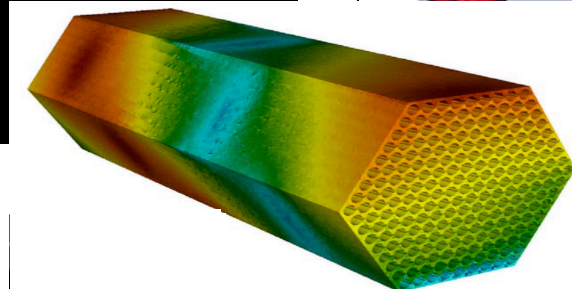
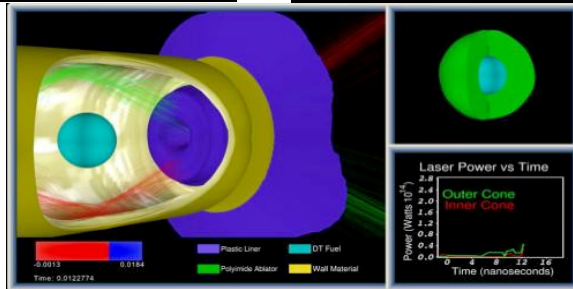
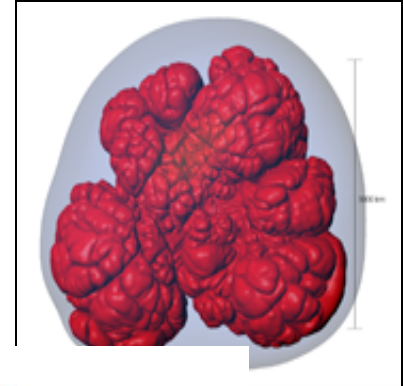
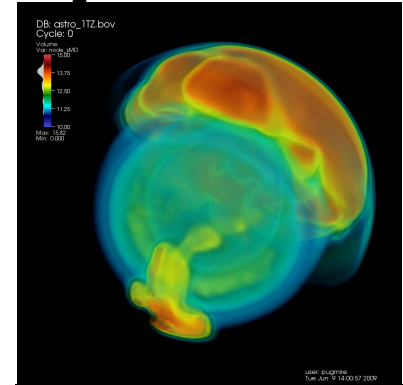
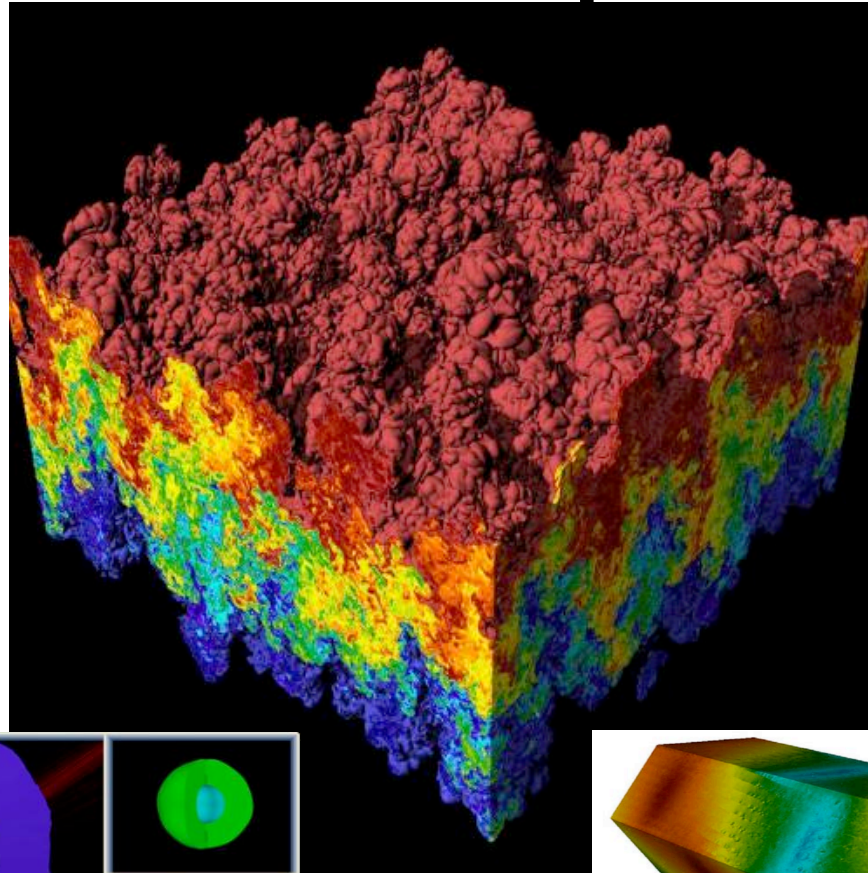
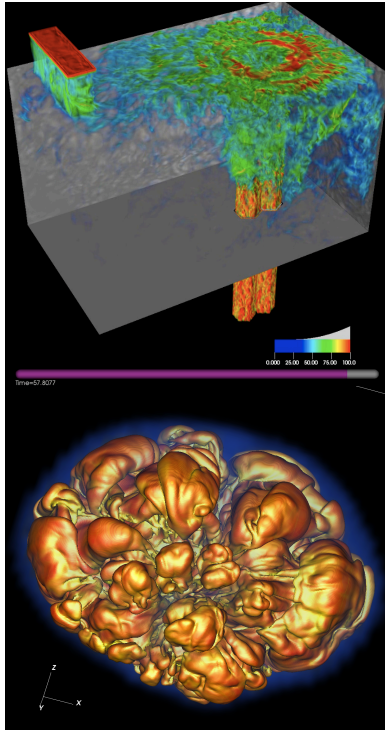




