Outline

- What is Computer Graphics?
- Class Overview (Syllabus)
- Project 1A Overview
Outline

- What is Computer Graphics?
- Class Overview (Syllabus)
- Project 1A Overview
Computer Graphics

- Defined: pictorial computer output produced, through the use of software, on a display screen, plotter, or printer.
What is computer graphics good for?

- Ed Angel book:
  - Display of information
  - Design
  - Simulation and animation
  - User interfaces
What are the challenges?

- How do you transform geometric primitives to images?
  - We will focus on the “rasterization” method, which is what specialized hardware (GPUs) use
    - The next lecture will give an overview of rasterization
    - In the coming weeks, we will learn how to transform the geometry so it can be rendered, which will require linear algebraic techniques
  - In the latter part of the course, I will give an overview of an alternate method, called ray tracing
What are the challenges?

How do you do it quickly?

→ we will explore optimizations, including
  - Texture Mapping
  - Clipping
  - Advanced data structures for geometry organization

→ we will learn how to use the GPU to accelerate rendering, through the OpenGL interface
What are the challenges?

- How do you make it pretty?
  - we will explore different shading models
  - we will learn how to control light sources
  - in the latter part of the course, I will discuss how to use transparency
  - in the latter part of the course, I will discuss how to add shadows
We are starting this class off “old school.”

Drum Plotter, which follows “Pen-Plotter Model”
We are starting this class off “old school.”

- Pen-Plotter Model has two commands:
  - Moveto\((x, y)\)
  - Lineto\((x, y)\)

- Example:
  - Moveto\((0,0)\)
  - Lineto\((1,0)\)
  - Lineto\((1,1)\)
  - Lineto\((0,1)\)
  - Lineto\((0,0)\)
We are starting this class off “old school.”
We are starting this class off “old school.”

- We will start by setting the colors for individual pixels in an image.
- Then we will learn how to take geometries in specific orientations and render them (think dot matrix printer).
- Then we will learn how to take geometries in any orientation and render them. (i.e., arbitrary camera locations)
- Then we will learn how to program a GPU to do all this in hardware.
- Then we will do cool things.
Outline

- What is Computer Graphics?
- Class Overview (Syllabus)
- Project 1A Overview
Syllabus

- All syllabus information is located at:
  - http://ix.cs.uoregon.edu/~hank/441

- Please email a google account to me:
  - hank@cs.uoregon.edu
  - Subject line: google account for CIS 441
Expectations

- This is a projects-driven class. The projects will help you learn the theory of computer graphics, but they will also help you become better programmers, and provide you with experiences, anecdotes, and images that will impress potential employers.

- The grading is designed to make sure you are keeping up with the assignments. Staying on top of the projects will be critical to succeeding in this class.
Expectations

- The projects in this class will be hard work. It is difficult to quote exactly how much time, since there is variation in background and programming skill. I expect those who have less developed programming skills will find this class to be a considerable effort, but also that they will have significant improvement by the end of the course.
My two worst “first day” experiences

- ECS 110: Data structures & C++
- ECS 152: Networking

- As a parent, I’ve learned you always want to put your kids in a winning situations and you should never, never ask them to do something you can’t do.

- We are a community. Help each other.

- I am a resource.
  - I will have surge OH this first week to get over some of the initial barriers.
Norms for this class

- Please ask questions
- Please ask me to slow down
- Please give feedback
This Class In a Nutshell...

- 1st Half: Learn the theory behind computer graphics and implement it in software
- Middle: Learn OpenGL and write an OpenGL program
- End: Design and implement a final project
- Final: administered on the last day of instruction.

Most of the learning will happen with projects. The lectures are designed to help you do the projects.
I recommend the Ed Angel book, but it is not required and students who comprehend the lectures will not need it.

We will also use the graphics notes from Ken Joy.

PowerPoint lectures will be posted online.

