Final Exam

Thursday, May 12, 4:00–5:50 p.m.
Room 317, Warren Weaver Hall
Graphics on the Web

Raster Graphics

GIF: Graphics Interchange Format
JPEG: Joint Photographic Experts Group
PNG: Portable Network Graphics
HTML5 Canvas (renders directly)
Graphics on the Web

Vector Graphics

Flash (SWF files)

SVG: Scalable Vector Graphics

CSS3 (vector-like capability)
The Internet and the Web are separate but related things.

The Internet is a massive network of networks, a networking infrastructure that connects computers globally.

The Web is a way of accessing information over the medium of the Internet, an information sharing model that is built on top of the Internet.

The Web is just one of the ways that information can be disseminated over the Internet but it is the one we are focused on in this class.
The technologies discussed in this class are not all established standards. They are mostly agreed upon but not evenly implemented. Most modern browsers support everything we will do, but not everyone uses up-to-date, modern browsers (and that’s ok). Web graphics technologies are, as ever, in a state of flux.
Drawing on the Web
CSCI-UA 380

Class 1
Introduction and Overview
Simple Net Art Diagram

The art happens here

MTAA ca. 1997

http://www.mtaa.net/mtaaRR/off-line_art/snad.html
Drawing on the Web
CSCI-UA 380

HTML
HyperText Markup Language
HTML

HyperText Markup Language

A language for describing Web pages

HTML is not a programming language, it is a markup language

JavaScript is a programming language

A markup language is a set of markup tags

HTML uses markup tags to describe Web pages
HTML Tag
Opening/Closing

Keywords surrounded by angle brackets, for example: `<html>`

HTML tags normally come in pairs, like `<h1>` and `</h1>`

The first tag in a pair is the “start tag,” the second tag is the “end tag”

Start and end tags are also called “opening” and “closing” tags
HTML Element
Everything from the start tag to the end tag

Example:
<p>This is a paragraph.</p>

Start tag:
<p>

Element content:
This is a paragraph.

End tag:
</p>
Web Pages

HTML Documents

Web pages are plain text documents that contain HTML tags

HTML documents are Web pages

Recommended plain text editors: Atom, Sublime Text, Notepad++
Web Browsers

Render HTML Documents

Reads HTML documents and displays them as Web pages

Web browsers do not display HTML tags, but use them to interpret the content of the page

This is where most of our drawing will happen

Recommended browsers:
Chrome, Safari, Firefox
FTP
File Transfer Protocol

FTP, like HTTP, is a communications protocol of the Internet.

FTP allows us to transfer files between computers (i.e. laptop to Web server).

It requires a secure FTP connection, known as SFTP.

Recommended SFTP clients: Fetch, Cyberduck, WinSCP, Transmit.
**File Permissions**

**chmod**

Standard file permission: 644
Owner can read and write file; group can read file; others can read file

Standard directory permission: 755
Owner can read, write and execute file; group can read and execute file; others can read and execute file
HTML5
New standard for HTML

First version published in 2008
An official W3C recommendation as of October 2014

• Simpler doctype declaration
• New elements
• New attributes
• Full CSS3 support
• Video and audio elements
• 3D graphics support
• Web applications
Drawing on the Web
CSCI-UA 380

CSS
Cascading Style Sheets
CSS
Cascading Style Sheets

Defines a Web page’s appearance

CSS separates style and content

Consists of a plain text file with rules for the display of HTML elements

Formatting includes fonts and colors as well as layout and position

Can be created outside of your HTML and applied to multiple Web pages

Well-formed HTML is important for your CSS to work properly
CSS can be applied in three different ways to a Web page:

- In an external .css file
- In the <head> section of an HTML document
- Inline with HTML code
Style Construction

Selector: Indicates which HTML element will be formatted

Declaration block: Describes the formatting to apply

Property/value pair: Specifies format

Style rules are separated by a ;

```css
h1 {
    color: green;
    background: yellow;
}
```
The principle of the “cascade” is applied when style rules are in conflict. Three primary factors determine which style rule wins out:

- Inheritance
- Specificity
- Location
CSS
Cascading Style Sheets

CSS3

Latest standard for CSS

CSS2 is best supported

CSS3 is still evolving but offers new features for designers and developers

Modern browsers support many aspects of CSS3

Backwards compatible with CSS2
CSS Box Model
Setting an Element’s Dimensions

• **Auto**
  Browser calculates the width

• **Length**
  Defines the width in a unit like pixels

• **%**
  Defines the width in percent of the containing block

• **Inherit**
  Specifies that the value should be inherited from the parent element
Responsive Web Design
Responsive Web Design