# Introduction to Computer Graphics





# Hi, I'm Hank...



- Professor of CIS at UO
- Arrived at UO March 2013
- Research interests include visualization, high-performance computing, computer graphics
- Director of CDUX Research Group (3 PhD, 1 MS, 1 BS)
  - + many alumni: 8 PhD, 4 MS, and 9 BS



# My Background

- Previously:
  - Lawrence Livermore, 1999-2009
  - Lawrence Berkeley, 2009-2016
  - UC Davis 2009-2013
- Education:
  - B.S. (CS/Math), 1994-1999
  - Ph.D., 2000-2006

I spent 14 years programming,  
almost exclusively using C, C++, and Unix



# My Recent Background

- Haven't taught for one year
  - sabbatical
  - paternity leave



# Online Delivery

- I am concerned about cheating



# Online Delivery

- I am concerned about cheating
- I am concerned about building community
  - We may turn Thursdays into community days



# Online Delivery

- I am concerned about cheating
- I am concerned about building community
  - We may turn Thursdays into community days
- I am concerned about maintaining the quality of this course
  - Delivering material well
  - Supporting you
  - Still a “transformative” course / high bar / important outcomes



# Online Delivery

- I am concerned about cheating
- I am concerned about building community
  - We may turn Thursdays into community days
- I am concerned about maintaining the quality of this course
  - Delivering material well
  - Supporting you
  - Still a “transformative” course / high bar / important outcomes
- Feedback welcome!





# Instruction Team



Hank Childs



Abhishek Yenpure



# These slides are online

## Classes

CIS 410/510 Computational Science	✓				CIS 314, CIS 422, or instructor's consent.
CIS 410/510 Computer and Network Security II				✓	
CIS 410/510 Computer Vision				✓	
CIS 410/510 Internet Measurements		✓			
CIS 410/510 Introduction to Game Programming				✓	
CIS 410/510 Multi-agent Systems				✓	
CIS 410/510 Natural Language Processing		✓			
CIS 410/510 Program Analysis and Transformation				✓	
CIS 410/510 Secure Software Development				✓	CIS 322, CIS 330, and CIS399 Applied Cryptography
CIS 410/510 Selected Topics on Optimization	✓				
CIS 413/513 Advanced Data Structures		✓			CIS 315.
CIS 420/520 Automata Theory	✓				CIS 315.
CIS 422/522 Software Methodology	✓	✓		✓	CIS 313.
CIS 429/529 Computer Architecture				✓	CIS 330.
CIS 431/531 Introduction to Parallel Computing		✓			CIS 330.
CIS 432/532 Intro to Computer Networks	✓				CIS 330. CIS 415 recommended.
CIS 433/533 Computer & Network Security		✓		✓	CIS 415.
CIS 441/541 Intro Computer Graphics				✓	CIS 330.
CIS 443/543 User Interfaces	✓				CIS 313.
CIS 445/545 Modeling and Simulation				✓	CIS 315, 330.
CIS 451/551 Database Processing	✓				CIS 313, 314.
CIS 471/571 Introduction to Artificial Intelligence	✓	✓			CIS 315.
CIS 472/572 Machine Learning	✓			✓	CIS 315.
CIS 473/573 Probabilistic Methods		✓			CIS 315.