Some lectures will be chalkboard.
Grading

- Project 1: 30%
- Project 2: 15%
- Proposal for Project 3: 0%
- Final: 25%
- Project 3: 30%
Project 1 (30%): Implement GPU algorithms in Software

- Project 1A (due April 5): Install and run example program (2%)
- Project 1B (due April 10): Rasterize screen-space triangles (7%)
- Project 1C (due April 12): Rasterize colored triangles (2%)
- Project 1D (due April 19): Add Gouraud shading (4%)
- Project 1E (due April 26): Add world-space-to-camera-space transforms (8%)
- Project 1F (due May 3rd): Add Phong shading (7%)

These projects may change based on class progress.
Project 1 (30%): Implement GPU algorithms in Software

- Project 1A (due April 5): Install and run example program (2%)
- Project 1B (due April 10): Rasterize screen-space triangles (7%)
- Project 1C (due April 12): Rasterize colored triangles (2%)
- Project 1D (due April 19): Add Gouraud shading (4%)
- Project 1E (due April 26): Add world-space-to-camera-space transforms (8%)
- Project 1F (due May 3rd): Add Phong shading (7%)

May 3rd
Projects 2, 3, and Final

- Project 2: 15%
  - Project 2A (due May 10th): Basic OpenGL program (8%)
  - Project 2B (due May 15th): Advanced OpenGL program (7%)
- Final Project Proposal (due May 17th): 0%
- Final (June 7th): 25%
- Project 3: 30%
  - Student-defined projects due on Finals week (30%)
Final Project Ideas

- Make a video game
- Make a screen saver
- Do some advanced graphics techniques
- Model Deschutes
- Do some visualization
Grading

- Some projects will be graded via "show-and-tell" with you and me, either in my office or in the computer lab.
  - Project 1A: you can just email me if you want

- If you have a laptop, you will come to my office and show me your project and we will discuss it.
  - This allows you to implement the projects on whatever platform you prefer.

- The final projects will be demonstrated in front of the whole class on the day of the Final.
Office Hours

- Starting the quarter with “Surge” Office Hours:
  - Weds & Fri, 10-11:30 & 2-4, Deschutes 358
  - Later in the quarter, grading will occur during these times as well.
  - Final OH TBD based on class needs

- There are no OH this afternoon, but I will hang around after class to discuss late adds.
- There will be an extra OH tomorrow 12-2:30.
Working Together

- All projects are individual projects.
- Copying code from other students is cheating.
- However: I highly encourage you to discuss your roadblocks with each other and lean on each other to figure out solutions to your problems.
I have set up a forum on Blackboard.

I will monitor (and respond) to the forum and encourage you all to do the same.

I may award extra credit to students who are particularly helpful on the forum.

The amount of credit will vary based on involvement, with a maximum of 5%.
Evaluation Criteria

- For Project 1, I will provide some test configurations. I will also provide the correct images and share them with you.
  - If your program produces the correct images, you will receive full credit.

- For Project 2, you will be evaluated on how many of the requested OpenGL features you implement.

- For Project 3, you will be evaluated on the results of your project, including how ambitious your undertaking is.
  - We will jointly establish if you are embarking on a project where you will be able to earn the full 30% when we discuss your Final Project Proposal on the week of May 20th.
Evaluation Criteria, pt 2

- I withhold the right to provide reduced scores if I believe you have borrowed too much of your implementation from a peer, the internet, or other sources.
  - Although borrowing code from elsewhere is a great strategy in the real world, it is not going to help you learn computer graphics or how to program.
- I will be looking at your source code and asking you about your implementation travails.
Late Passes

- You have 3 "late passes."

- Late passes allow you to turn in your project after the due date for full credit.
  - Submitting a project with a Weds deadline on Friday (i.e., two days later) costs one late pass.
  - Submitting a project with a Friday deadline on the following Weds (i.e., five days later) costs two late passes.
  - Projects with a Friday deadline can possibly be submitted on a Monday for one late pass, but my availability on Mondays is not guaranteed and you may be forced to wait until Weds (and pay two late passes).

- If you run out of late passes, then you may continue to earn half credit on any project.

- Every unused late pass is worth 1% extra credit.
Class Summary

- This class will teach you the theory of computer graphics
- This class will improve your programming skills
- This class may help you land a job
- This class will require a lot of work
Outline

- What is Computer Graphics?
- Class Overview (Syllabus)
- Project 1A Overview
Goal: write a specific image

Due: midnight, Friday, April 5th … in 60 hours (!)

