Overview

- What is CS 4600?
- What should know (pre-reqs)?
- What will you get out of this course?

Website: www.eng.utah.edu/~cs4600

What is CS 4600?

- Broad introduction to Computer Graphics
  - Software
  - Hardware
  - Applications
- Bottom-Up (Bottom-up/Top-down approach)
- Shader-Based WebGL
  - Integrates with HTML5
  - Code runs in latest browsers
  - Start with HTML 2D graphics

Prerequisites

- Good programming skills in C (or C++, or Java)
- Basic Data Structures
  - Linked lists
  - Arrays
- Geometry
- Simple vector operations

Plans for this semester

- Start with bottom-up (unlike the book)
  - Shaders before the concepts are a bad idea
  - We’ll use JavaScript for 2D drawing
  - We’ll move to WebGL (OpenGL) for 3D and include shaders
- Shaders
  - Most computer graphics use OpenGL but still use fixed-function pipeline
  - does not require shaders
  - Does not make use of the full capabilities of the graphics processing unit (GPU)
- WebGL
  - With HTML5, WebGL runs in the latest browsers
  - makes use of local hardware
  - no system dependencies (headache for Mac users and grading)

References

- The OpenGL Programmer’s Guide (the Redbook) 8th Edition
- OpenGL ES 2.0 Programming Guide
- WebGL Programming Guide
- WebGL Beginner’s Guide
- WebGL: Up and Running
- JavaScript: The Definitive Guide
Web Resources

- www.cs.unm.edu/~angel/WebGL/7E
- www.opengl.org
- get.webgl.org
- www.kronos.org/webgl
- www.chromeexperiments.com/webgl
- learningwebgl.com

Examples

Book website:

www.cs.unm.edu/~angel/WebGL/7E

What you’ll get out of CS 4600

- You will learn the important ideas of Computer Graphics
- You will learn a commonly-used API, OpenGL
- You will have seen some of the most interesting applications of mathematics to Computer Science
  - Sampling
  - Projective Geometry
- You can create some interesting programs
- You will be exposed to more interesting ideas than we have time to pursue in class

What you’ll get out of CS 4600

// You will be able to read a program like this by the end of the semester...

```java
public void init(GLDrawable drawable) {
    float[] mat_specular = {1.0f, 1.0f, 1.0f, 1.0f};
    float[] mat_diffuse = {1.0f, 1.0f, 1.0f, 1.0f};
    float[] mat_ambient = {1.0f, 1.0f, 1.0f, 1.0f};
    float mat_shininess = 100.0f;
    float[] light_ambient = {0.0f, 0.0f, 0.0f, 1.0f};
    float[] light_diffuse = {1.0f, 1.0f, 1.0f, 1.0f};
    float[] light_specular = {1.0f, 1.0f, 1.0f, 1.0f};
    GL gl = drawable.getGL();
    /* set up ambient, diffuse, and specular components for light 0 */
    gl.glLightfv(GL.GL_LIGHT0, GL.GL_AMBIENT, light_ambient);
    gl.glLightfv(GL.GL_LIGHT0, GL.GL_DIFFUSE, light_diffuse);
    gl.glLightfv(GL.GL_LIGHT0, GL.GL_SPECULAR, light_specular);
    gl.glShadeModel(GL.GL_SMOOTH); /* enable smooth shading */
    gl.glEnable(GL.GL_LIGHTING);   /* enable lighting */
    gl.glEnable(GL.GL_LIGHT0);     /* enable light 0 */
    gl.glEnable(GL.GL_DEPTH_TEST); /* enable z buffer */
    gl.glClearColor(1.0f, 1.0f, 1.0f, 1.0f);
    gl.glColor3f(0.0f, 0.0f, 0.0f);
}
```

What Computer Graphics isn't

- Let’s talk about what graphics isn’t.
- **Image processing:** taking an image and looking for features
  - Red-eye removal: look for red dots, and change them
  - Color correction
  - Finding parts on assembly line
  - Face detection and recognition
- **Machine Vision:** application of Image Processing to Robotics

