Course Introduction

CSE681: Introduction to 3D Image Generation

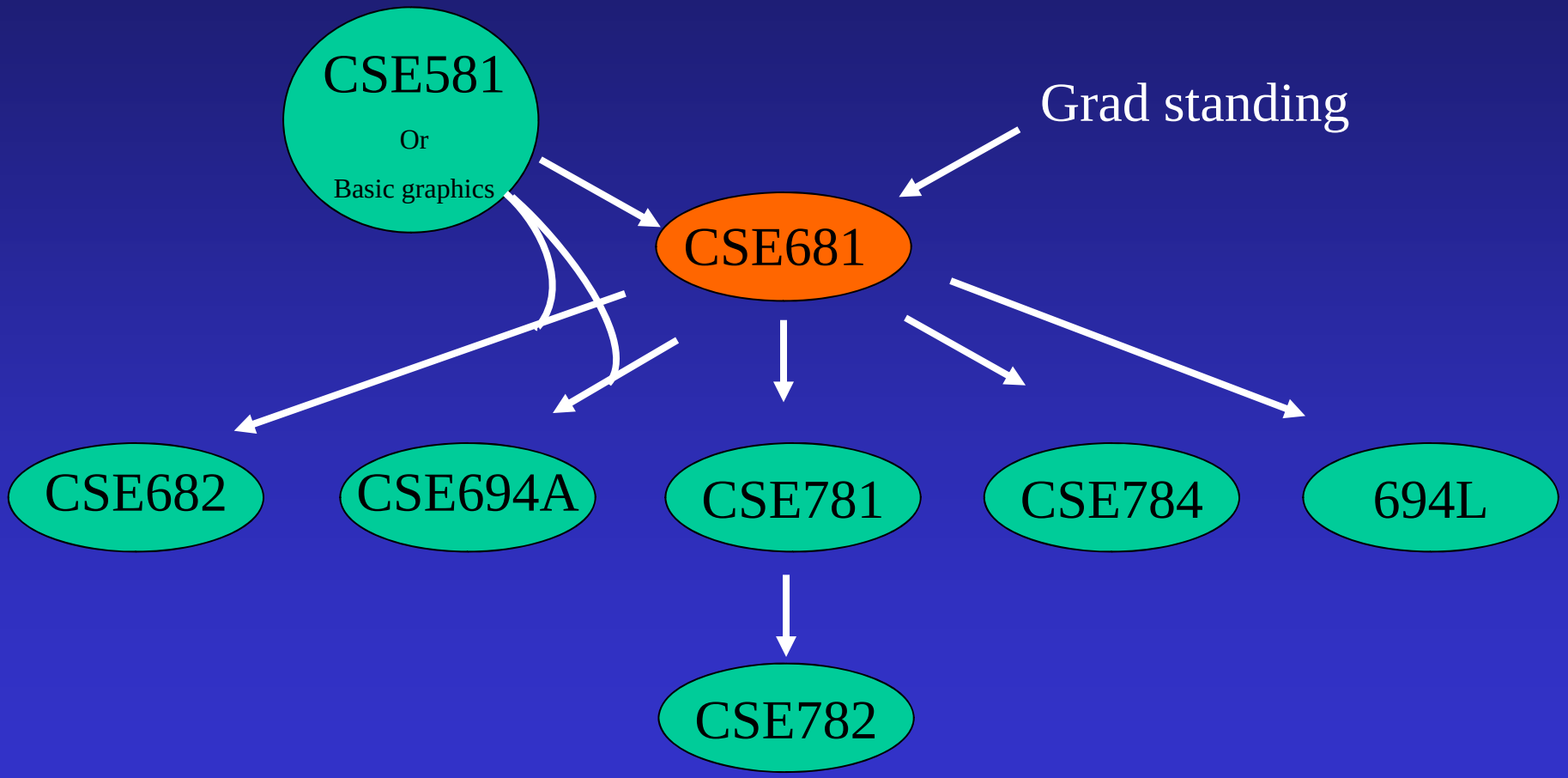
Rick Parent

Email: parent@cse.ohio-state.edu
www.cse.ohio-state.edu/~parent
www.cse.ohio-state.edu/~parent/classes/681/

Office: DL 787

Office Hours: T,F 1:30-2:30
(tentative)

Graphics Curriculum



CSE 681 Information

www.cse.ohio-state.edu/~parent/classes/681/index.html

Schedule
Labs
Announcements
Etc.

OSU Course Offerings Bulletin

Introduction to display hardware and applications, interactive techniques, 2D scan conversion, 2D and 3D transformations, clipping, 3D viewing, introduction to visible surface algorithms and illumination models.

Contents

- Ray Tracer
  - illumination modeling
  - texture mapping
  - object modeling.

- Entry course for graduate students

- Undergraduates should take CSE581 first

Prerequisites

- Basic Programming Skills (C++ or C)
- Basic sense of 2D and 3D geometry, coordinate systems
- Basic Matrix Math

Texts

Realistic Ray Tracing, by Peter Shirley

Optional Text and Additional Material taken from:
Introduction to Ray Tracing, by Andrew Glassner, Morgan-Kaufmann

Grading

- Labs: 50%
- Homeworks: 5%
- Midterm: 20%
- Final: 25%

Grading Policy

(www.cse.ohio-state.edu/~parent/generalInfo/gradingPolicy.html)

Grader grades quizzes and labs

Computing your grade - see web page

No curve, no rounding

If you need a certain grade – earn it!

Academic Misconduct

(www.cse.ohio-state.edu/~parent/generalInfo/acdmcMisconduct.html)

- Don't cheat.
- University's Academic Misconduct Committee
- Discussion of assignments OK; Do your own work.

Other Info

Class Directory: /usr/class/cse681/parent

Class Newsgroup: cse.course.cse681

Labs - tentative

1. Basic Ray Tracing - display a sphere
2. Scene description file, Illumination & Shadows
3. Refraction & Reflection
4. Anti-aliasing & Texture Mapping
5. Distributed ray tracing
6. Optimizatized rendering

Software

1. Default programming environment
   UNIX, gcc

2. Work out alternatives with grader

Programming Advice

1. Top Down Design
2. Think first, program later
3. Get something working, then add to it
4. Debugging graphics programs can be hard, Program accordingly

What to expect

1. I teach *algorithms*, not C or C++
2. If you don’t have the prereqs, and can’t keep up, then drop the course; if you do and can’t keep up, see me
3. Ask Questions - give me feedback
4. Use the newsgroup, email me

Topics

1. Review vectors, transformations
2. Ray tracing geometry and organizing ray tracer
3. Illumination
4. Shadows
5. Refraction & Reflection
6. Texture Mapping: solid & surface
7. Anti-aliasing
8. Speed-ups to ray tracing