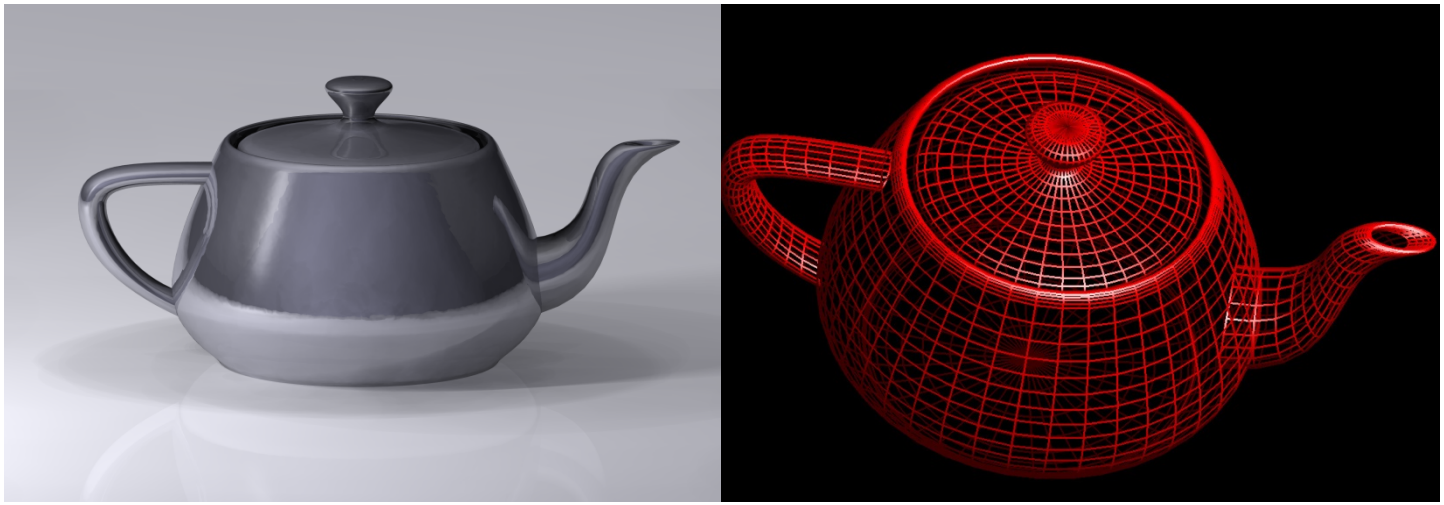


Foundations of Computer Graphics

Saurabh Ray

What is Computer Graphics?

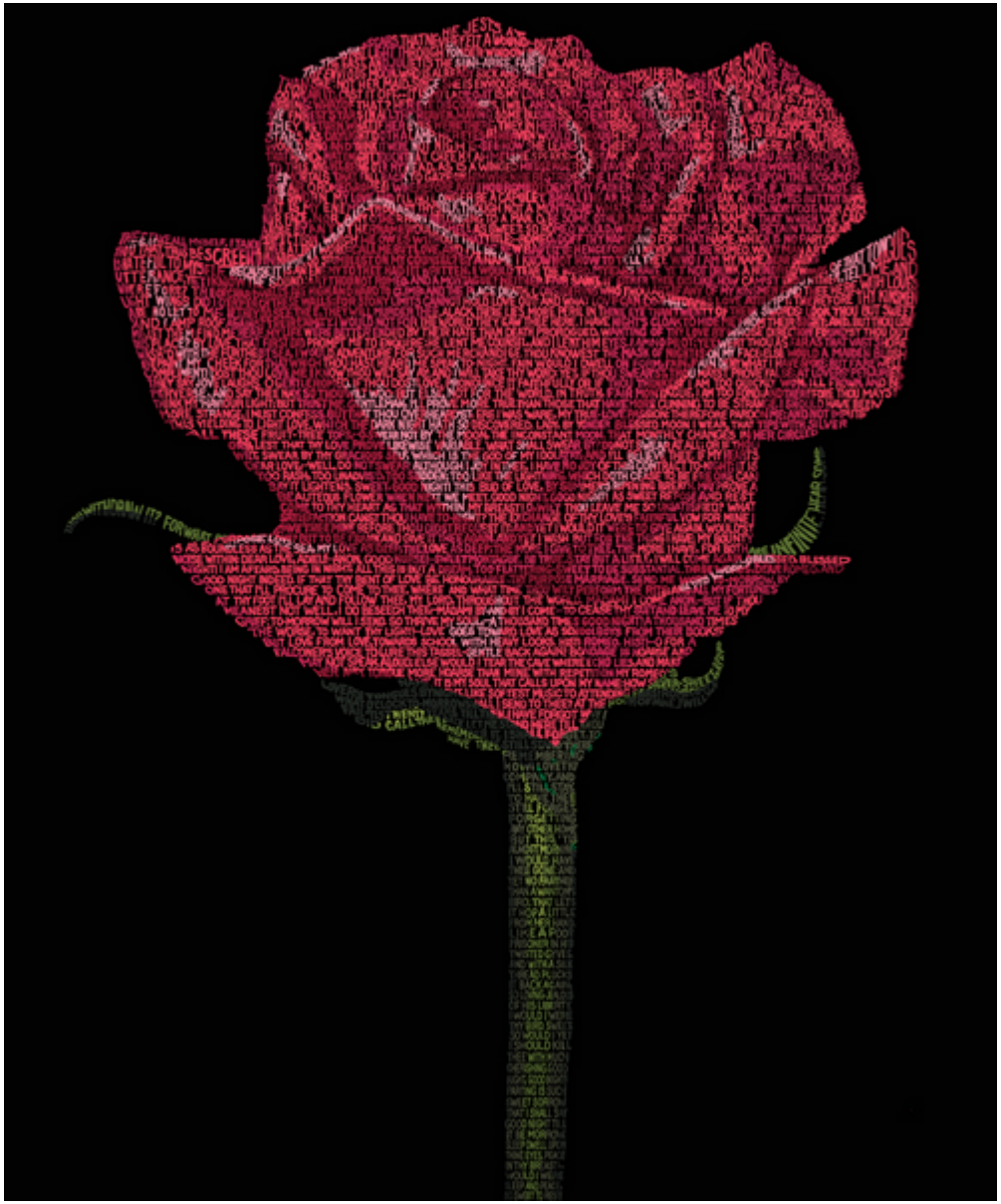
Almost everything on computers that is not text or sound.

The science and art of communicating visually via a computer's display and its interaction devices.

Cross-disciplinary field: physics, mathematics, human perception, human computer interation (HCI), engineering, graphic design, art etc.

