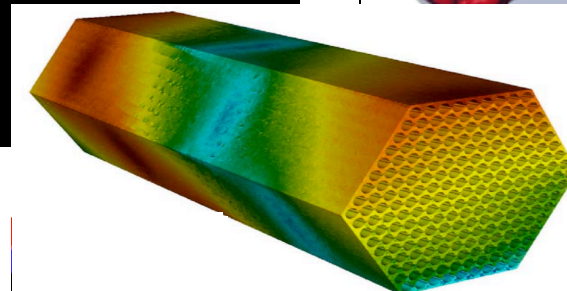
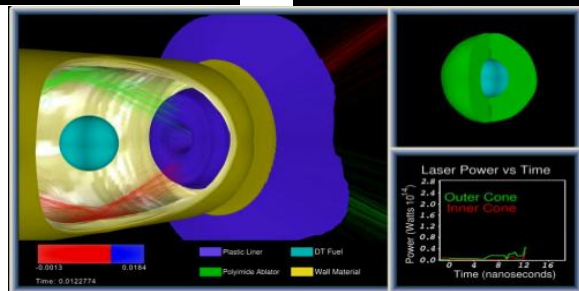
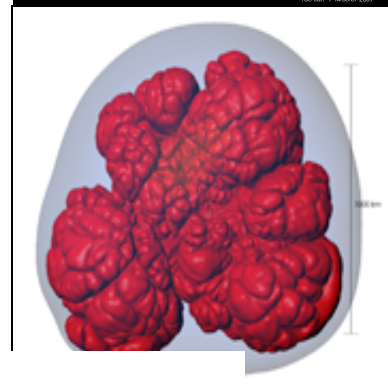
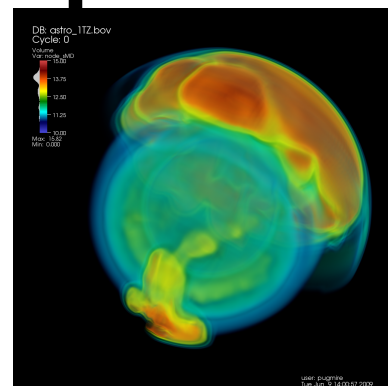
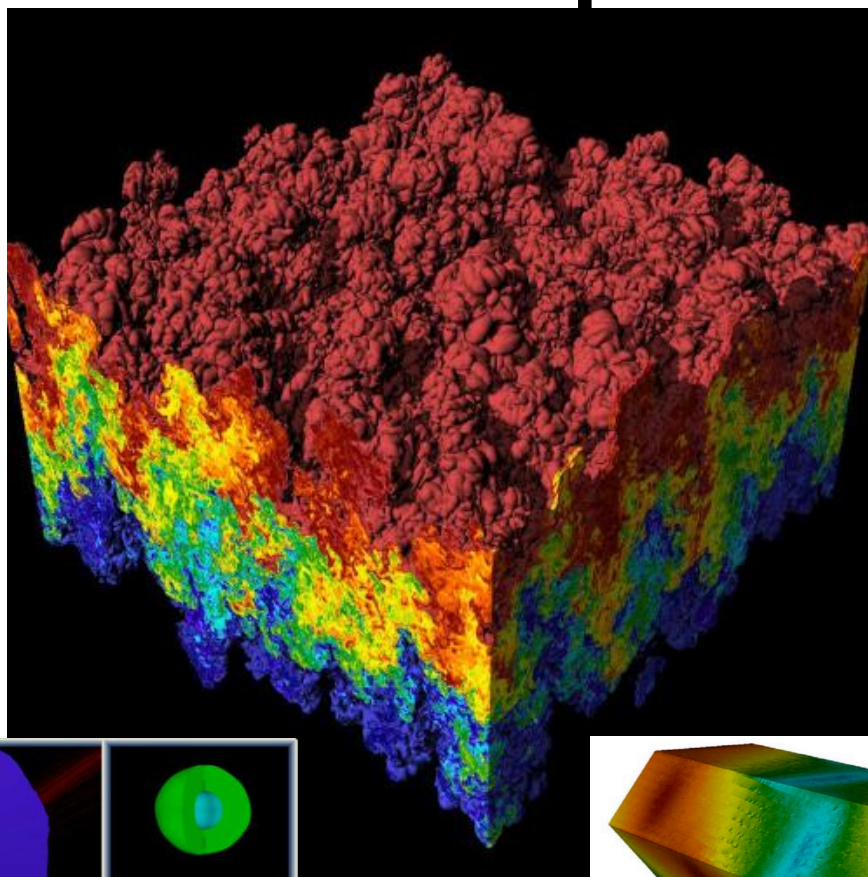
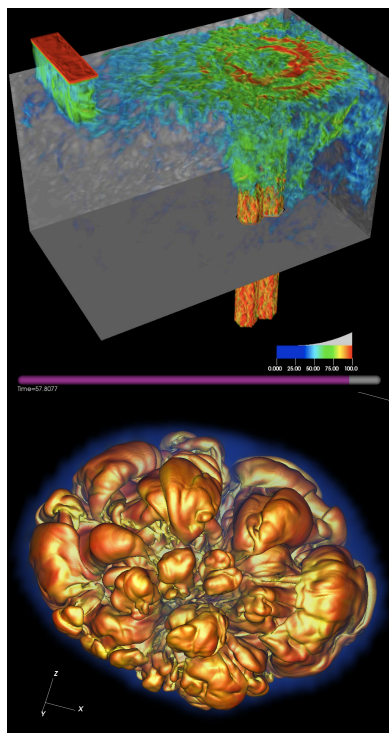




