As long as it is cross platform

Installing glfw

- *http://www.glfw.org/*
- Ubuntu:
  - apt-get install glfw3-dev
- OSX
  - Install Xcode with command line tools
  - Install homebrew
  - Install toolchain, glfw and glew
- Windows
  - Install MSYS2/MinGW
  - Install toolchain, glfw and glew with pacman

OpenGL Extension Wrangler (GLEW)

- Maps OpenGL extensions at run time
  - Provides headers for latest OpenGL
  - Finds vendor support at run time
- Check support for specific functions or OpenGL version at run time
  - Crashes if unsupported features are used
- Use only if you have to (Windows mostly)
  - Set -dUSEGLEW to selectively invoke it
  - Do NOT require GLEW (I don't need it)
  - See Canvas for installation instructions

CSCIx239 Library

- Includes GLFW and GLEW headers
- Many convenience functions
  - InitWindow starts GLFW and GLEW
  - Projection, Print, Fatal, ErrCheck, ...
  - Load textures and OBJs
  - Simple objects (Cube, Sphere, ...)
  - Compile Shaders
  - Matrix operations
  - Performance (FPS, elapsed)
- **Make sure you know what it does**

OpenGL Verions

- I will use different OpenGL versions depending on what is convenient for the problem at hand
  - OpenGL 2.x
    - Feature rich
    - Flat learning curve
    - Convenient in many applications
  - OpenGL 3.x or OpenGL 4.x
    - Somewhat different syntax
    - Needed for advanced shaders
- OpenGL Core & Compatibility Profiles
- **You can use whatever version you want**

Assignment 1

- Due: **Tomorrow** Wednesday January 15
- NDC to RGB shader
  - For every point on the objects, the color should be determined by its position in normalized device coordinates
- The goal is to make this as short and elegant as possible
  - Shader Golf
  - Figure this out for yourself
  - Make every operation count
- Test your toolchain

Nuts and Bolts

- Complete assignments on any platform
  - Assignments reviewed under Ubuntu 22.04.3 LTS
  - Ubuntu provides glfw 3.3
- Submit using Canvas
  - ZIP without creating an extra folder
  - Name projects hw1, hw2, … (lower case)
  - Include all source code, makefile and data files
  - Set window title to *Homework X: Your Name*
- Include number of hours spent on assignment
- ***Check my feedback and resubmit if requested***

Project

- Should be a program with a significant graphics component
  - Something useful in your research/work
  - Graphical front end to simulation
  - Graphical portion of a game
  - Expect more from graduate students
- Deadlines
  - Proposal: Thursday March 20
  - Progress: Thursday April 17
  - Final: Monday April 28

A few hints

- My machine runs Linux x86_64
  - gcc/g++ with nVidia & GLX
    - -Wall is a **really** good idea
  - case sensitive file names
  - int=32bit, long=64bit
  - little-endian
  - fairly good performance
- How to make my life easier
  - Try it on another machine
  - Stick to C/C++ unless you have a good reason
- **Maintain thy backups...**

What to Present

- Should be (mostly) the assigned topic
  - Rabbit holes can be very interesting
  - Keep it within reach of the class
- Show what you did for the assignment
  - Cover principles or theory I omitted
  - Show and describe code of interest
  - Demonstrate “gotchas” you encountered
  - Impress your friends
- Keep it interesting