CS354
Course Introduction

Don Fussell
Computer Science Department
The University of Texas at Austin
CS 354 - Computer Graphics

- **Instructor:** Don Fussell
  - fussell@cs.utexas.edu
  - Office: GDC 5.510
  - Office Hours: TTh 10-11am

- **Teaching Assistant:** Randy Smith
  - agrippa@cs.utexas.edu
  - Office Hours: TBD

- **Location:** GDC 5.302
- **Lectures:** TTh 11:00-12:30pm
Objectives

- Fundamentals of computer graphics
  - Transformations and viewing
  - Rasterization and ray tracing
  - Lighting and shading
  - Graphics hardware technology
  - Mathematics for computer graphics

- Practical graphics programming
  - OpenGL programming
  - Shader programming
Course Expectations

- You should
  - Attend regularly and keep up – in-class short quizzes
- Do the programming assignments
  - Nearly everything you learn in this course will come from these
  - You need to know C/C++
  - Use office hours if you need help
  - No cheating (see syllabus and UT Austin policy)
  - If they’re not fun, you’re doing it wrong
- Tests and homework
  - Less fun, and useful, than programming projects
  - Good for covering math and concepts
Grading

- Programming projects 60%
- Homework and quizzes 10% (if relevant, otherwise this 10% goes to programming projects)
- Exams 30%
  - 2 exams - Middle semester and end of class 15% each
  - No final
Textbook

  - by Edward Angel and Dave Shreiner
  - Pearson, 7th edition

- Currently only recommended
  - It costs $147 list
  - Very helpful, but we don’t require it
  - Older editions also useful
Other Useful Resources

- OpenGL
  See links on course webpage

  **OpenGL Programming Guide**
  “the red book”

  **OpenGL SuperBible**
  **OpenGL A Primer**

- Supplemental books

  Eric Lengyel
  *Mathematics for 3D Game Programming and Computer Graphics*

  Real-Time Rendering
  Eric Haines, Tomas Akenine-Moller, Naty Hoffman
Computer Graphics Applications

Film, television

[Pixar 2010]

Product design

[CATIA]
Computer Graphics Applications

Games

Training

[Commercial simulators]

[Skyrim]
Computer Graphics Applications

GUls

[Android 4.0]

Apps

[Audi]
Computer Graphics Applications

2d and 3d printing

[HP]

Digital imaging, computational photography

[Canon]

[MakerBot]
Computer graphics

Very interdisciplinary compared to many CS topics

Geometry and Mathematics of Surfaces

Animation & Simulation

Display & Input Technology

Physics of Light Transport

Human Perception
What we will cover

- **Computer-based representation of**
  - **Geometry**
    - [Litke et.al. 2001]
  - **Appearance**
    - [george3738]
  - **Motion**
    - [Chai & Hodgins, 2005]
What we won’t cover

- Digital content creation
  - No Photoshop, no Maya or 3D Studio Max
  - Computer Science class, not an art class
- 2d stuff, GUIs
- C/C++ programming
  - You should already know C or C++ under Linux
    - Not just the language
    - Need to know debugging and software practices
    - Programming projects assume Linux – supported in GDC labs
- Many advanced techniques
Graphics and vision

- **Computer graphics**
  - Takes an abstract representation of a “scene” within a computer’s memory and converts it to concrete representing a view of that scene.
  - 40 year old discipline – now very advanced because this is the easy stuff.

- **Visual system**
  - Takes concrete imagery and converts into an abstract representation of a scene in your brain (what you see is a model you construct).
  - Computer vision tries to do this with a computer, it’s very hard.

---

University of Texas at Austin    CS354 - Computer Graphics    Don Fussell
In computer graphics, we form images which are generally two dimensional using a process analogous to how images are formed by physical imaging systems.

- Cameras
- Microscopes
- Telescopes
- Human visual system
Elements of Image Formation

- Objects
- Viewer
- Light source(s)

- Attributes that govern how light interacts with the materials in the scene
- Note the independence of the objects, the viewer, and the light source(s)
Light

- *Light* is the part of the electromagnetic spectrum that causes a reaction in our visual systems.

- Generally these are wavelengths in the range of about 350-750 nm (nanometers).

- Long wavelengths appear as reds and short wavelengths as blues.
One way to form an image is to follow rays of light from a point source finding which rays enter the lens of the camera. However, each ray of light may have multiple interactions with objects before being absorbed or going to infinity.
Global vs Local Lighting

- Cannot compute color or shade of each object independently
  - Some objects are blocked from light
  - Light can reflect from object to object
  - Some objects might be translucent
Luminance and Color Images

- **Luminance Image**
  - Monochromatic
  - Values are gray levels
  - Analogous to working with black and white film or television

- **Color Image**
  - Has perceptual attributes of hue, saturation, and lightness
  - Do we have to match every frequency in visible spectrum? No!
Three-Color Theory

- Human visual system has two types of sensors
  - Rods: monochromatic, night vision
  - Cones
    - Color sensitive
    - Three types of cones
    - Only three values (the *tristimulus* values) are sent to the brain
- Need only match these three values
  - Need only three *primary* colors
Images are 2-d array of numbers corresponding to pixels on screen
Numbers are in frame buffer memory
1-1 correspondence between frame buffer pixels and screen pixels
Additive and Subtractive Color

- **Additive color**
  - Form a color by adding amounts of three primaries
    - Monitors, projection systems, positive film
  - Primaries are Red (R), Green (G), Blue (B)

- **Subtractive color**
  - Form a color by filtering white light with cyan (C), Magenta (M), and Yellow (Y) filters
    - Light-material interactions
    - Printing
    - Film
Next Lecture

- Vector and affine math

- Assignments
  - Make sure your CS Unix account is active
  - Our first assignment will be pretty large – a ray tracer

- Thanks to Mark Kilgard and Ed Angel for material in many of these slides