Graduate



## CIS 441/541 (S21), INTRODUCTION TO COMPUTER GRAPHICS

Time: Tuesdays and Thursday 8:15-9:45  
Location: Virtual  
Instructor: Hank Childs  
Hank's Office Hours: TBD  
OH Location: Virtual

- [Syllabus](#): objectives, expectations, grading, academic misconduct, late policy
- [Projects](#)
- [Lectures](#)

### Contact Information

**Email:** [hank at uoregon.edu](mailto:hank@uoregon.edu)  
**Phone:** (541) 346-3414 (terrible way to get in touch with me!)

**Address:**  
301 Deschutes Hall  
1202 University of Oregon  
Eugene, OR 97403-1212 USA

## CIS 441/541: LECTURES

### Lectures

- Lecture 1, March 30th: intro/syllabus/proj 1A [ [PPTX](#) | [PDF](#) ]
  - [Project 1A Orientation Video](#)
- Online Lecture A, March 30th: [images](#)
  - Slides for this lecture are the three slides on images from Lecture 1



# Outline

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



# Outline

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



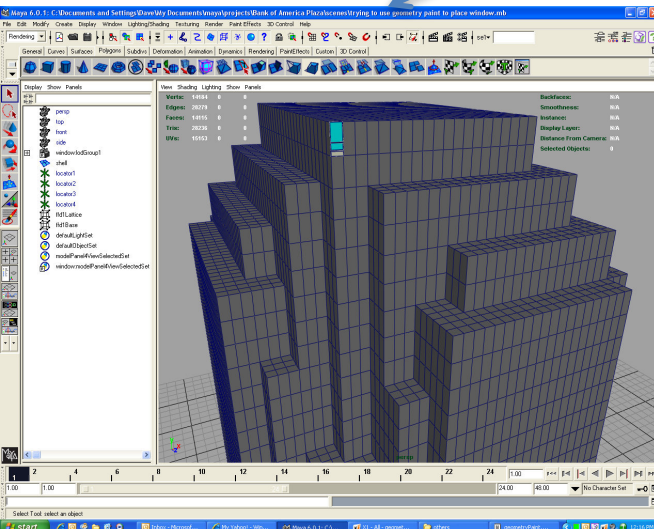
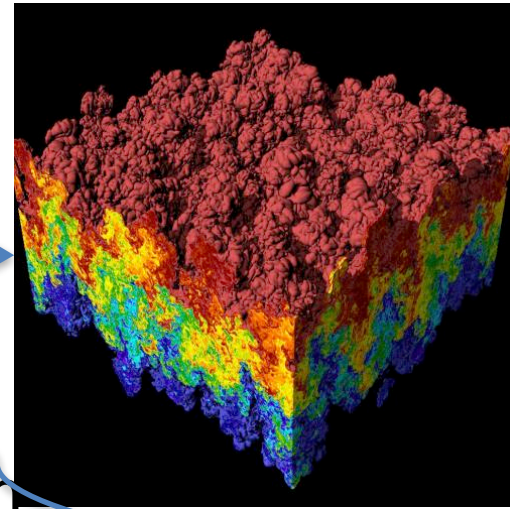
# Computer Graphics

- 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”

