CS 488 Computer Graphics 1
Introduction to Real-Time Rendering

Dr. Angus Forbes, Instructor
Kyle Almryde, TA

Creative Coding Research Group at EVL
Department of Computer Science
University of Illinois at Chicago

http://evl.uic.edu/creativecoding/cs488

Your instructor

- New faculty at UIC in the EVL
- Formerly an assistant professor at University of Arizona
- Founder and Director of the Creative Coding Lab
- Four Eyes Lab / Experimental Visualization Lab / Allosphere
- Graduate work in Computer Science and Media Arts & Technology at UC Santa Barbara
- Founder / Lead developer of Synaesthetic Software
- Worked at Micromuse, Inc.
- Research interests include data visualization, human-computer interaction, mobile computing
- Multi-media art installations involving interactive projection, mobile devices, fluid simulation, databases

Your teaching assistant

- New PhD student at UIC in the EVL
- Formerly researcher at University of Arizona
- Worked with Dr. Elena Plante in a Speech+Hearing lab at UA
- Set up MRI scans and helped analyze brain data
- Research interests include volume visualization and interactive scientific visualization

Overview of this course

1. Introduction to topics in computer graphics

2. Focus on programming real-time computer graphics

3. Project-based

What is Computer Graphics?

Graphics is an important topic for a number of applications:

- Entertainment
  - Films / Animation
  - Video Games
  - Motion Graphics / Advertising
- Design
  - Architecture / Industrial Design
  - Media Arts Installations
  - Advertising
- Research
  - Data Vis (Scientific Vis / Info Vis / Visual Analytics)
  - Virtual Reality
  - Physical Simulation

Computer Graphics Topics

Simulating:

- Modeling and visualizing real phenomena to better understand it
    - Visualizing weather, wind, water, gas (fluid dynamics)
    - Modeling how light interacts with objects

Creating:

- Making made-up things look realistic / believable
    - Humanity: hair, skin, expressions, crowds
    - Nature: fire, explosions, waves, snow, lightning, shadows, etc

Augmenting:

Making real-things looks unrealistic

- Image processing / Instagram filters
- Non-photorealistic rendering

Real-Time Programming

How fast does something have to be to be "real-time"?

"frames" are measured equivalently in either Hertz (Hz) or "frames per second" (fps).

(I.e. 20 Hz = 20 frames per second, meaning that each frame lasts for 50 milliseconds:1000ms/20hz = 50ms)

- Non-interactive films are usually 24 fps
- Average computer screen refreshes at 60 hz
- VR apps usually at 120 hz (60 fps per eye)
- Video games feel fast when moving at or close to 60fps
- Interactive applications usually are noticeably slow < ~30fps

Real-Time Programming

Goals of Real-time programming:

- To find **novel** ways to present interesting visual information

- To find increasingly **faster** ways of presenting visual information

Real-Time Programming

Being able to present a large number of pixels quickly is such an a ubiquitous issues that it has led to the development of specialized hardware that is focused entirely on this activity.

This hardware is called, alternatively, a “video card”, or a “graphics card”, or a “graphics programming unit”; often abbreviated to “GPU”.

Based largely on the needs of the video game industry, the animation industry, and the special effects industry, GPUs are becoming increasingly **powerful** (they can push more pixels) and increasingly **flexible** (you can control what happens to the pixels before they reach the screen).

Real-time 3D graphics

A major focus of this class will be about understanding how to **project** an interactive 3D model of a virtual environment onto a flat 2D screen. Next week we will start diving into what's called "the rendering pipeline"

- Mathematics of computer graphics
  - Linear algebra, matrix and vector operations, linear/affine transformations, computational geometry

- Programming computer graphics
  - OpenGL (library containing functions to send information to GPU)
  - GLSL (programming language that is compiled on the GPU)

CPU / GPU

CPU programming

1. define data needed to represent the **model** (geometry, movement, position of lights, camera position) of the **scene**
2. send this data to the special graphics card ...

GPU programming

1. **project** the geometry into 2D
2. **shade** the geometry (colors, lighting, materials)
3. draw pixels on the screen

Intro to Coding on Thursday

- Introduction to OpenGL

- Emphasis on GLSL Shader programming

- Using an cross-platform wrapper called Aluminum for teaching purposes

Assignment #1

1. Find one exciting example of each of the following uses of computer graphics:
   - Film
   - Gaming
   - Research
   - Interaction

You will get points only if your answer is **recent** and **original**. That is, a) the example is recent (less than two years old) and b) no one else has chosen the same example. Include an image that represents it and explain why its exciting or interesting. (40 pts.)

Assignment #1

2. Describe your interest in computer graphics topics. Why do you want to take this class? (10 pts.)

3. List your experience with programming computer graphics. (10 pts.)

4. Tell us what your preferred platform is. Find out what graphics card (or integrated chip) your main computer has and the highest version of OpenGL it supports. (30 pts.)

5. Describe your level of familiarity with C++. (10 pts.)