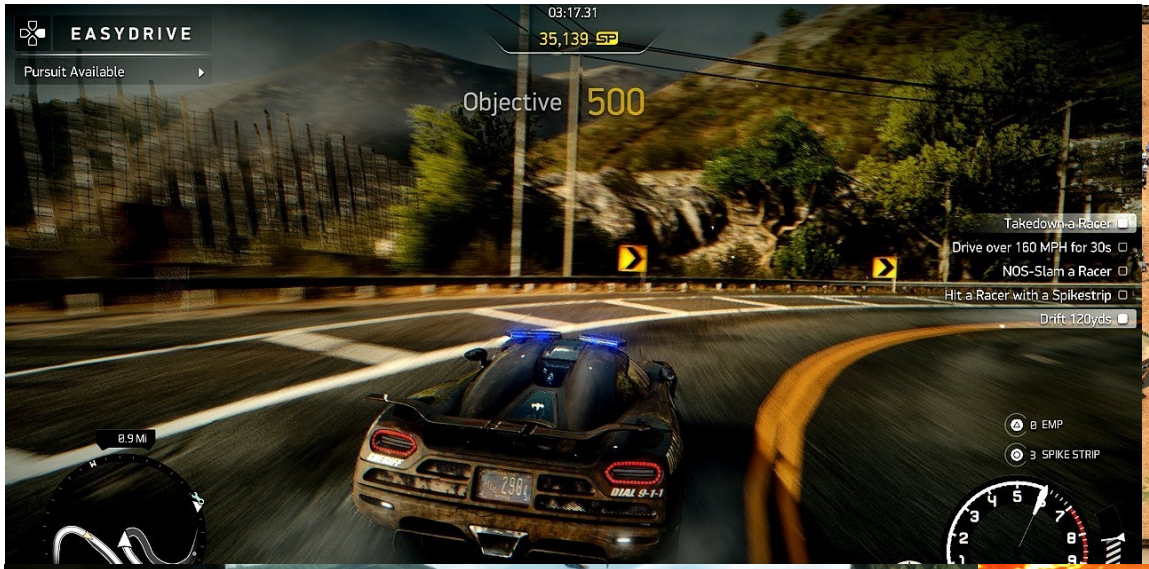
Why Computer Graphics?

A picture is worth a thousand words



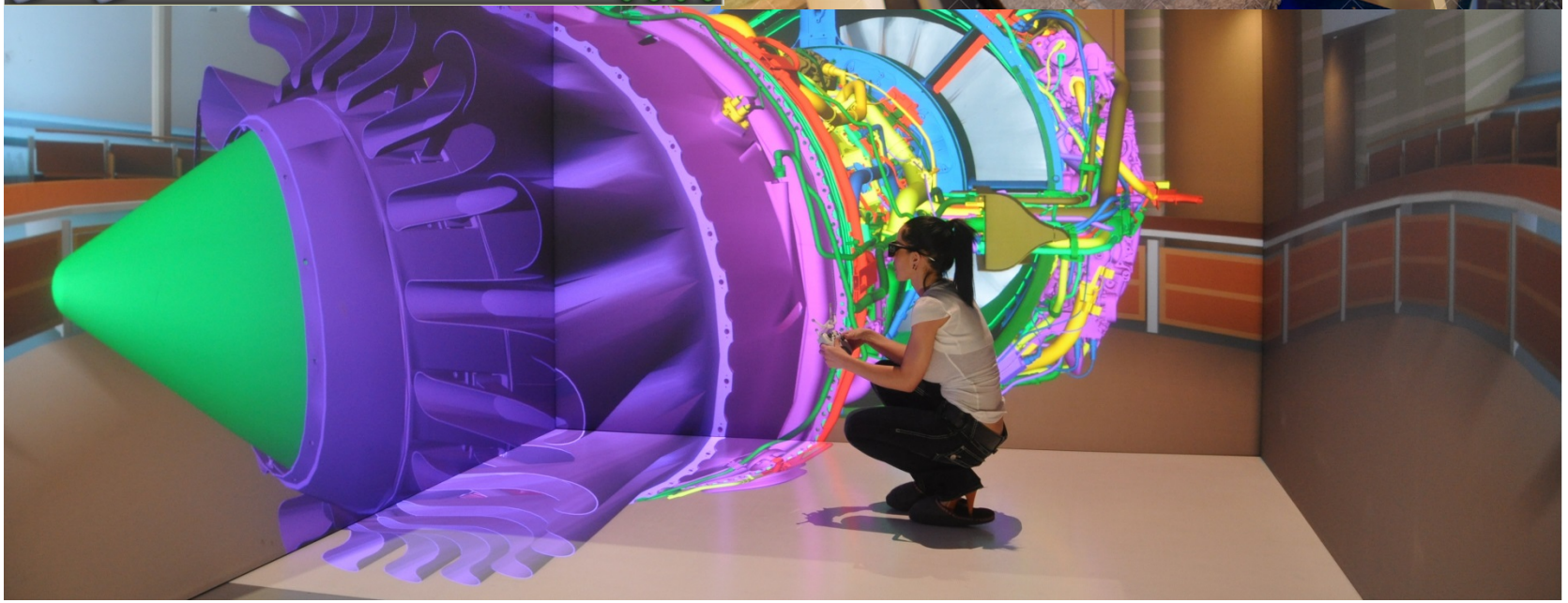
Movies

Video Games


EASYDRIVE
Pursuit Available
03:17.31
35,139 SP
Objective 500
Takedown a Racer
Drive over 160 MPH for 30s
NOS-Slam a Racer
Hit a Racer with a Spikestrip
Drift 120yds
0.9 Mi
0 EMP
3 SPIKE STRIP