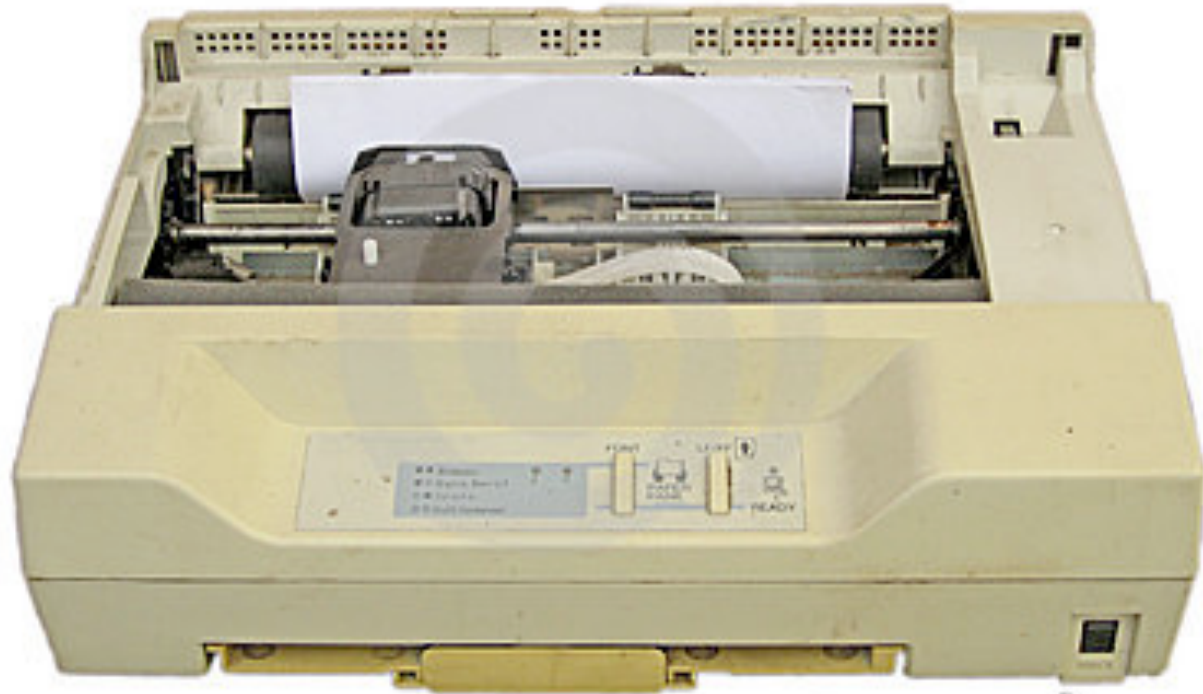


# We are starting this class off “old school.”

- 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.”



[dreamstime.com](http://dreamstime.com)

Dot Matrix Printer



# 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 online at:
  - <https://classes.cs.uoregon.edu/21S/cis441/>





# 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.



# Norms for this class

- Please ask questions
- Please ask me to slow down
- Please give feedback
  
- You are welcome to call me “Hank”



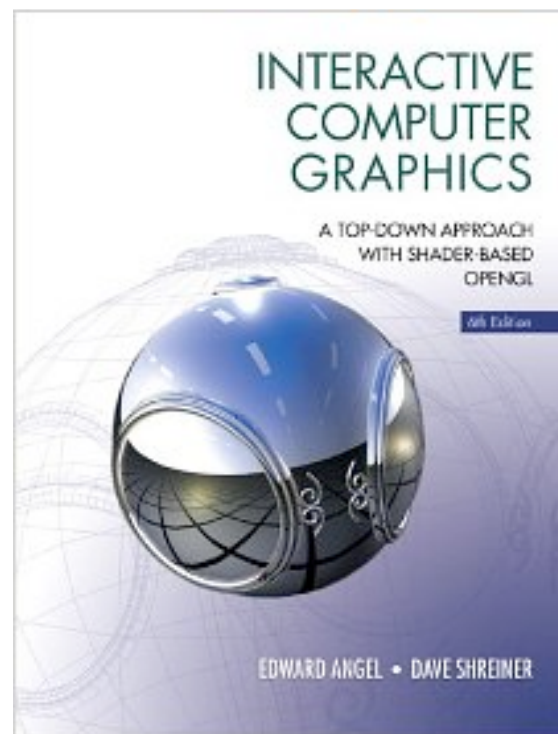
# This Class In a Nutshell...

- 1<sup>st</sup> 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: present final project
- Throughout: ~5 Quizzes

Most of the learning will happen with projects. The lectures are designed to help you do the projects.