Media Queries

Features you can include in media queries: • width • height • device-width • device-height • orientation • aspect-ratio • device-aspect-ratio • color • color-index • monochrome • resolution • scan grid

Most of the above can be combined with min- and max- prefixes

Most common are min-width and max-width prefixes

Media queries can be used to load an alternate style sheet or offer alternate styles within an existing style sheet.
Responsive Web Design

Media Query Syntax

Two possible values: only or not

only screens out older browsers from reading the rest of the query

not negates the result: not screen means everything except screen

type is the media type

feature: value

Enclosed by parentheses and preceded by the word, and

Predefined media features

Multiple features and values can also be combined with and
Basic Style Sheet Link

<link rel="stylesheet" href="main.css">

Style Sheet Link with Media Query

<link rel="stylesheet" media="only screen and (min-width: 640px)" href="main.css">
Basic CSS Rule Set

body {background-color: orange;}

CSS Rule Set with @media Rule

@media only screen and (min-width: 480px) {
  body {background-color: orange;}
}

Drawing on the Web
CSCI-UA 380
CSS
Cascading Style Sheets
Drawing on the Web
CSCI-UA 380

SVG
Scalable Vector Graphics
SVG
Scalable Vector Graphics

SVG is a language for describing two-dimensional graphics in XML.

SVG allows for three types of graphic objects: vector graphic shapes, images, and text.

SVG drawings can be interactive, dynamic, and animated.

Scalable Vector Graphics (SVG) 1.1
Introduction
www.w3.org/TR/SVG/
To be scalable means to increase or decrease uniformly
In terms of graphics, means not being limited to a single, fixed, pixel size
On the Web, scalable means that a particular technology can grow
SVG is scalable in both senses of the word

Scalable Vector Graphics (SVG) 1.1 Concepts
http://www.w3.org/TR/SVG/concepts.html
Vector graphics contain geometric objects such as lines and curves. This gives greater flexibility compared to raster-only formats.

Since all modern displays are raster-oriented, the difference between raster-only and vector graphics comes down to where they are rasterized.

Vector graphics are rasterized client-side; raster graphics are, by nature, already rasterized on the server.
Most existing XML grammars represent either textual information or raw data. They typically provide only rudimentary graphical capabilities. SVG provides a rich, structured description of vector and mixed vector/raster graphics.

Scalable Vector Graphics (SVG) 1.1 Concepts
http://www.w3.org/TR/SVG/concepts.html
**SVG**

**Advantages**

- SVG images can be created and edited with any text editor
- SVG images can be searched, indexed, scripted, and compressed
- SVG images are scalable, can be printed at any resolution, and are zoomable without degradation
- SVG is an open standard
- SVG files are pure XML

SVG Introduction
http://www.w3schools.com/svg/svg_intro.asp
SVG
XML

XML stands for Extensible Markup Language

XML is a markup language much like HTML

XML was designed to carry data, not to display data

XML tags are not predefined. You must define your own tags

XML is designed to be self-descriptive

SVG is written in XML

Introduction to XML
http://www.w3schools.com/xml/xml_whatis.asp
The advantages of style sheets are now generally accepted, certainly for use with text.

SVG extends this control to the realm of graphics.

It allows for script-based manipulation of the document tree and the style sheet.

Scalable Vector Graphics (SVG) 1.1 Concepts
http://www.w3.org/TR/SVG/concepts.html
SVG Animation

SVG supports the ability to change vector graphics over time.

Using SVG’s animation elements, you can define motion paths, fade-in or fade-out effects, and objects that grow, shrink, spin or change color.

SVG has been designed to allow SMIL to use animated or static SVG content as media components.

Scalable Vector Graphics (SVG) 1.1 Concepts
www.w3.org/TR/SVG/animate.html
There are a variety of ways in which SVG content can be included within a Web page

- A stand-alone SVG Web page
- Embedding by reference, using the HTML `img` element
- Embedding inline
- External link, using the HTML `a` element
- Referenced from a CSS property

Scalable Vector Graphics (SVG) 1.1 Concepts
http://www.w3.org/TR/SVG/concepts.html
CSS3
Transforms, Transitions, and Animation
CSS3 Transforms
2D Transforms

With CSS3 transforms, we can move, scale, turn, spin, and stretch elements.

A transform is an effect that lets an element change shape, size and position.