Imperial Age

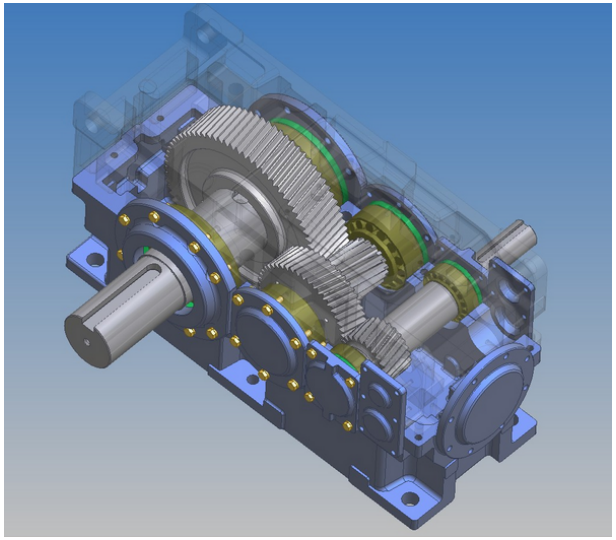
Virtual Reality



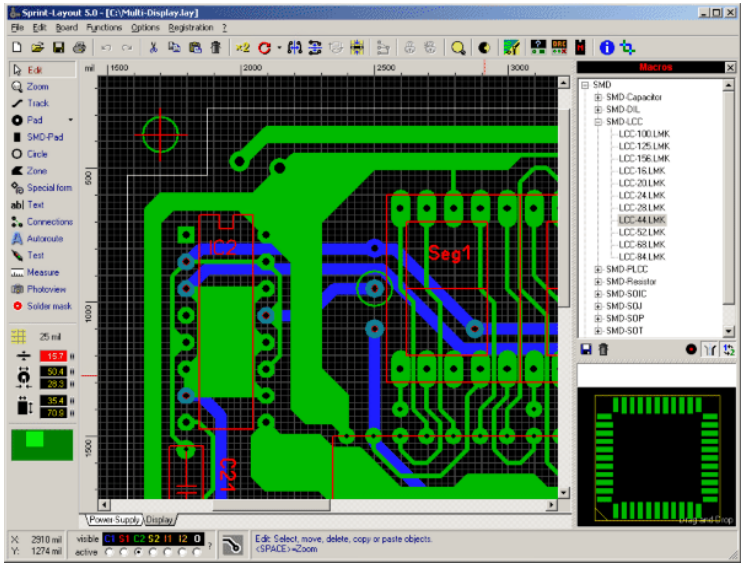
Computer Aided Design

Architecture

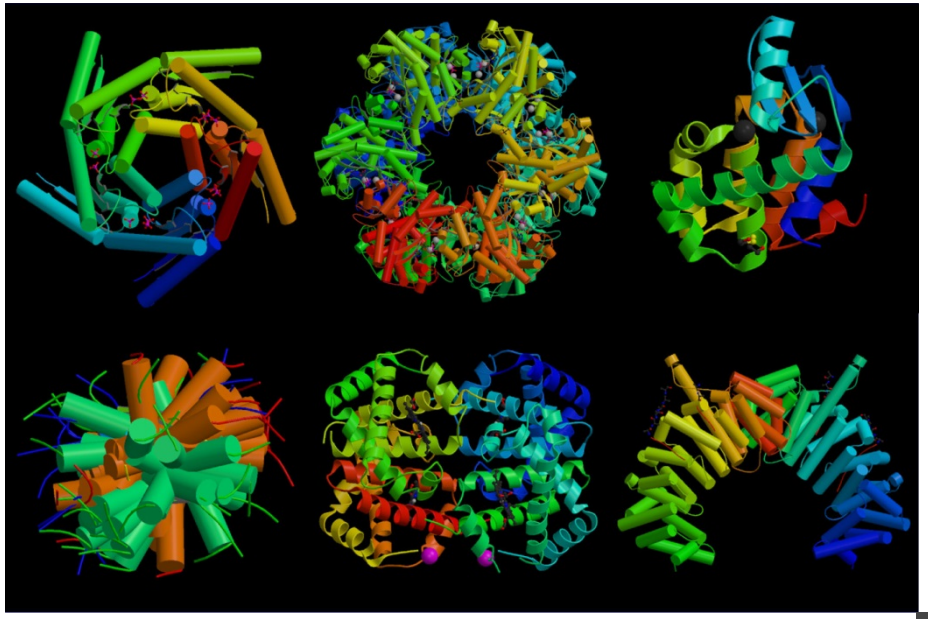
Mechanical Engineering

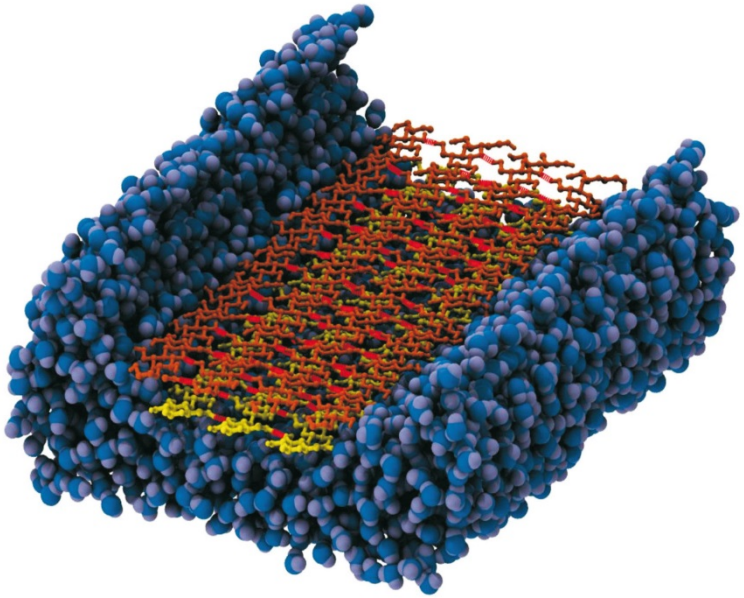


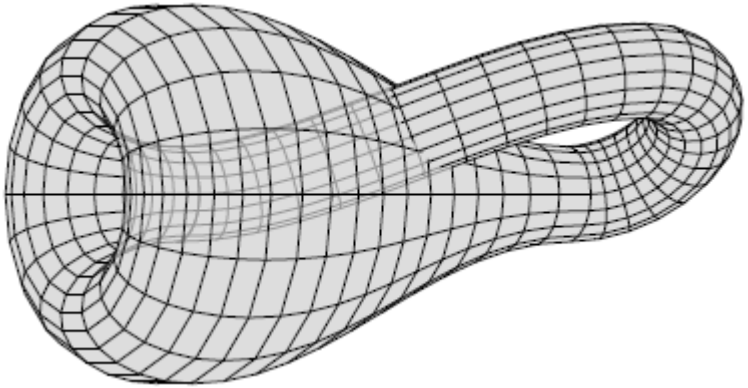
Electronics

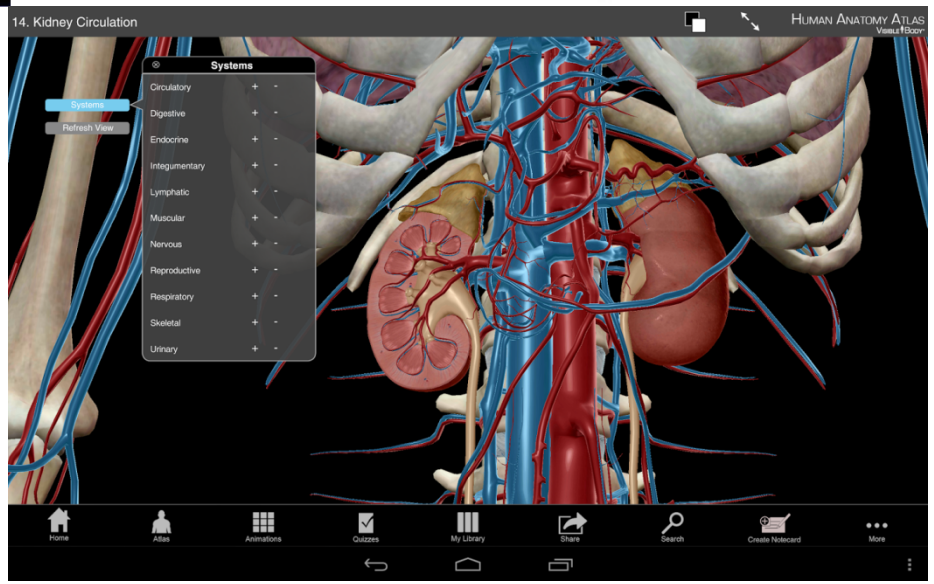


