Your Instructor Team

- Hank Childs
- Associate Professor of Computer and Information Science
- Research interests include visualization, high-performance computing, computer graphics

- Roba Binyahib
- 6th year PhD student
- Research interests include visualization and high-performance computing
Outline

• Class Overview (Syllabus)
• Overview of Visualization: Scientific Visualization versus Information Visualization
• Project 1 Overview
Outline

• Class Overview (Syllabus)

• Overview of Visualization:
  Scientific Visualization versus Information Visualization

• Project 1 Overview
Syllabus is Online

• http://ix.cs.uoregon.edu/~hank/410

CIS 410/510: INTRODUCTION TO SCIENTIFIC VISUALIZATION

Lecture Time: Tues/Thurs 10:00-11:20
Lecture Location: 166 Lawrence Hall

Instructor: Hank Childs
Teaching Assistant: Roba Binyahib

Office Hour Schedule:
• Hank: Tuesday, 230-330 (301 Deschutes)
• Roba: TBD, (232 Deschutes)

- Syllabus: objectives, expectations, grading, academic misconduct, late policy
- Projects
- Lectures
- Course materials: The textbook is The Visualization ToolKit. The textbook is helpful, but most students find they do not need it to be successful at the class.

Contact Information

Email: hank at uoregon.edu
Phone: (541) 346-3414 (terrible way to get in touch with me!)
Fax: (541) 346-5373

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

SYLLABUS FOR WINTER 2020 CIS 410/510

Objectives

The high-level objectives of the course are (1) to develop understanding of scientific visualization, including understanding of scientific data, scalar visualization techniques, and vector visualization techniques and (2) to develop skills in translating this understanding to functioning software.

Expectations

This is a projects-driven class. The projects will help you learn the theory behind scientific visualization, 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.
Prerequisites For This Course

• CIS 330 is the only prerequisite
  – Waived for anyone who has good C++ knowledge
• CIS 441 is not a prerequisite
Grading (410)

• Introductory Projects: 45 points
• Advanced Project: 35 points
• Quizzes: 20 points
• Grades: 90%: A-, 80%: B-, etc.
Grading (510)

- Introductory Projects: 45 + 10 points
- Advanced Project: 35 + 30 points
- Quizzes: 30 points
- Presentation: 10 points
- Grades: 90%: A-, 80%: B-, etc.
(Planned) 410 Introductory Projects

• Install and run example program
• Field interpolation
• Coloring
• Advection and streamlines
• Isolines
• Isosurfaces
• Using a visualization toolkit
(Planned) 510 Introductory Projects

• All 410 projects + 2 more
  – Advections and streamlines, part 2
  – Using a visualization toolkit, part 2
Advanced Projects

• Two tracks:
  – Custom project
  – Pre-defined project

• 410: you choose

• 510: do both
Pre-defined Project

- Ray-casted volume rendering
- 35% of grade
- There will be a grading rubric that outlines different levels of complexity
- → most students will only do 30/35
Custom Project

• You design and implement a project involving visualization

• Milestones:
  – Proposal: ~February 20th (3% of grade)
  – Implementation: March 16th @ 8am (27% of grade)
  – Presentation: March 16th @ 8am (5% of grade)
Custom Project Presentations

• All students attend project presentations for custom projects
  – (even if you do pre-defined project)
• -4% of grade if missed
• Have a good excuse for missing?
  – Tell me by this Friday, Jan 10
  – Otherwise you will lose 4%.
    • (unless extenuating circumstances)
Language Used For This Class

- The introductory projects will be implemented in C++.
  - Each project will start with a skeleton
- The advanced project can be implemented in whatever language you like.
Quizzes

• Quiz #1 on advection: 10 pts (410) / 15 (510)
• Quiz #2 on isosurfacing: 10 pts (410) / 15 (510)
Expectations

• This is a projects-driven class.
• The projects will help you learn the theory and techniques behind scientific visualization, 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 for those who have not done significant programming before.

• 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
• Feel free to call me Hank

• Please always bring paper and pencil/pen ... we will do exercises during class
This Class In a Nutshell...

• 1\textsuperscript{st} Half: Learn the basics behind scientific visualization
• 2\textsuperscript{nd} Half: Advanced project

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

• Textbook: The Visualization ToolKit ... but most don't get it

• PowerPoint lectures will be posted online.

• Some lectures may be complemented with chalkboard exercises.
What is VTK?