# Course Materials

- I enjoy 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 points
- Project G (grad students): 15 points
- Project 2: 15 points
- Proposal for Project 3: 0 points
- Quizzes: 25 points
- Project 3: 30 points



# Project 1 (30): Implement graphics algorithms in software

- Project 1A (due April 3): Install and run example program (2%)
- Project 1B (due April ~6): Rasterize screen-space triangles (7%)
- Project 1C (due April ~12): Rasterize colored triangles (2%)
- Project 1D (due April ~17): Add Gouraud shading (4%)
- Project 1E (due April ~24): Add world-space-to-camera-space transforms (8%)
- Project 1F (due April ~30): Add Phong shading (7%)

These projects build on one another.

If you do not do these projects in a timely way, you will likely not pass the class.



# Projects 2, 3, and quizzes

- Project G: 15%
- Project 2: 15% ← up in the air
  - Project 2A (due May ~7): Basic OpenGL program (8%)
  - Project 2B (due May ~14): Advanced OpenGL program (7%)
- Quizzes (~5): 25%
- Final Project Proposal (due May ~17): 0%
- Project 3: 30%
  - Student-defined projects due on Finals week (30%)
    - This class: Tues June 8, 8am





# Final Project Ideas

- Make a video game
- Make a screen saver
- Do some advanced graphics techniques
- Model Deschutes
- Etc...



# Grading

- Projects 1A-1F, 2A, and 2B will be submitted to Canvas and graded by me or the TA. In almost all circumstances, producing the correct picture will result in full credit.
  - Even slightly wrong? → Less than half credit
- The final projects will be demonstrated in front of the whole class on the day of the Final. Since we have such a large class, we may need to be creative on how to do this (i.e., multiple rounds with subsets of us, culminating in a finale during the Finals period).



# Grading

- Project 1: 30 points
- Project G: 15 points
- Project 2: 15 points
- Proposal for Project 3: 0 points
- Quizzes: 25 points
- Project 3: 30 points
- **→ 100 points**
  - denominator for 541: 115:
    - 541 students expected to get 30 on Project 3, participate in Project G



# Quizzes

- Planning for ~5 Quizzes
- Will not be pop quizzes – will be announced at least one lecture in advance



# Final Project

- Most final projects score 25/30
- Exceptional projects get 30/30



# In All...

- I am happy to give everyone A's
- But:
  - If you want to get an A, you should be aiming for perfect scores on 1A-1F, 2A, 2B
  - Since most people will lose points on the quizzes and final project



# Office Hours

- Only come to OH if you have started the project
- Sit and listen to others questions is not an option
- Surge OH this week to help with 1A, 1B
  - When?
- When should regular OH be?



# Academic Misconduct (1 of 2)

- The programming projects are individual efforts
  - You may discuss the projects with your classmates.
  - Do not let someone look at your code on your screen. (BUT: helper can look at helpee's code)
  - Absolutely, positively do not email code.
  - Do not search the internet for previous implementations (includes github)





# Academic Misconduct (2 of 2)

- If I detect collusion, all individuals involved will receive an F in the course immediately
  - I choose to not enumerate cases that involve collusion. Whiteboard conversations are fine. If appropriate, the helper can look at the helpee's code. If you feel you are in a gray area, then you should email me.
  - Please note that if you are the one providing too much help, then you will also get an F



# Important: Form on Canvas

☰ CIS 441/541 > Assignments

Spring 2021

Search for Assignment

Home

Assignments

Discussions

Grades

People

Syllabus

Conferences

Chat

Library Research

Help

Panopto Recordings

Zoom Meetings

▼ Upcoming Assignments



**Misconduct Form**

Due Apr 1 at 8:15am



**Project 1A**

Due Apr 4 at 6am | -/2 pts



# 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.



# Working Together, part 2

- ~~Piazza~~ (what technology to use?)
  - I encourage you all to monitor (and respond) to the forum.
  - 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  $1/3^{\text{rd}}$  grade (i.e., B to B+).



# 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 are very likely to receive full credit.
- For Project 2, your evaluation is to be determined
- 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 during Week 7.