Introduction to Computer Graphics



Hi, I'm Hank...

- Associate Professor of CIS at UO
- Arrived at UO March 2013
- Research interests include visualization, high-performance computing, computer graphics
- Director of CDUX Research Group (10 PhD, 0 MS, 3 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 have spent 15 years programming,
almost exclusively using C, C++, and Unix



Trying to Add the Class?

- Let's talk after class...



Outline

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



Outline

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

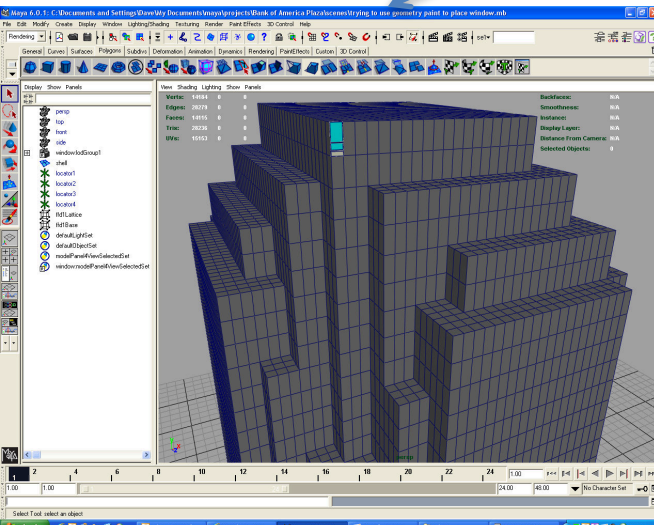
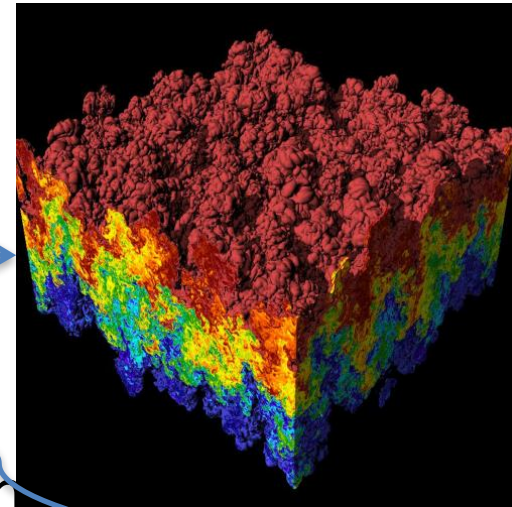


Computer Graphics

- 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

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://www.cs.uoregon.edu/Classes/19W/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”