- VTK: Visualization ToolKit
- Most popular visualization library in the world
- We will use this for our introductory projects
- Cross-platform, so you can use whatever development platform you please.
Office Hours

• Hank: Tuesday 230-330 (NOT TODAY)
• Roba: Monday 1-2:30, Thursday 12-1:30
• Will be extra OH this week
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.
- Absolutely, positively do not email code.
- Do not search the internet for previous implementations.
• If I detect collusion, all individuals involved will receive an F in the course immediately
  – I choose to not enumerate cases that involve collusion. Having a conversation without showing code is as far as you should go. Whiteboard conversations are fine. 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
• I will be looking at your source code
Piazza

- I have will set up a forum on Piazza (today)
- I will monitor (and respond) to the forum and encourage you all to do the same.
Evaluation Criteria

• I will describe evaluation criteria on a project-by-project basis.
• For the most part,
  – I give you:
    • the right answer
    • a differencing program
  – And ask that you send me:
    • a screenshot of the differencing program showing you have no differences with the right answer
    • your source code
Evaluation Criteria, pt 2

• If you hand in something that produces the wrong answer, then assume you will get <50% on it.
  – I would rather you turn in something that works correctly late than on time and incorrect
Late Passes

• You have 2 "late passes."
• Late passes allow you to turn in your project after the due date for full credit.
  – One late pass gives you an extra two days.
  – Example: due Friday, use a late pass, now due Sunday
  – Can combine 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 0% extra credit.
Class Summary

- This class will teach you the theory and techniques behind scientific visualization
- This class will improve your programming skills
- This class may help you land a job
- This class will require a lot of work
Grading for 410
(repeat of previous slide)

• Introductory Projects: 45%
• Advanced Project: 35%
• Quizzes: 20%
Lecture hiccups

• Unfortunate: travel week of Feb 3rd
• Plan: to be determined
  – Possibly: Roba lecture
  – Possibly: YouTube lecture
Lecture Plan

• First six weeks are mapped out.
• Last four weeks can be adaptive based on your project plans.
Outline

• Class Overview (Syllabus)
• Overview of Visualization: Scientific Visualization versus Information Visualization
• The Very Basics of Computer Graphics
• Project 1 Overview
Scientific Visualization

• An interdisciplinary branch of science
  – primarily concerned with the visualization of three-dimensional phenomena (architectural, meteorological, medical, biological, etc.)
  – the emphasis is on realistic renderings of volumes, surfaces, illumination sources, and so forth, perhaps with a dynamic (time) component.

• It is also considered a branch of computer science that is a subset of computer graphics.

• The purpose of scientific visualization is to graphically illustrate scientific data to enable scientists to understand, illustrate, and glean insight from their data.

Source: wikipedia
Information Visualization

- The study of (interactive) visual representations of abstract data to reinforce human cognition.
  - The abstract data include both numerical and non-numerical data, such as text and geographic information.

Source: wikipedia
SciVis vs InfoVis

• “it’s infovis when the spatial representation is chosen, and it’s scivis when the spatial representation is given”
What sorts of data?

Of course, lots of other data too...
What Is Visualization Used For?

• 3 Main Use Cases:
  – Communication
  – Confirmation
  – Exploration
How Visualization Works

• Many visual metaphors for representing data
  – How to choose the right tool from the toolbox?
• This course:
  – Describe the tools
  – Describe the systems that support the tools
This Class

• Focus is on SciVis
• We can add some InfoVis in the second half to fit advanced projects.
Outline

• Class Overview (Syllabus)
• Overview of Visualization:
  Scientific Visualization versus Information Visualization
• Project 1 Overview
Project #1

• Goal: write a specific image
• Due: “Friday Jan 10th” → “6am Saturday Jan 11th”
• % of grade: 2%
• Goal: get multi-platform issues shaken out ASAP.
• Experience last year was pretty good.

Worth 2% of your grade

Assignment:
1) Download, build, and install VTK.
2) Download and install CMake. Use version 3.9
3) Download the file called data.vtk
4) Make directory called “project1”
5) Download file project1.cxx and CMakeLists.txt from class website and copy them into directory project1
6) Update the VTK_DIR variable in CMakeLists.txt to point to the path of the VTK you just installed.
7) Run CMake. This will create build files.
8) Compile the program. For Unix/Mac, this means “make”
9) Run the program. (How to run is platform dependent … on Linux and Mac, a binary gets generated and you invoke it.)
10) Submit a screenshot of the working program via Canvas
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
What is the Visualization Toolkit (VTK)?

• 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
How do you install VTK, part 2?

• See if your favorite package manager has a version already.

• Note: consult project 1 prompt for version of VTK to use
What do I do again?

• Install CMake & VTK.
• Download
  – “project1.cxx” from class website
  – “CMakeLists.txt” from class website
  – “data.vtk” from class website
• Run CMake
• Compile project1.cxx
• Run program
• Upload a screenshot of the result to Canvas by 6am Sat. morning.
What should you do if you run into trouble?

1) Start with Piazza
2) Ask questions at Thurs class
3) Attend OH (additional times will be announced)
Don’t forget

• This lecture is available online
  – [http://ix.cs.uoregon.edu/~hank/410](http://ix.cs.uoregon.edu/~hank/410)

• All project prompts are (will be) available online