

Introduction to CS200/CSD2100/CSD2101

Prasanna Ghali

Plan

- Objectives
- Learning Outcomes
- Assumed Knowledge
- Teaching Rationale
- Teaching Strategies
- Course Overview
- Course Administration

What is Computer Graphics?

- All aspects of creation, manipulation, and display of images using a computer
- See [this](#) wiki page for comprehensive information on history and applications of computer graphics

Objectives

- Teach fundamental 2D and 3D mathematical elements and algorithms underlying computer graphics
- Introduce industry standard graphics API to provide experience in developing interactive graphics applications

Learning Outcomes

- Have low-level, under-the-hood understanding of mathematical elements and algorithms of 2D and 3D computer graphics
- Have foundation to explore and use ever-evolving graphics APIs and tools:
 - [OpenGL](#), [OpenGL ES](#), [WebGL](#), [Vulkan](#), [OpenCL](#), [Direct3D](#), [Metal](#), [RenderMan](#), ...
- Apply knowledge of computer graphics in game projects
- Be confident and excited about learning more advanced material in CSD2150, CSD2170, CSD3150, ...

Assumed Knowledge

- Formal prerequisites are MAT140/CSD1240/CSD1241 and CS170/CSD1170/CSD1171
- Must have working knowledge of:
 - Introductory linear algebra including vectors and matrices from CSD1241 [[here's](#) an excellent review of these topics]
 - Programming in both C and C++

Teaching Rationale (1/3)

- Computer graphics is fun
 - Can make cool games
 - Create amazing special effects in movies
 - Visualize data from CT scans
 - Scientific visualization
 - Computer aided design
 - Myriad other applications ...
- But ...

Teaching Rationale (2/3)

- Computer graphics is hard
 - Gazillions of photons originate from light sources, get bounced around objects and end up on cones in our retinas
 - Ginormous computing power required to perform real-time simulation of these photons and their interactions doesn't (yet) exist
 - Instead, we've to come up with mathematical approximations and clever algorithms

Teaching Rationale (3/3)

- This means that you're going to have to work hard to understand the math and algorithms
- But then you get reward of making pretty pictures
- Which is even more fun because you earned it!!!

Questions ...

- If you're lost, ask questions ...
- If you're getting bored, ask questions ...
- If you'd like to ask questions, ask questions ...

Teaching Strategies

- Lectures
- Tutorials
- Quizzes
- Assignments
- 3D Project
- Comprehensive midterm and final tests

Teaching Strategies: Lectures

- Theoretical concepts are introduced
- Almost always accompanied by derivation of mathematical elements and algorithms
- Will also include exercises, demonstrations, and live coding examples
- 1st half of semester introduces modern OpenGL toolbox with emphasis on 2D
- 2nd half of semester introduces 3D
- Expect pop quizzes on Moodle ...

Teaching Strategies: Quizzes

- Aim is to clarify material covered in lectures
 - We'll focus on a few problems and work through them in detail
- One quiz per week

Teaching Strategies: Assignments

- Integral and important part of course
- Will allow you to apply math and theory discussed in class
 - Class lectures explain theory in terms of using correct, efficient math and algorithms
 - You'll then implement computer graphics by translating theory into code
- We'll exclusively use modern OpenGL framework

Teaching Strategies: Assignments

- Assignments are *not* group projects
 - Must represent own individual work
 - Reasonable to discuss general solutions and strategies
 - *Not* reasonable to give and/or take code
 - *Not* reasonable to copy code from web

Teaching Strategies: Assessments

- Worksheets/Quizzes: 15%
- Programming assignments: 35%
- 3D Project: 10%
- Midterm: 20%
- Final: 20%

Submissions: Quizzes

- You'll be assigned one quiz/worksheet per week
- You'll use Moodle to provide answers
- After deadline, I'll provide my [correct] solutions
- Absolutely no late submissions ...

Submissions: Programming Assignments

- You'll be assigned one assignment per week during 1st half of course
- Each assignment builds on previous submission
- Computer graphics is hard!!! Therefore, timeliness and discipline are required in getting work done
- Submissions graded once per week [at end of week]

Submissions: Programming Assignments

- Penalty for late submissions:

Submission	Penalty	How to Submit?
Within deadline	None	Moodle
Within one week after deadline	Maximum C grade	Moodle [Notice of late submission through email is required]
Between one week and two weeks after deadline	Maximum D grade	Email only [Notice of late submission through email is required]
After two weeks	Zero grade	N/A

Submissions: Contingencies

- Extensions for medical leave
 - Must always be supported by appropriate document(s)
 - Add days specified in medical certificate to submission deadline
- Family emergencies: speak to me

Course Overview (1/2)

- Overview of CSD2101
- Introduction to 3D Graphics Pipe
 - Graphics hardware
 - Fixed and programmable pipe
 - Coordinate systems
- Introduction to OpenGL toolbox
 - Evolution of OpenGL
 - The OpenGL toolbox
 - GLSL

Course Overview (2/2)

- Coordinate systems in graphics pipes: model, object, world, view, clip, NDC, device
- Affine transforms
- Intro to texture mapping
- Intro to 3D graphics pipe
 - 3D wireframe rendering using Bresenham algorithm
 - 3D solid rendering using Barycentric interpolation
- 3D transforms: linear, translation, view, perspective
- Hidden surface removal using depth buffer algorithm
- View frustum culling and clipping
- Lighting and shading: Phong illumination model; flat, Gourard, and Phong shading

Course Management

- Email: pghali@digipen.edu [Specify cs200 or csd2100 or csd2101 in email subject field]
- distance3.digipen.edu is online repository for ALL information including syllabus, class notes, assignment specs, class announcements, ...
- Post questions regarding material covered in class or an assignment to **Teams** page for this course