What Computer Graphics isn't

- **Graphics User Interfaces:** writing user interfaces
  - Menus and widgets
- We’ll use GUIs though and you’ll have to program them in JavaScript

Xerox 8010 compound document en.wikipedia.org
What Computer Graphics isn't

• **Visualization**: Can be used to
  – Tracking a storm
  – Show Gulf Stream currents

[Image of Gulf Stream currents]

What Computer Graphics is

• All aspects of creating an image with a computer
  – Modelling the objects in a scene
  – Modelling physics of the objects & interactions
  – Capturing object’s geometry
    • Lines, Points, ...
  – Picking a point of view
    • Pick a FOV (Zoom)
  – Rendering the image

[Image of Pixar Tuturial]

What skills are important?

• **Mathematics**
  – Modelling the objects
    – Following transformations from objects to points on screen

• **Programming**
  – Writing programs to create objects and images
  – Making the calls to the graphics system

• **Creativity**
  – Create interesting stories. Draw memorable characters.

• Some of you will know more of one subject than the other.
  – We will need to help each other out.

Why Study Computer Graphics?

• Widely used in two major sectors of the US economy
  – Video Games
  – Motion Pictures

• Used in many other areas, such as Visualization

• An interesting application of ideas
  – Some very clever algorithms, cutting edge hardware

• Can help you understand why it takes so long to refresh an image
  – It is fun!

Prereqs:

• Normalized Vector?
• Matrix multiply?
• Vector multiply?
• Dot-product? (what is it?)

\[ \mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| |\mathbf{b}| \cos \theta \]
Prereqs:
- Cross Product?
  - Properties?

Prereqs:
- Cross Product?
  - Properties?
  - Compute?

\[
\mathbf{a} \times \mathbf{b} = \begin{vmatrix}
i & j & k \\
\mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 \\
\mathbf{b}_1 & \mathbf{b}_2 & \mathbf{b}_3
\end{vmatrix}
\]

Cross Product
\[
\mathbf{V}_1 = [3, -1, 0]
\mathbf{V}_2 = [4, -3, 1]
\]

\[
\begin{array}{ccc}
x & y & z \\ 3 & -1 & 0 \\ 4 & -3 & 1 \\
\end{array}
\]
Det:

\[
\begin{array}{ccc}
x & y & z \\ 3 & -1 & 0 \\ 4 & -3 & 1 \\
\end{array}
\]

\(-1 \cdot 0\),

\(\times\) WRONG!
Cross Product

\[ \mathbf{V}_1 = [3, -1, 0] \quad \mathbf{V}_2 = [4, -3, 1] \]

\[
\begin{array}{ccc}
 x & y & z \\
 3 & -1 & 0 \\
 4 & -3 & 1 \\
\end{array}
\]

\[ \text{Det: } -1, -3, -9 \cdot (-4) \]

Cross Product

\[ \mathbf{V}_1 = [3, -1, 0] \quad \mathbf{V}_2 = [4, -3, 1] \]

\[
\begin{array}{ccc}
 x & y & z \\
 3 & -1 & 0 \\
 4 & -3 & 1 \\
\end{array}
\]

\[ \text{Det: } -1, -3, -5 \]

Cross Product Sarrus' scheme

\[ \mathbf{V}_1 = [3, -1, 0] \quad \mathbf{V}_2 = [4, -3, 1] \]

\[
\begin{array}{ccc}
 x & y & z \\
 3 & -1 & 0 \\
 4 & -3 & 1 \\
\end{array}
\]

\[ \text{Det: } -0, -1 \]

Cross Product Sarrus' scheme

\[ \mathbf{V}_1 = [3, -1, 0] \quad \mathbf{V}_2 = [4, -3, 1] \]

\[
\begin{array}{ccc}
 x & y & z \\
 3 & -1 & 0 \\
 4 & -3 & 1 \\
\end{array}
\]

\[ \text{Det: } -1, -3, -9 \cdot (-4) \]
Cross Product **Sarrus’ scheme**

V1 = [3, -1, 0]  V2 = [4, -3, 1]

\[
\begin{array}{ccc|ccc}
\times & y & z & \times & y & z \\
3 & -1 & 0 & 3 & -1 & 0 \\
4 & -3 & 1 & 4 & -3 & 1 \\
\end{array}
\]