% of grade: 2%

Q: Why do I only get 2 days to complete this project?

A: We need to get multi-platform issues shaken out ASAP.

May be a little painful
Project #1A: background

- **Definitions:**
  - Image: 2D array of pixels
  - Pixel: A minute area of illumination on a display screen, one of many from which an image is composed.

- Pixels are made up of three colors: Red, Green, Blue (RGB)

- Amount of each color scored from 0 to 1
  - 100% Red + 100% Green + 0% Blue = Yellow
  - 100% Red + 0% Green + 100% Blue = Purple
  - 0% Red + 100% Green + 0% Blue = Cyan
  - 100% Red + 100% Blue + 100% Green = White
Colors are 0->1, but how much resolution is needed? How many bits should you use to represent the color?

- Can your eye tell the difference between 8 bits and 32 bits?
  - No. Human eye can distinguish ~10M colors.
  - 8 bits * 3 colors = 24 bits = ~16M colors.

- Red = (255,0,0)
- Green = (0,255,0)
- Blue = (0,0,255)
An “M by N” 8 bit image consists of MxNx3 bytes.

- It is stored as:
  
  \[
P_0/R, P_0/G, P_0/B, P_1/R, P_1/G, P_1/B, \ldots, P(MxN)/R, 
P(MxN)/G, P(MxN)/B
  \]

- P0 is the bottom, left pixel
- P(M-1) is the bottom, right pixel
- P((MxN)-M+1) is the top, left pixel
- P(MxN) is the top, right pixel
Project #1A: background

- The red contributions are called the "red channel".
  - Ditto blue & green.
- There are 3 channels in the image described above.
- There is sometimes a fourth channel, called "alpha"
  - It is used for transparency.
- Images are either RGB or RGBA
Project #1A in a nutshell

Assignment:
- Install CMake
- Install VTK
- Modify template program to output specific image
What is \textit{CMake}?

- Cmake is a cross-platform, open-source build system.
- \texttt{CMake} is a family of tools designed to build, test and package software.
- \texttt{CMake} is used to control the software compilation process using simple platform and compiler independent configuration files.
- \texttt{CMake} generates native makefiles and workspaces that can be used in the compiler environment of your choice.
How do you install CMake?

- Go to www.cmake.org & follow the directions
What is the Visualization Toolkit?

- The Visualization Toolkit (VTK) is an open-source, freely available software system for 3D computer graphics, image processing and visualization.
- VTK consists of a C++ class library and several interpreted interface layers including Tcl/Tk, Java, and Python.
- VTK is cross-platform and runs on Linux, Windows, Mac and Unix platforms.
How do you install VTK?

- Go to [www.vtk.org](http://www.vtk.org), go to Resources->Download and follow the directions.
What is the image I’m supposed to make?

- R=0
  - (0,0,0)
  - (0,0,128)
  - (0,0,255)
  - (0,128,0)
  - (0,128,128)
  - (0,128,255)
  - (0,255,0)
  - (0,255,128)
  - (0,255,255)
  - (128,0,0)
  - (128,0,128)
  - (128,0,255)
  - (128,128,0)
  - (128,128,128)
  - (128,128,255)
  - (128,255,0)
  - (128,255,128)
  - (128,255,255)
  - (255,0,0)
  - (255,0,128)
  - (255,0,255)
  - (255,128,0)
  - (255,128,128)
  - (255,128,255)
  - (255,255,0)
  - (255,255,128)
  - (255,255,255)
What do I do again?

- Install CMake & VTK.
- Download “project1A.cxx” from class website
- Download “CMakeLists.txt” from class website
- Run CMake
- Modify project1A.cxx to complete the assignment
- And…
- Mail me the source and image result by Friday midnight.
- (in the future, we will set up appointments for showing projects off)
What should you do if you run into trouble?

1) Start with the class forum
2) OH this week:
   ◆ Thurs: 12-2:30
   ◆ Fri: 10:30-11:30
   ◆ Fri: 2-4
3) Email me: hank@cs.uoregon.edu
Don’t forget

- Please email a google account to me:
  - hank@cs.uoregon.edu
  - Subject line: google account for CIS 441

- This lecture is available online.