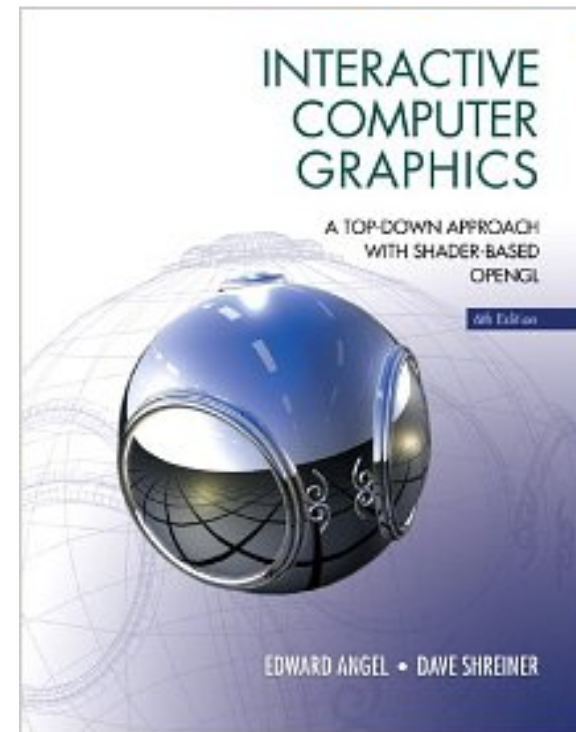
This Class In a Nutshell...

- 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: present final project. (Midterm a few weeks earlier)

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

Course Materials

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



Project 1 (30): Implement graphics algorithms in Software

- Project 1A (due Jan 12th): Install and run example program (2%)
- Project 1B (due Jan ~15): Rasterize screen-space triangles (7%)
- Project 1C (due Jan ~21): Rasterize colored triangles (2%)
- Project 1D (due Jan ~26): Add Gouraud shading (4%)
- Project 1E (due Feb ~2): Add Phong shading (7%)
- Project 1F (due Feb ~8): Add world-space-to-camera-space transforms (8%)

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 midterm

- Project G: 15%
- Project 2: 15%
 - Project 2A (due Feb ~15th): Basic OpenGL program (8%)
 - Project 2B (due Feb ~22nd): Advanced OpenGL program (7%)
- Midterm (Feb ~26): 25%
- Final Project Proposal (due Feb ~26): 0%
- Project 3: 30%
 - Student-defined projects due on Finals week (30%)
 - This class: Monday March 18th, 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
- Midterm: 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



Midterm

- 25 points
- NOT: A = 22.5, B = 20, C = 17.5, etc...
- INSTEAD:
 - A students typically score >20 points
 - B students typically score >15 points
 - ...



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 midterm and final project
- 20/25 on midterm, 25/30 on final project, and perfect scores on 1A-1F, 2A, 2B will earn you an A.



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
 - Thursday, Friday
- 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



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
 - 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 1/3rd 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 be a question on the final
 - All late projects will be scored at 50% and then credit from late passes will be awarded at the end of the term



Planned Absences

2019 Exascale Computing Project Annual Meeting

HELD AT
Royal Sonesta Houston Galleria
Houston, Texas

January 14-17, 2019 – ECP Annual Meeting

JANUARY 14-18, 2019 - INDIVIDUAL OR GROUP MEETINGS

06

Days

08

Hours

33

Mins

35

Secs

[REGISTER NOW](#)

Workshop on In Situ Data Management



Sponsored by the U.S. Department of Energy,
Office of Advanced Scientific Computing
Research (ASCR)
North Bethesda, MD
January 28 – 29, 2019



[News »](#) [Committee](#) [Call for Proposals »](#) [Coming Seminars »](#) [Reports](#) [FAQ »](#)

Call for Proposals

We are constantly accepting proposals.

Next due date is December 15th, 2017.

The seminars we are calling for, are those to be held between 8 months ahead and 2 years ahead of the next submission due date.

(Example: In the case that the next due date is December 15th, 2017, we are calling for the seminar to be held from August...

