CSE 5542 - Real Time Rendering
TBT

(Not So) Real Time Rendering
Where?

Time - TR 11:10 AM – 12:30 pM
Place - DL 0264
Labs?

• Your own machine …

• Graphics PC Lab – CL 112D?

• Platforms: PC (visual studio), Mac OS X or Linux
Who Am I?
The Instructor

Name: Raghu Machiraju

Email: machiraju.1@osu.edu

Office hours: M: 3-4 PM
             TR 12:30 PM – 1:30 PM
Grading
The Grader

Name: Tzi-Husan Wei

Email: TBA

Office hours:
  Monday: 4:30-5:30 PM
  Wed: 3:00-5:00 PM
  Fri: 3:00-5:00 PM
Grading

- Labs: 45% (10+12+13+10) - Strict deadlines.
- Final Project: 15% - No Final Exam
- Quizzes: 20%(5x4)
- (Take Home) Midterm: 15%
Course Information
Information

Web:  http://www.cse.ohio-state.edu/~raghu/5542
Piazza:  https://piazza.com/osu/spring2015/cse5542/home
Prerequisite:  3901 (560) or 3902 or 3903
           math 2568 (568) or 571
    permission from the instructor
Pre-reqs
Must Haves

✓ You need to be enthusiastic about Computer Graphics

✓ You need to be fluent in C/C++/Java programming

✓ You need to be comfortable with linear algebra

✓ You need to be willing to get hands-dirty: OpenGL, WebGL, 3D Printing, GLSL, hardware
CAVEATS – Not a …
Closer to …

Capstone

What is a capstone project, you ask? The Foundation defines it as being both a process and event that asks each Bonner to reflect on past experiences and to articulate the communal and personal transformation brought about through participation in the program.

Senior Capstone
The Book

INTERACTIVE COMPUTER GRAPHICS
A Top-Down Approach with WebGL
Earlier ...
Diff 7e 6e

WebGL
Will Follow Text Closely
Useful Books – OpenGL, GLSL

OpenGL programming Guide, 8th edition

OpenGL Shading Language, 3rd edition

http://proquest.safaribooksonline.com
Useful Books - WebGL
Useful Books - JavaScript

Activate Your Web Pages

JavaScript
The Definitive Guide

David Flanagan

O'REILLY®

THE OHIO STATE UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
Useful Tool - Blender
Reference Books – OpenGL/ES

[Image of OpenGL ES 2.0 Programming Guide]

[Image of OpenGL ES 3.0 Programming Guide]

https://www.khronos.org/
Reference Books – HTML5

Beginning WebGL for HTML5

Brian Danchilla

Apress®
Reference Books

Computer Graphics Principle and Practice
3rd edition

Real-Time Rendering
3rd edition
Reference Books


OpenGL 4.0 Shading Language Cookbook
Reference - Und Others

[Images of books: OpenGL Development Cookbook, OpenGL SuperBible, and GLSL Essentials]
Reference
What do we Study?
Teapots
A Real One
Drawing Teapots
Step 1
Step 2
Step 3
Step 4
Step 5
Step 6
Manufacturing Teapots & Minions
Manufacturing Teapots

http://oncampus.osu.edu/replicating-in-3d/

https://www.ted.com/talks/lisa_harouni_a_primer_on_3d_printing

http://news.cornell.edu/stories/2013/04/rapid-reality-students-design-3-d-printed-products
You will all make these 😊

http://www.thingiverse.com/thing:68880
Manufacturing Teapots

http://onderin.de.buro.la/cura-ani-500-64c.gif
And These
Maybe Not This
Manufacturing Minions

https://www.youtube.com/watch?v=-2uY7rjhhMs
We will not do this …

Kevin Wolf, 3D Printer, Mechanical Eng., OSU
Virtual Teapots
An Icon !
Which one is real?
Real vs Virtual – Boston Museum
More fake ones!

http://codegolf.stackexchange.com/questions/22620/draw-the-utah-teapot
Shutterbug !
Toy Story
A Platonic Relationship !
Building with Blender

https://www.youtube.com/watch?v=QyiBwL2Scec
Another Method

https://www.youtube.com/watch?v=x8AiEi4aJ4g
You will be also doing this ...
And better than this!
OpenGL, GLSL
OpenGL, GLSL
Topics and Outcomes
To Reiterate
Learning Outcomes - Familiarity

- Basic understanding of graphics hardware/software
- Basics of modeling and 3D Printing
- Basics of interaction
Learning Outcomes - Skills

- OpenGL/GLSL to control graphics hardware
- WebGL/OpenGL-ES to allow interactions
- Blender to build models
Learning Outcomes - Cognitive

- Advanced real time rendering algorithms

- Integrating three different software suites – WebGL, GLSL, OpenGL

- Solid and curve modeling nuances
Specific Topics

• Overview of Graphics Hardware and Software
• Coordinate systems
• WebGL Interaction modes
  – HTML 5
  – Immediate vs. retained mode
• OpenGL geometry drawing
  - OpenGL vertex buffer objects
• OpenGL Shading Language
  - Vertex and fragment shaders
• 3D transformation and Viewing
Specific Topics

• Illumination
  – Flat, Gourad, Phong shading models
  – Fixed function pipeline and shaders