Education

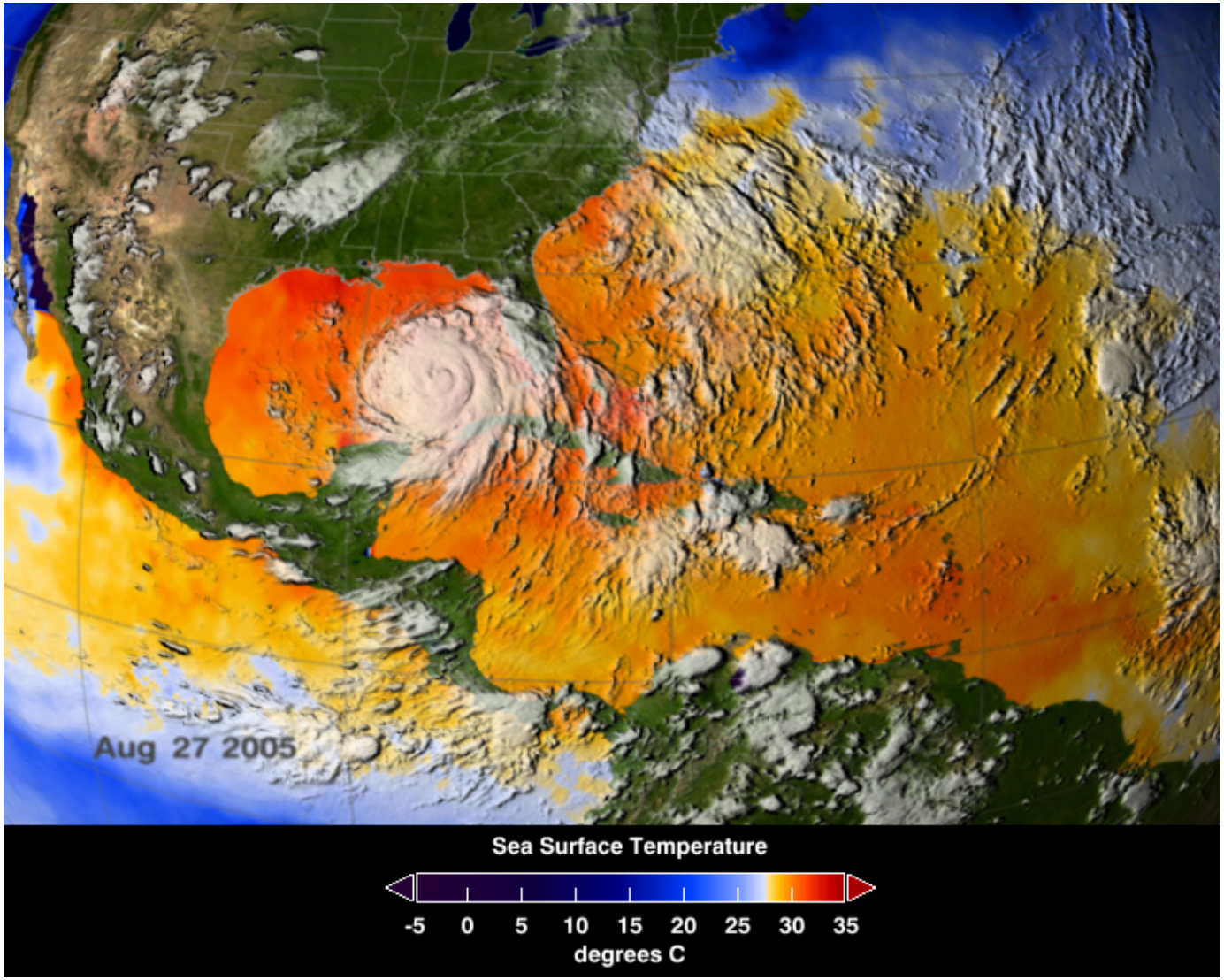






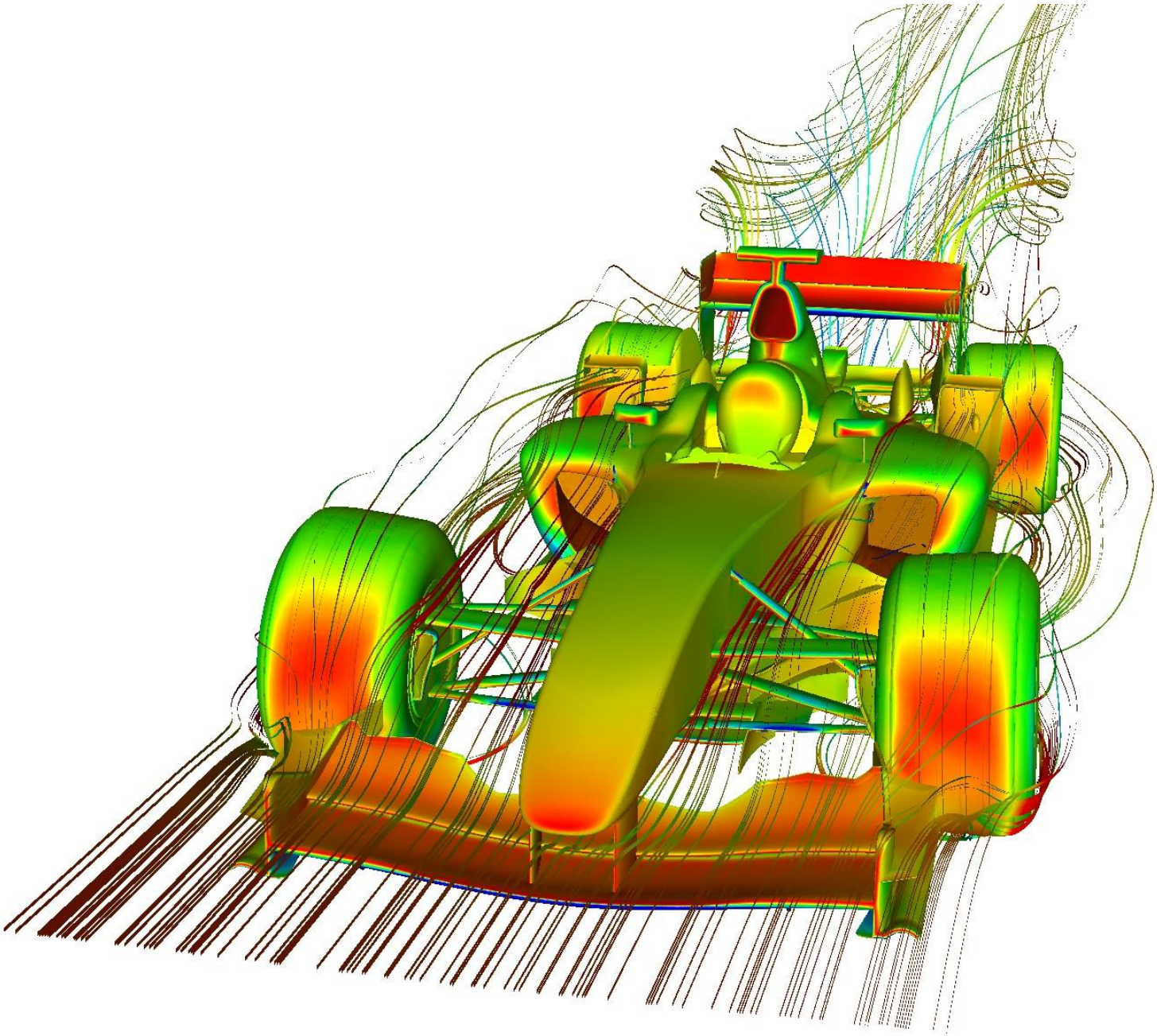

14. Kidney Circulation
Human Anatomy Atlas
Systems
Systems
Refresh View
Circulatory
Digestive
Endocrine
Integumentary
Lymphatic
Muscular
Nervous
Reproductive
Respiratory
Skeletal
Urinary
Home
Atlas
Animations
Quizzes
My Library
Share
Search
Create Notecard
More