# Late Passes

- You have 2 "late passes."
- Late passes allow you to turn in your project two days after the due date for full credit.
- If you run out of late passes, then you may continue to earn half credit on any project.
- Every unused late pass is worth 0% extra credit.
- Don't need to tell me if you want to use it
  - Late pass assignment will figured out the last week
  - All late projects will be scored at 50% and then credit from late passes will be awarded at the end of the term



# 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**





# Project #1A

- Goal: write a specific image
- Due: 11:59pm, April 3<sup>rd</sup> ... in 86 hours (!)
- % of grade: 2%
- Q: Why do I only get 4 1/2 days to complete this project?
- A: We need to need to get multi-platform issues shaken out ASAP.
- May be a little painful
- And: 1B is coming soon after...

CIS 441/541: Project #1A Due 11:59pm April 3<sup>rd</sup>, 2021  
(which means 6am April 4<sup>th</sup>, 2021)

Worth 2% of your grade

Setup:

- 1) Download and install CMake. I recommend you use at least version 3.16.
- 2) Download, build, and install VTK. I strongly recommend you use version 8.2.0.
- 3) Make a directory called "project1A"
- 4) Download file project1A.cxx and CMakeLists.txt from class website and copy them into directory project1A.
- 5) Update the CMakeLists.txt file to point at the correct location for your VTK.
- 6) Run CMake. This will create build files.
- 7) Compile the project1A program. For Unix/Mac, this means "make"
- 8) Run the program.
- 9) It should output an image that is 1024x1024 called proj1A.png. The first pixel of the file should be red (although that might be hard to eyeball, especially if the corners of your window are curved, like with Macs).

Assignment:

- 1) You are to make an image that is 1024x1350. (1024 in width, 1350 in height)
  - a. The image will be broken into 27 horizontal strips, with each strip consisting of 50 rows of pixels
- 2) The color for the X<sup>th</sup> strip should be:
  - a.  $X \% 3 = 0 \rightarrow B=0$
  - b.  $X \% 3 = 1 \rightarrow B=127$
  - c.  $X \% 3 = 2 \rightarrow B=255$
  - d.  $(X/3) \% 3 = 0 \rightarrow G = 0$
  - e.  $(X/3) \% 3 = 1 \rightarrow G=127$
  - f.  $(X/3) \% 3 = 2 \rightarrow G=255$
  - g.  $X/9 = 0 \rightarrow R=0$
  - h.  $X/9 = 1 \rightarrow R=127$
  - i.  $X/9 = 2 \rightarrow R=255$
- 3) Examples
  - a. The first strip (which is at the beginning of the image buffer and at the bottom of the image) is to be black.  $R=0, G=0, B=0$
  - b. The strip immediately above that should be dark blue,  $R=0, G=0, B=127$
  - c. Above that should be bright blue  $R=0, G=0, B=255$
  - d. Above that should be dark green,  $R=0, G=127, B=0$

The correct answer is located on the class website. There is also an image differencer program on the class website. You can use that to verify that your image is correct. You should do this for this assignment.

If your program produces the wrong output, you can receive no more half credit. (It is critical you check your work with the differencer).

What to submit: your source code (project1A.cxx)



# Project #1A in a nutshell

- Assignment:
  - Install CMake
  - Install VTK – Visualization ToolKit
  - Modify template program to output specific image



# What is *CMake* ?

- ❑ Cmake is a cross-platform, open-source build system.
- ❑ CMake is a family of tools designed to build, test and package software.
- ❑ CMake is used to control the software compilation process using simple platform and compiler independent configuration files.
- ❑ 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](http://www.cmake.org) & follow the directions



About ▾ Resources ▾ Developer Resources ▾ Download 🔍



Build with CMake. Build with Confidence.

Learn How

CMake is an open-source, cross-platform family of tools designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files, and generate native makefiles and workspaces that can be used in the compiler environment of your choice. The suite of CMake tools were created by Kitware in response to the need for a powerful, cross-platform build environment for open-source projects such as ITK and VTK.

CMake is part of Kitware's collection of commercially supported [open-source platforms](#) for software development.