[Read More](#)





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, Jan 12th ... in 84 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 Jan 12th, 2019
(which means 6am Jan 13th, 2019)

Worth 2% of your grade

Setup:

- 1) Download and install CMake. I recommend you use version 3.0 or later.
- 2) Download, build, and install VTK. I strongly recommend you use version 8.1.2.
- 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 oneRedPixel.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.
 - a. The image will be broken into 27 horizontal strips, with each strip consisting of 50 rows of pixels
- 2) The color for the Xth strip should be:
 - a. $X \% 3 = 0 \rightarrow B=0$
 - b. $X \% 3 = 1 \rightarrow B=128$
 - c. $X \% 3 = 2 \rightarrow B=255$
 - d. $(X/3) \% 3 = 0 \rightarrow G = 0$
 - e. $(X/3) \% 3 = 1 \rightarrow G=128$
 - f. $(X/3) \% 3 = 2 \rightarrow G=255$
 - g. $X/9 = 0 \rightarrow R=0$
 - h. $X/9 = 1 \rightarrow R=128$
 - 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, $R=0, G=0, B=128$
 - c. Above that should be bright blue $R=0, G=0, B=255$
 - d. Above that should be dark green, $R=0, G=128, B=0$

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: 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->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.
 - 8bits * 3 colors = 24 bits = ~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
- $P((M \times N) - M + 1)$ is the top, left pixel
- $P(M \times N)$ 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 *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 & follow the directions



The screenshot shows the CMake website homepage. At the top left is the CMake logo, a 3D triangle with red, blue, and green faces. To its right is the word "CMake" in a large, bold, sans-serif font. In the top right corner, there is a "Kitware" logo and a search bar. Below the header, there are navigation links: "PROJECT", "RESOURCES", "HELP", and "OPEN SOURCE". The main content area on the left contains a welcome message: "Welcome to **CMake**, the 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." Below this is a "News" section with a list of recent releases and a "More News >" link. On the right side, there is a large banner for "CMake 2.8.10 Available for Download" featuring the CMake logo and a "Download Now >" link.

CMake

Welcome to **CMake**, the 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.

News [More News >](#)

- 11.07.2012** CMake 2.8.10 Just Released
- 08.09.2012** CMake 2.8.9 is Now Available!
- 07.18.2012** Kitware Announces New Fall Courses
- 04.19.2012** CMake 2.8.8 is Now Available
- 03.02.2012** CDash 2.0.2 Now Available

CMake 2.8.10 Available for Download

[Download Now >](#)

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 , go to Resources->Download and follow the directions



The screenshot shows the VTK website homepage. At the top left is the VTK logo with the text "Visualization Toolkit". To the right is the "Kitware" logo and a search bar. Below the logo is a navigation bar with links: "PROJECT", "RESOURCES", "HELP", and "OPEN SOURCE". A blue arrow points from the text "go to Resources->Download" in the list above to the "RESOURCES" link. Below the navigation bar is a large text block describing VTK as an open-source software system for 3D computer graphics, image processing, and visualization. It mentions that VTK consists of a C++ class library and several interpreted interface layers including Tcl/Tk, Java, and Python. It also mentions that Kitware offers professional support and consulting services. To the right of this text is a box titled "Kitware receives HPCwire's Editors' Choice Award for VTK." with a "Learn More >" link. Below the main text block is a "News" section with a "More News >" link. The news section lists several updates with dates and titles.

Kitware Search
Tell us what you think

Visualization Toolkit

PROJECT RESOURCES HELP OPEN SOURCE

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. **Kitware**, whose team created and continues to extend the toolkit, offers [professional support and consulting](#) services for VTK. VTK supports a wide variety of visualization algorithms including: scalar, vector, tensor, texture, and volumetric methods; and advanced modeling techniques such as: implicit modeling, polygon reduction, mesh smoothing, cutting, contouring, and Delaunay triangulation. VTK has an extensive information visualization framework, has a suite of 3D interaction widgets, supports parallel processing, and integrates with various databases on GUI toolkits such as Qt and Tk. VTK is cross-platform and runs on Linux, Windows, Mac and Unix platforms.