Scientific Visualization

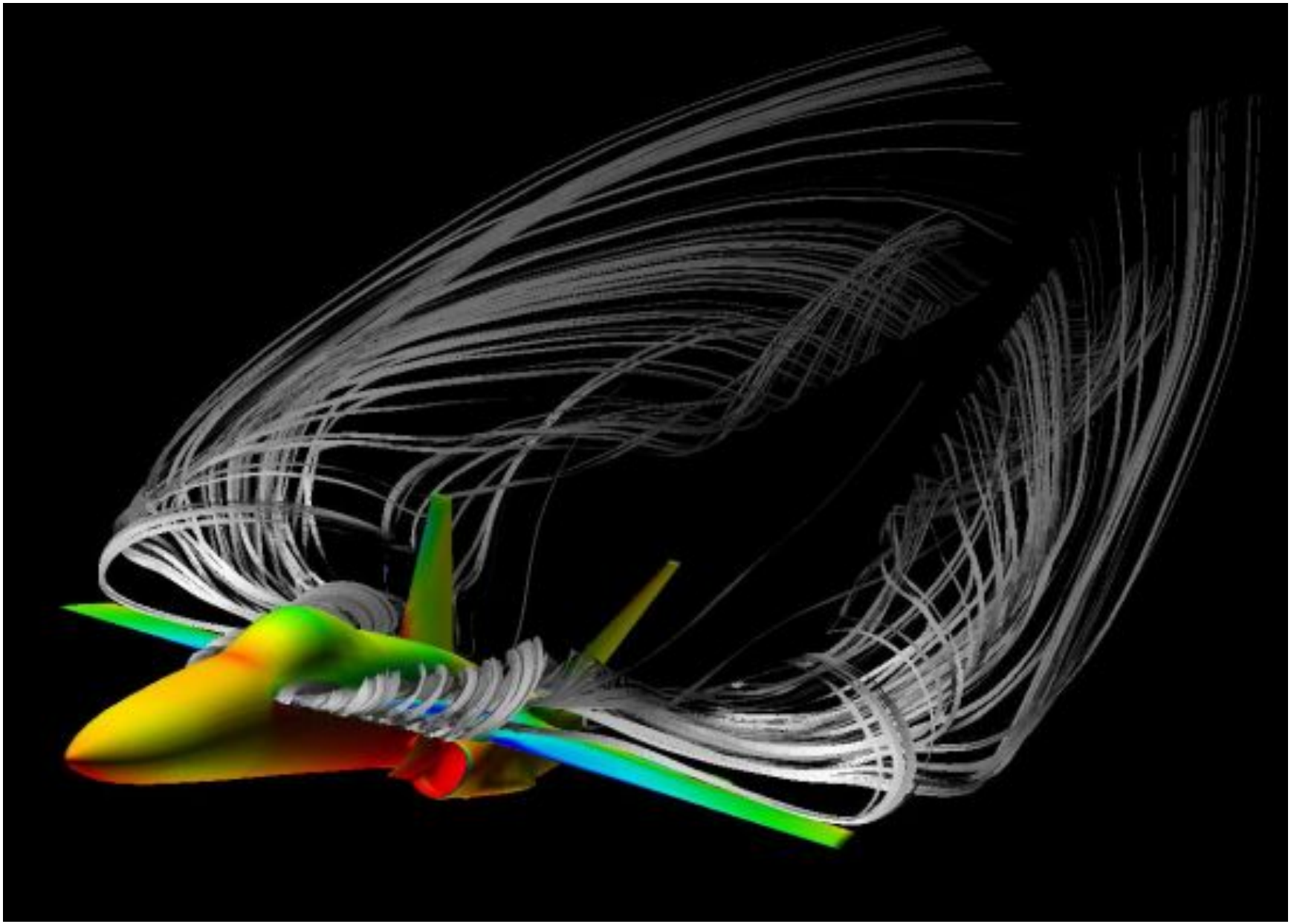


Hurricane Katrina

Scientific Visualization



Scientific Visualization



This course is about...

Fundamentals of computer graphics:

- Concepts, Algorithms
- Software, Hardware
- Applications

Modelling, representing and displaying geometric objects.

Input:

- a model of objects in a scene (geometric description + other attributes)
- a model of light (sources + directions of radiation)
- eye (or camera) location and direction of viewing

Goal: produce the view

Focus: Build things from scratch.

This course is not about...

Graphic Design

- Web page design
- Poster design
- Sketching and Painting

Software Packages

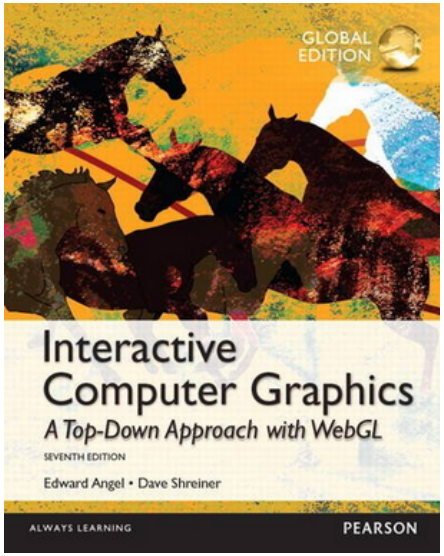
- Photoshop
- Autocad
- Maya, Blender

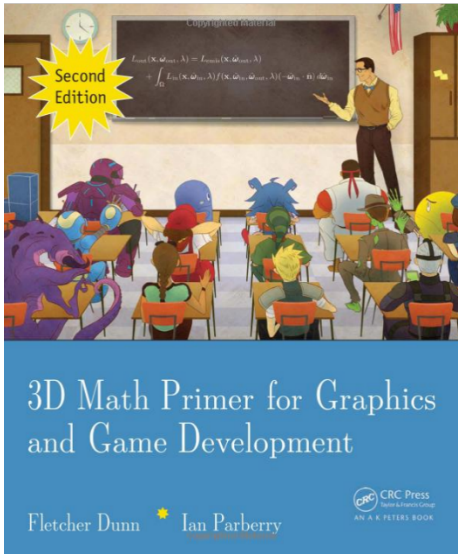
Game Design

Though not part of this course, I encourage you to learn a bit of Blender and Three.js

Administrativia

Books:





Useful website for learning WebGL:

https://webgl2fundamentals.org/

Office hours: Wednesdays 16:00 to 17:00
A2 187 (Computational Research Building)

Administrativia

Prerequisites:

- Programming
- Basic algorithms and data structures
- Basic linear algebra

Grading:

- 30% Assignments — *3 assignments, both theory and coding.*
- 30% Midterm Exam — *On* **April 2** *during class hours.*
- 40% Final Exam — *In the final exam period.*

There will also be weekly exercises that are not graded but will be useful for your own learning.

Grade Mapping Schema.

The following table indicates the minimum grade received if the overall score is above a certain threshold. For instance, if the overall score is 83% or more, the student receives a grade of B or better.

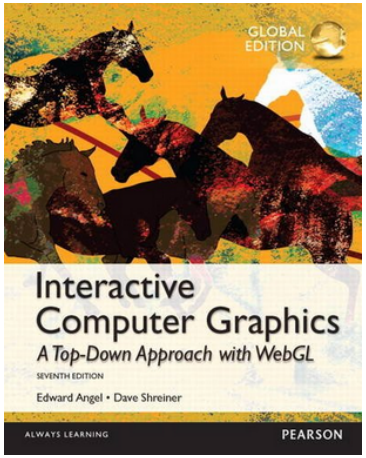
Minimum %	Grade
95	A
90	A-
87	B+
83	B
80	B-
77	C+
73	C
70	C-
67	D+
63	D
0	F

Administrativia

Main topics:

- Mathematics - Basic linear algebra, 3D transformations, projections
- Programming - WebGL API, Javascript, HTML, GLSL ES
- Graphics techniques - Lighting, Texturing, Meshing, Ray Tracing, Animation

Almost all of



+ selected topics

You will need to read the textbooks regularly. There are many things that cannot be discussed in the class.