# What is the



# Visualization Toolkit

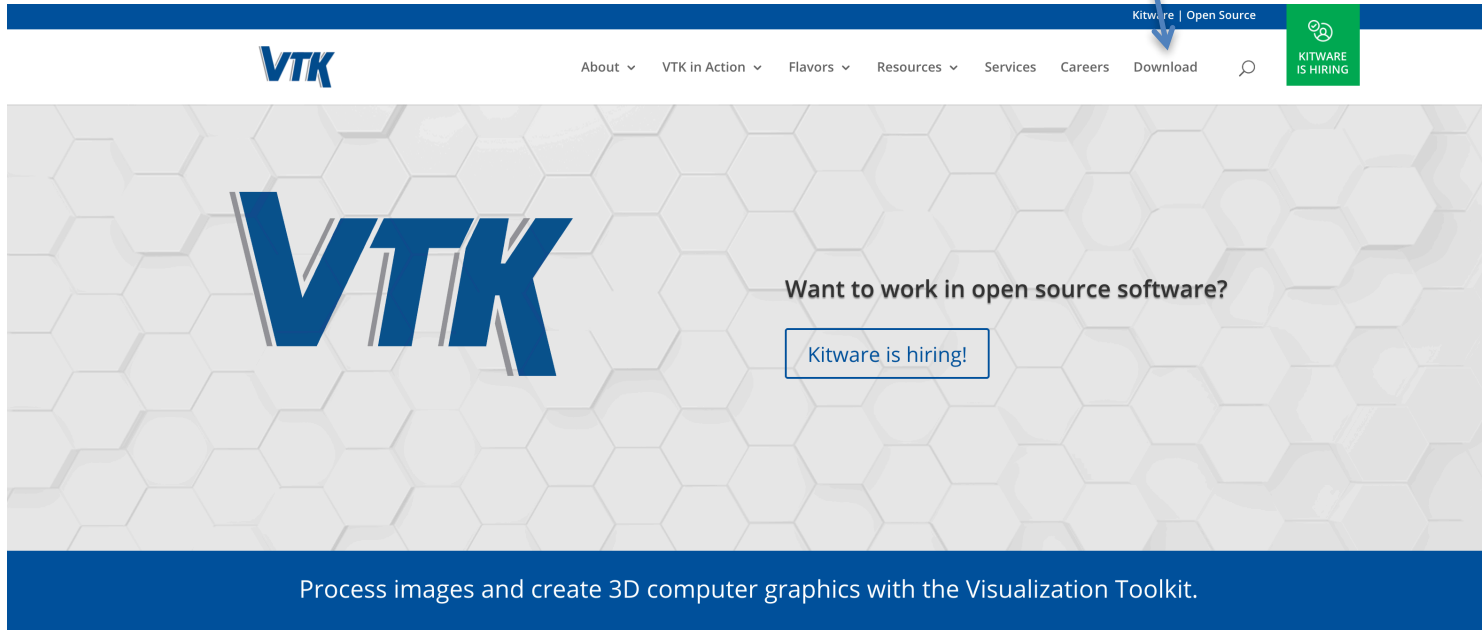
# ?

- 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.