• Visibility and Z-buffering

• Texture Mapping
  – Image and procedural textures

• Bump, environment, & projective texture mapping
Specific Topics

• Real time shadows
• Particle Methods
• Advanced topics in shaders
  – Geometry shader
  – Tessellation shader
• Advanced topics in rendering and graphics
Image Formation
Image Formation

- Cameras
- Microscopes
- Telescopes
- Human visual system
Synthetic Camera Model
Image Formation (Eine Explanation)

Ray Tracing and Geometric Optics

Problemo?
The Real Thing!

http://www.uccs.edu/~rtirado/Astronomy_Texts/Light(Image_Formation).pdf
Essentials of Image Formation

- Objects
- Camera
- Light source(s)
- Light-material interaction
- Independence of objects, viewer, and light source(s)
Light
Luminance vs. Color

http://www.workwithcolor.com/color-luminance-2233.htm
How Many Colors?

Human visual system

– Rods: monochromatic, night vision
– Cones
  • Color sensitive
  • Three types of cones
  • Only three values (the \textit{tristimulus} values) are sent to the brain
– Three \textit{primary} colors – R, G, B
In Days Long Gone
Even in these days!
The Camera
Real Cameras
Viewing
Eine Simple Kamera – The Pinhole

Simple Perspective - find projection of point at \((x,y,z)\)

\[
x_p = -\frac{x}{z/d} \quad y_p = -\frac{y}{z/d} \quad z_p = d
\]
Perspective

... whilst in this view, some of the bottles have been adjusted in shape, size or position, mostly by re-drawing parts of them outside the construction lines. Note also how the shadows belong to 'sit them down' on the shelf.
Not Bad 😊
Let Us Go Digital

Rasters
Pixels & Resolution
Is A Pixel Really A Square?

A Pixel Is Not A Little Square,
A Pixel Is Not A Little Square,
A Pixel Is Not A Little Square!
(And a Voxel is Not a Little Cube)¹

Technical Memo 6

Alvy Ray Smith
July 17, 1995

http://alvyray.com/Memos/CG/Microsoft/6_pixel.pdf
An image?

8 bits/pixel

Gray scale Gray level

Black \[ \rightarrow \] 0
Gray \[ \rightarrow \] 128
White \[ \rightarrow \] 255
CCD Cameras - Resolution

Image Digitization

**Sampling:** Resolution

**Quantization:** Measured Value
Image Digitization

Sampling

Quantization
https://get.webgl.org/
Kewl Sites

http://www.chromeexperiments.com/webgl/
Coding in WebGL

• Can run WebGL on any recent browser
  – Chrome
  – Firefox
  – Safari
  – IE

• Code written in JavaScript

• JS runs within browser
  – Use local resources
Example: triangle.html
Example Code

```html
<!DOCTYPE html>
<html>
<head>
<script id="vertex-shader" type="x-shader/x-vertex">
attribute vec4 vPosition;
void main(){
    gl_Position = vPosition;
}
</script>
<script id="fragment-shader" type="x-shader/x-fragment">
precision mediump float;
void main(){
    gl_FragColor = vec4( 1.0, 0.0, 0.0, 1.0 );
}
</script>
</head>
</html>
```
<script type="text/javascript" src="../Common/webgl-utils.js"></script>
<script type="text/javascript" src="../Common/initShaders.js"></script>
<script type="text/javascript" src="../Common/MV.js"></script>
<script type="text/javascript" src="triangle.js"></script>
</head>
<body>
<canvas id="gl-canvas" width="512" height="512">
Oops ... your browser doesn't support the HTML5 canvas element
</canvas>
</body>
</html>
var gl;
var points;

window.onload = function init(){
    var canvas = document.getElementById( "gl-canvas" );
    gl = WebGLUtils.setupWebGL( canvas );
    if ( !gl ) { alert( "WebGL isn't available" );}
}

// Three Vertices

var vertices = [
    vec2( -1, -1 ),
    vec2(  0,  1 ),
    vec2(  1, -1 ),
    vec2(  1, -1 )
];
// Configure WebGL

gl.viewport(0, 0, canvas.width, canvas.height);
gl clearColor(1.0, 1.0, 1.0, 1.0);

// Load shaders and initialize attribute buffers

var program = initShaders(gl, "vertex-shader", "fragment-shader");
gl.useProgram(program);

// Load the data into the GPU

var bufferId = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, bufferId);
gl.bufferData(gl.ARRAY_BUFFER, flatten(vertices), gl.STATIC_DRAW);
// Associate out shader variables with our data buffer

    var vPosition = gl.getAttribLocation( program, "vPosition" );
    gl.vertexAttribPointer( vPosition, 2, gl.FLOAT, false, 0, 0 );
    gl.enableVertexAttribArray( vPosition );
    render();

};

function render() {
    gl.clear( gl.COLOR_BUFFER_BIT );
    gl.drawArrays( gl.TRIANGLES, 0, 3 );
}
JavaScript Notes

- **JavaScript (JS) is the language of the Web**
  - All browsers will execute JS code
  - JavaScript is an interpreted object-oriented language
JS Notes

• Is JS slow?
  – JS engines in browsers are getting much faster
  – Not a key issues for graphics since once we get the data to the GPU it doesn’t matter how we got the data there

• JS is a (too) big language
  – We don’t need to use it all
  – Choose parts we want to use
  – Don’t try to make your code look like C or Java