Administrativia

Software:

HTML5 supports WebGL $\Longrightarrow$ you just need a browser, a text editor and a web server (to run code locally).

No system dependencies. Any major OS will work.

Any browser works. I use chrome because it includes a debugger.

Use any text editor that supports HTML and Javascript indentation.

I use notepad++ on windows.

I use mongoose webserver `https://cesanta.com/mongoose.shtml`
Works on Linux, Mac and Windows.

Basic Elements

Modelling

shape (geometry)

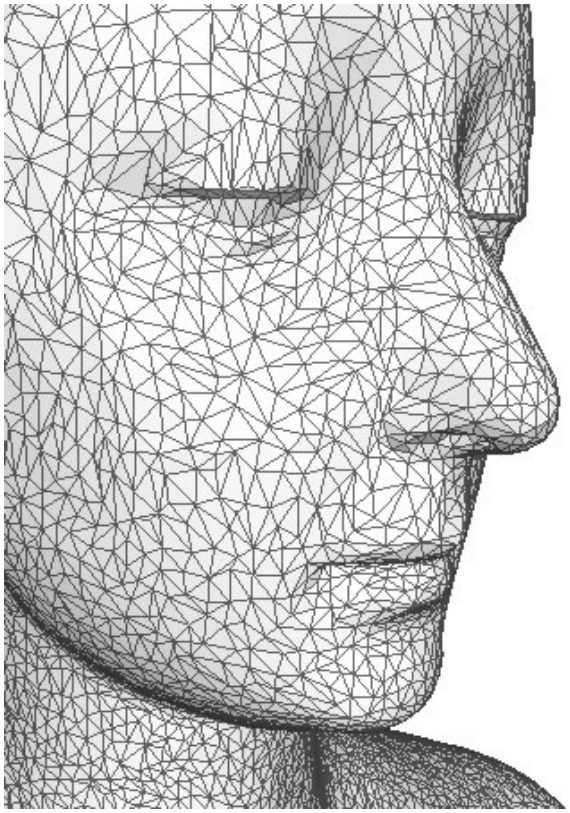
Rendering

display (shading, illumination, color, texture)

Animation

movement (dynamics)

Modelling



Description in terms of geometric primitives

Description using a mesh

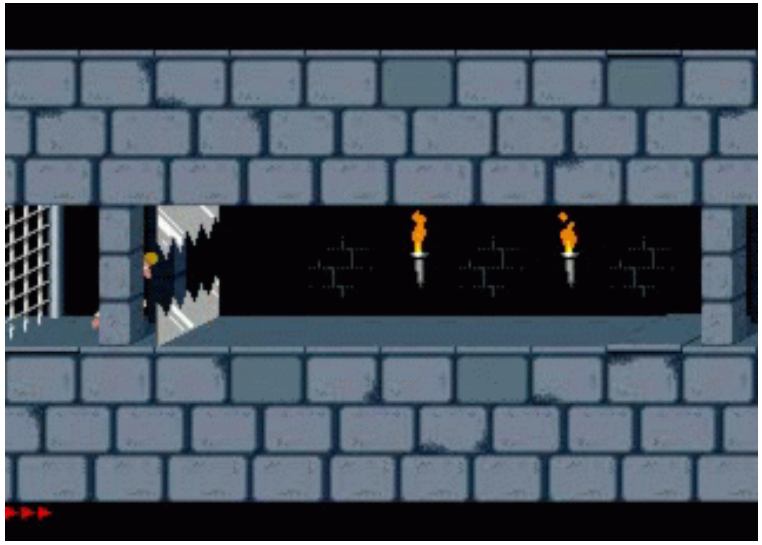
We are only concerened with the external surface, not the interior.

Rendering

Displaying a scence from a particular point of view.

Issues: transformation, clipping , color, shadows, texture etc

Animation



Evolution of an image over time.

- movement of objects in the scene
- movement of the camera
- change of lighting

Techniques:

- Keyframing, Procedural, Physics-based, Motion-capture

Demo: Mass Spring Systems

WebGL

WebGL is a technology that enables 3D graphics within web browsers.

Most popular technologies for 3D Graphics: **Direct3D, OpenGL**

Microsoft's proprietary API mainly on Windows

used on all major platforms

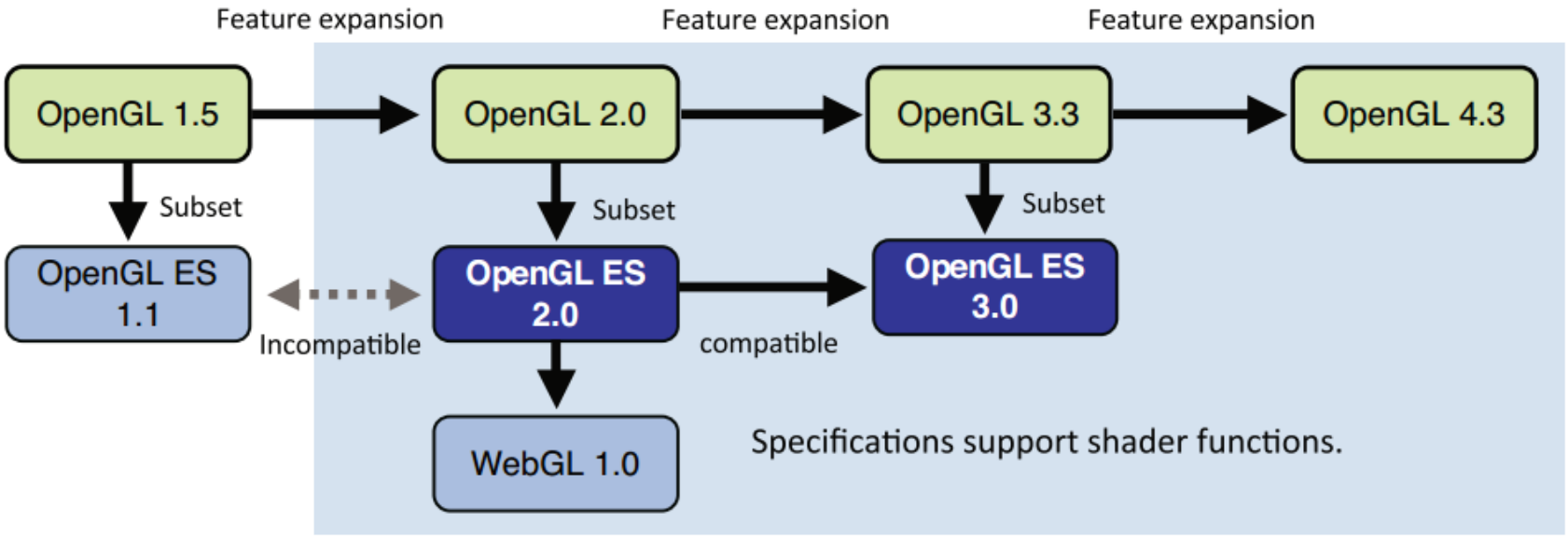
OpenGL was developed by Silicon Graphics Inc. in 1992.

Many versions since then.

Used in many applications (Maya, Blender, Photoshop, Google Earth etc).

OpenGL ES: derivative of OpenGL for embedded systems like smart phones, tablets, gaming consoles, etc.

WebGL



WebGL 2.0 based on OpenGL ES 3.0 was released in 2017.