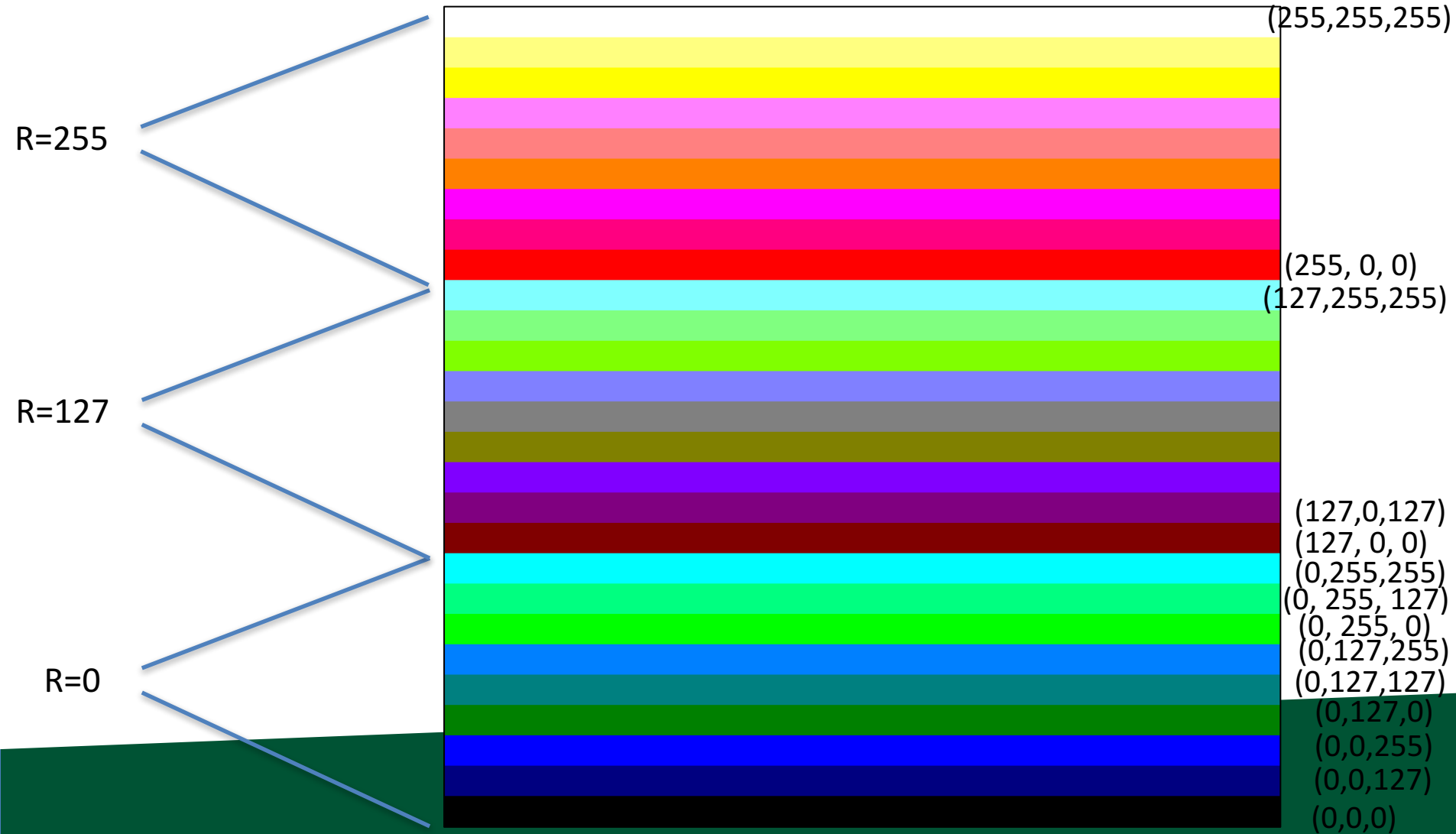
# How do you install VTK?

- Go to [www.vtk.org](http://www.vtk.org) , go to Download and follow the directions





# What is the image I'm supposed to make?





# 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
- Run differencer!
  - Less than half credit if it does not pass differencer
- Upload project1A.cxx to Canvas





# 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



# Project #1A: background

- Colors are 0- $\rightarrow$ 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?
  - $\rightarrow$  No. Human eye can distinguish  $\sim$ 10M colors.
  - 8bits \* 3 colors = 24 bits =  $\sim$ 16M colors.
- Red = (255,0,0)
- Green = (0,255,0)
- Blue = (0,0,255)



# Project #1A: background

- An “M by N” 8 bit image consists of  $M \times N \times 3$  bytes.
  - It is stored as:  
 $P_0/R, P_0/G, P_0/B, P_1/R, P_1/G, P_1/B, \dots, P_{(M \times N)}/R, P_{(M \times N)}/G, P_{(M \times N)}/B$
- $P_0$  is the bottom, left pixel
- $P_{(M-1)}$  is the bottom, right pixel
- $P_{(M \times (N-1))}$  is the top, left pixel
- $P_{(M \times N-1)}$  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



# What should you do if you run into trouble?

- 1) Start with Piazza (?)
- 2) Surge OH this week

Don't forget: these slides are available online  
Lectures are recorded and posted to Canvas