News [More News >](#)

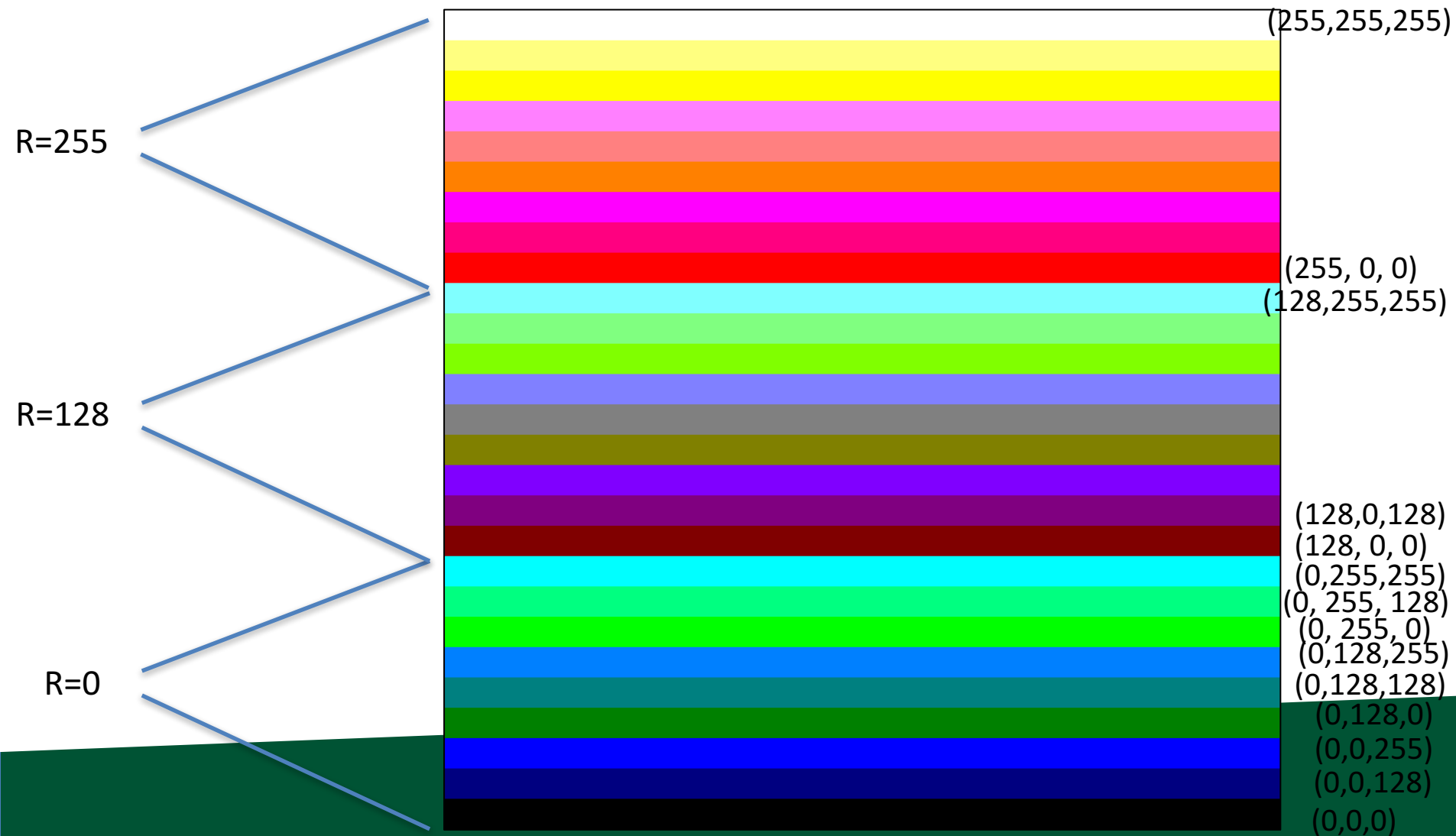
- 01.31.2013** Kitware Provides Mobile Visualization Support for the Visible Pat...
- 12.03.2012** ParaView 3.98.0 Now Available
- 11.27.2012** Kitware Receives DARPA Funding to Develop a Visualization Design ...
- 07.18.2012** Kitware Announces New Fall Courses
- 05.16.2012** VTK 5.10 Now Available

Kitware receives HPCwire's Editors' Choice Award for VTK.

[Learn More >](#)

HPC wire
Editors' Choice Awards 2011

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



What should you do if you run into trouble?

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

Don't forget: this lecture is available online