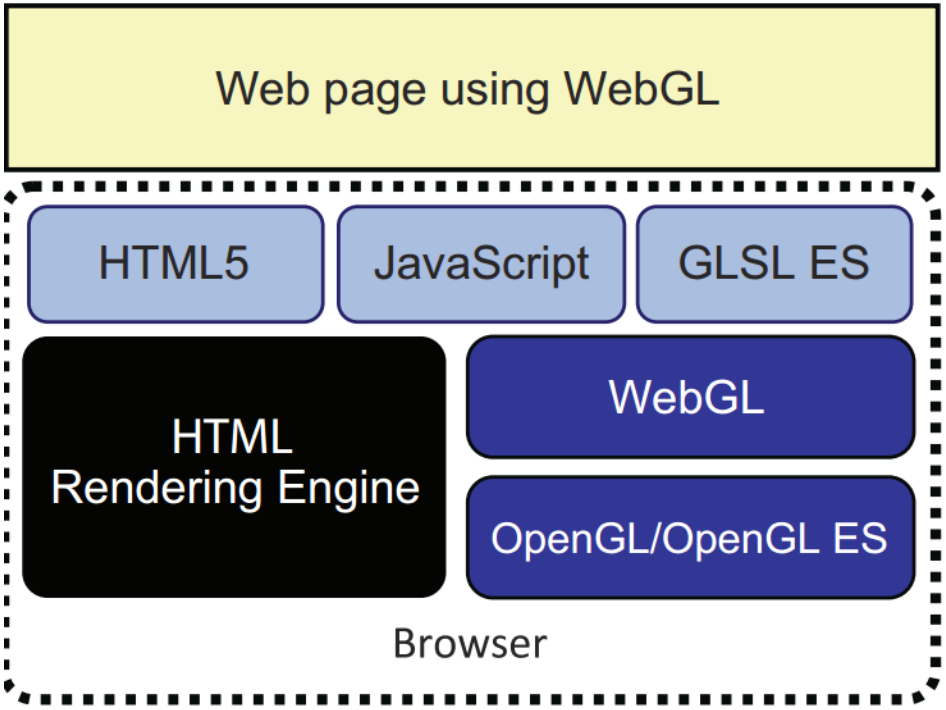
Key improvement in OpenGL version 2.0: **programmable shader functions**.

These enable sophisticated visual effects.

The shaders are written in the OpenGL Shading Language **GLSL**.

WebGL uses **GLSL ES.**

WebGL



Web pages are written in Hypertext Markup Language (HTML).

Latest version: **HTML5**

HTML describes the structure of the page. Functionality is described using the Javascript programming language.

GLSL ES code is written within Javascript.

WebGL Example: Clear the clutter

A simple example that clears the canvas with a background color.

File: ClearCanvas.html

```
<html>
<head>
  <script type="text/javascript" src="../Common/webgl-utils.js"></script>
  <script type="text/javascript" src="ClearCanvas.js"></script>
</head>
<body>
   <canvas id="gl-canvas" width="512" height="512">
    HTML5 Canvas not supported!
   </canvas>
</body>
</html>
```

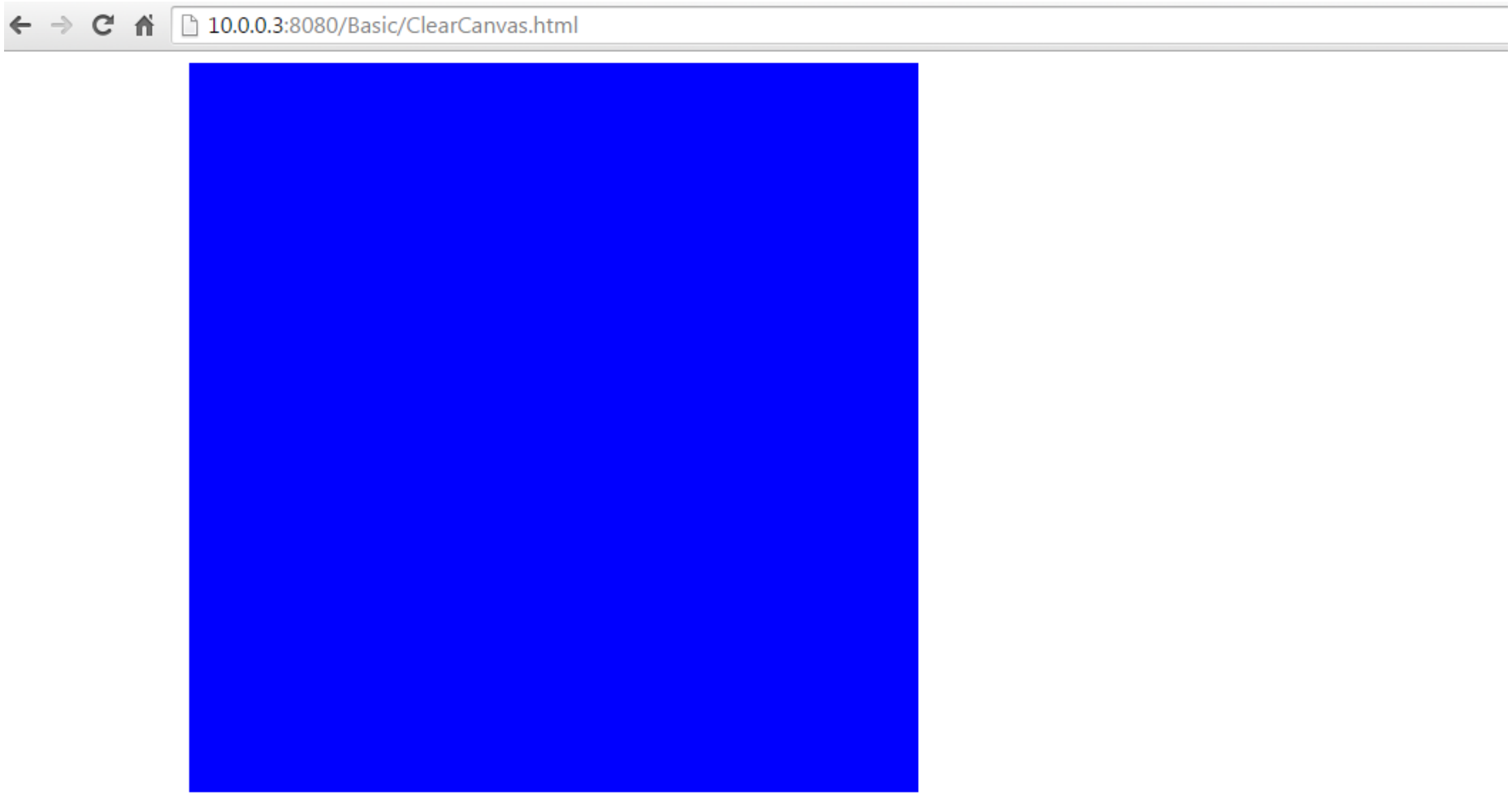
File: ClearCanvas.js

```
window.onload = function init() {
            var canvas = document.getElementById("gl-canvas");
            var gl = WebGLUtils.setupWebGL( canvas );
            if(!gl){alert("WebGL setup failed!");}

            gl.clearColor(0.0, 0.0, 1.0, 1);
            gl.clear(gl.COLOR_BUFFER_BIT);
}
```

WebGL Example: Clear the clutter

Loading the html file `ClearCanvas.html` in a browser gives you:



WebGL Example: Clear the clutter

File: `ClearCanvas.html`

```
<html>
<head>
  <script type="text/javascript" src="../Common/webgl-utils.js"></script>
  <script type="text/javascript" src="ClearCanvas.js"></script>
</head>
<body>
   <canvas id="gl-canvas" width="512" height="512">
    HTML5 Canvas not supported!
   </canvas>
</body>
</html>
```

We are asking the browser to load these Javascript files.