Det: -1, -3, -5

**Line Equations**

- Explicit form: \( y = mx + b \)
- Implicit form: \( f(x, y) = Ax + By + C = 0 \)
- Parametric form: \( P(x, y) = P_0 + t D \)

**Computer Graphics**

- **Computer graphics** deals with all aspects of creating images with a computer
  - Hardware
  - Software
  - Applications

**Example**

- Where did this image come from?
- What hardware/software did we use to produce it?
Preliminary Answer

- **Application**: The object is an artist’s rendition of the sun for an animation to be shown in a domed environment (planetarium)
- **Software**: Maya for modeling and rendering but Maya is built on top of OpenGL
- **Hardware**: PC with graphics card for modeling and rendering


- Computer graphics goes back to the earliest days of computing
  - Strip charts
  - Pen plotters
  - Simple displays using A/D converters to go from computer to calligraphic CRT
- Cost of refresh for CRT too high
  - Computers slow, expensive, unreliable

Cathode Ray Tube (CRT)

Can be used either as a line-drawing device (calligraphic) or to display contents of frame buffer (raster mode)

Shadow Mask CRT


- **Wireframe graphics**
  - Draw only lines
- Sketchpad
- Display Processors
- Storage tube
Sketchpad

- Ivan Sutherland’s PhD thesis at MIT
  - Recognized the potential of man-machine interaction
  - Loop
    - Display something
    - User moves light pen
    - Computer generates new display
  - Sutherland also created many of the now common algorithms for computer graphics

Display Processor

- Rather than have the host computer try to refresh display, use a special purpose computer called a display processor (DPU)

  ![Diagram of Display Processor]

- Graphics stored in display list (display file) on display processor
- Host compiles display list and sends to DPU

Direct View Storage Tube

- Created by Tektronix
  - Did not require constant refresh
  - Standard interface to computers
    - Allowed for standard software
    - Plot3D in Fortran
  - Relatively inexpensive
    - Opened door to use of computer graphics for CAD community
  - Drew lines - vector graphics


- Raster Graphics
  - Beginning of graphics standards
    - IFIPS
      - GKS: European effort
        - Becomes ISO 3D standard
      - Core: North American effort
        - 3D but fails to become ISO standard
  - Workstations and PCs

Raster Graphics

- Image produced as an array (the raster) of picture elements (pixels) in the frame buffer

Raster Graphics

- Allows us to go from lines and wire frame images to filled polygons
PCs and Workstations

- Although we no longer make the distinction between workstations and PCs, historically they evolved from different roots
  - Early workstations characterized by
    - Networked connection: client-server model
    - High-level of interactivity
  - Early PCs included frame buffer as part of user memory
    - Easy to change contents and create images


- Realism comes to computer graphics
  - Smooth shading
  - Environment mapping
  - Bump mapping


- Special purpose hardware
  - Silicon Graphics geometry engine
    - VLSI implementation of graphics pipeline
- Industry-based standards
  - PHIGS
  - RenderMan
- Networked graphics: X Window System
- Human-Computer Interface (HCI)


- OpenGL API
- Completely computer-generated feature-length movies (Toy Story) are successful
- New hardware capabilities
  - Texture mapping
  - Blending
  - Accumulation, stencil buffers

Computer Graphics: 2003-

- Photorealism
- Graphics cards for PCs dominate market
  - Nvidia, ATI
- Game boxes and game players determine direction of market
- Computer graphics routine in movie industry: Maya, Lightwave
- Programmable pipelines
- New display technologies

Generic Flat Panel Display

- Vertical grid
- Light emitting elements
- Horizontal grid
Computer Graphics 2011-

- Graphics is now ubiquitous
  - Cell phones
  - Embedded

- OpenGL ES and WebGL
- Alternate and Enhanced Reality
- 3D Movies and TV

Image Formation

- 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

Ray Tracing and Geometric Optics

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.
Luminance and Color Images

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

Additive and Subtractive Color

- Additive color
  - Form a color by adding amounts of three primaries
    - CRTs, 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
    - Negative film