Displayed if the `canvas` *element is not supported.*

An HTML file consists of a bunch of **tags**.

Each tag describes some document content. Format: `<tag> content </tag>`

`<head>` ... `</head>` : information about the document

`<body>` ... `</body>` : visible content

`webgl-utils.js`: contains useful functions for setting up WebGL.

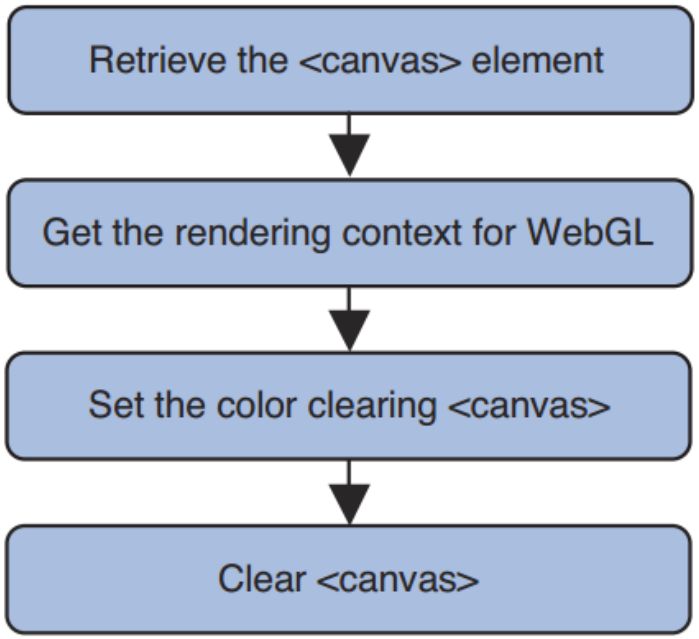
WebGL Example: Clear the clutter

File: ClearCanvas.html

This function should be called once all scripts are loaded.

```
window.onload = function init() {
    var canvas = document.getElementById("gl-canvas");
    var gl = WebGLUtils.setupWebGL( canvas );
    if(!gl){alert("WebGL setup failed!");}

    gl.clearColor(0.0, 0.0, 1.0, 1);
    gl.clear(gl.COLOR_BUFFER_BIT);
}
```



RGBA : (Red, Green, Blue, Alpha)

Alpha is for opaqueness.

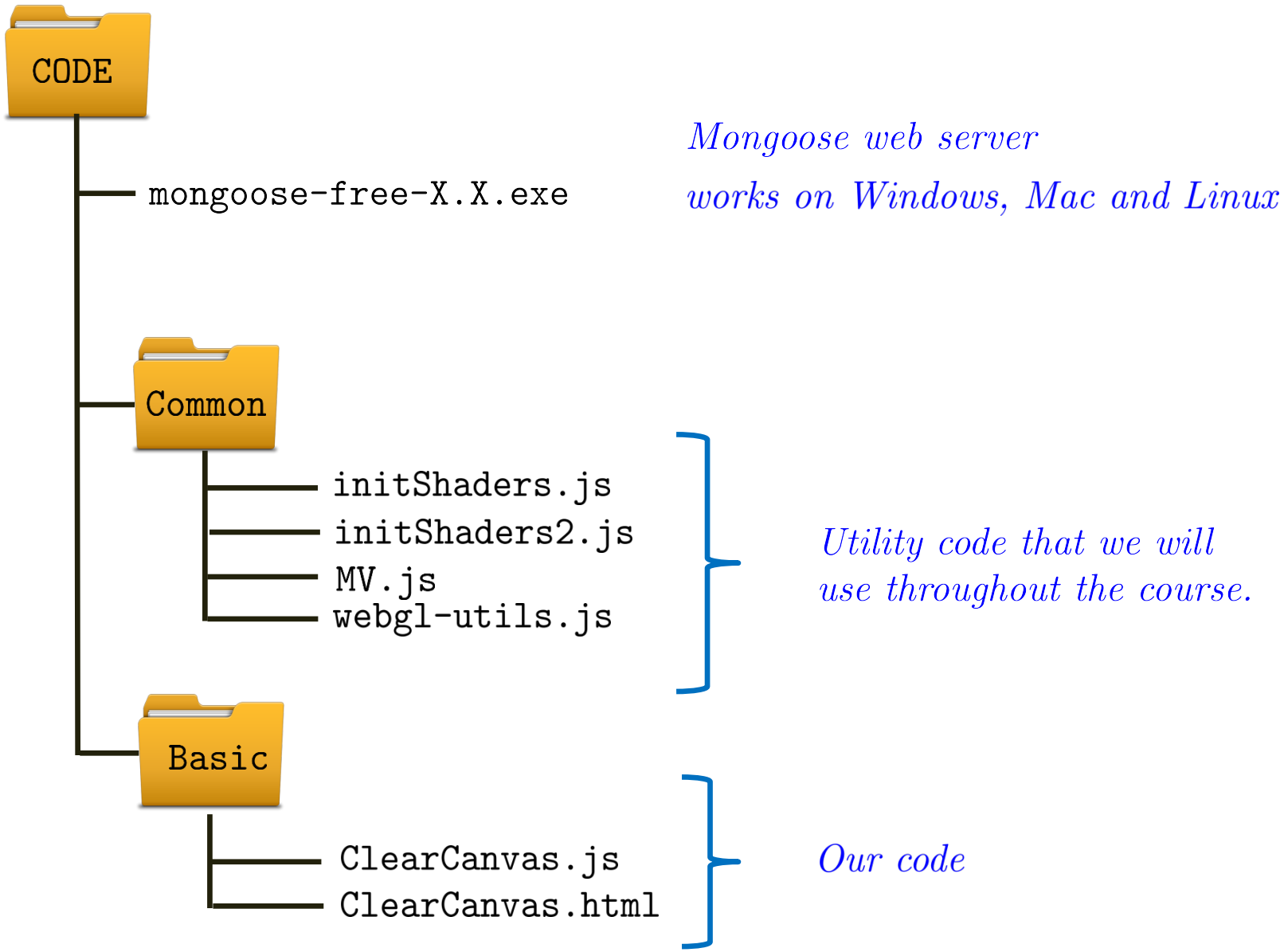
$\alpha = 0$: *fully transparent*

$\alpha = 1$: *fully opaque*

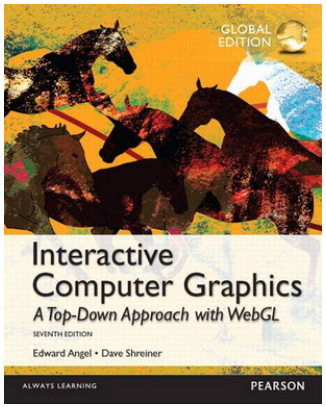
We are asking WebGL to use the color buffer.

WebGL Example: Clear the clutter

Folder Structure: *https://github.abudhabi.nyu.edu/sr194/CG-2020*



Please skim Chapter 1 of the book
by Angel and Shreiner.



Find out the WebGL versions supported by your browser:
http://webglreport.com/

Have fun with WebGL:
https://experiments.withgoogle.com/experiments?tag=WebGL