Chrome (36), Firefox (16), IE (10), Opera (23), and Safari (9) support 2D transforms.

2D transform methods include: translate, rotate, scale, skew, matrix.

W3Schools
CSS3 2D Transforms
www.w3schools.com/css/css3_2dtransforms.asp
CSS3 Transforms
3D Transforms

CSS3 allows you to format your elements using 3D transforms.

Chrome (36), Firefox (16), IE (10), Opera (23), and Safari (9) support 3D transforms.

With 3D transforms, an element can rotate around an X-, Y-, or Z-axis at a given degree.

3D transform methods include: translate, rotate, scale, and matrix.

W3Schools
CSS3 3D Transforms
www.w3schools.com/css/css3_3dtransforms.asp
CSS3 Transforms
Properties

transform
transform-origin
transform-style
perspective
perspective-origin
backface-visibility

W3Schools
CSS3 3D Transforms
www.w3schools.com/css/css3_3dtransforms.asp
CSS3 Transitions

CSS3 transitions are effects that let an element gradually change from one style to another.

Chrome (26), Firefox (16), IE (10), Opera (12.1), and Safari (6.1) support the transition property.

To apply a transition, you must specify two things:

• The CSS property you want to add an effect to
• The duration of the effect

W3Schools
CSS3 Transitions
www.w3schools.com/css/css3_transitions.asp
CSS3 Transitions
Properties

- transition
- transition-property
- transition-duration
- transition-timing-function
- transition-delay

W3Schools
CSS3 Transitions
www.w3schools.com/css/css3_transitions.asp
With CSS3, we can create animations in Web pages.

The `@keyframes` rule is where the animation is created.

Specify a CSS style inside the `@keyframes` rule and the animation will gradually change from the current style to the new style.

Chrome (43), Firefox (16), IE (10), Opera (30), and Safari (9) support the `@keyframes` rule and animation property.
CSS3 Animation Properties

@keyframes
animation
animation-name
animation-duration
animation-timing-function
animation-delay
animation-iteration-count
animation-direction
animation-play-state

W3Schools CSS3 Animations
www.w3schools.com/css/css3Animations.asp
CSS3

Vendor Prefixes

-webkit-
-moz-
-ms-
-o-

The purpose of vendor prefixes was to allow browser-makers to support experimental CSS declarations.

Even if details were not standardized, a new property could be implemented.

All of the experimental features we explore in this class are standardized in the latest browsers.
Version Control

A system that records changes to a file or set of files over time so that you can recall specific versions later.

Commonly used for software source code but any type of file can be placed under version control.

A Version Control System (VCS) allows you to:
- Revert files back to a previous state
- Review changes made over time
- Collaborate more efficiently
- Maintain project backups
Version Control

Relevance

The creative process is often an iterative one, with ideas and forms leading to new ideas and forms.

A primary characteristic of Web aesthetics is the remix of existing material.

Creative code introduces both new challenges and new opportunities for keeping track of project versions.

A well-executed system is good for archiving, iteration, and collaboration.
Drawing on the Web
CSCI-UA 380

Version Control
Project Management with Git

my-project-1.txt

my-project-2.txt

my-project-3.txt
Centralized Version Control Systems were developed to allow collaboration with developers on other systems.

With a CVCS, a single server contains all the versioned files and clients “check out” files from that central place.

For many years, this has been the standard for version control.

The downside of centralized version control is the vulnerability of having the entire history of a project in one place.
With Distributed Version Control Systems, clients don’t just check out the latest snapshot of files, they fully mirror the entire history of the project. If a server dies, anyone with a copy of all the versioned files can restore it to the server. Every checkout is really a full backup of all the data. You can also collaborate with different groups of people in different ways simultaneously within the same project.
Git

History

Git was created by Linus Torvalds and the Linux development community for Linux kernel maintenance.

Linux is an open source operating system project of fairly large scope.

Git was born in 2005 after the breakdown of a relationship with the company that maintained Linux’s former version control system.

Its goal was to be a fully distributed VCS with a simple design, support for non-linear development, and the ability to handle large projects efficiently.

Pro Git
Scott Chacon
git-scm.com/book
Git Basics

Git thinks of its data like a set of snapshots of a mini file system.

Every time you save the state of your project, it basically takes a picture of what all your files look like then and stores a reference to that snapshot.

To be efficient, if files have not changed, Git doesn’t store the file again—just a link to the previous identical file it has already stored.

This makes Git more like a mini file system with some powerful tools built on top of it.
Git

Three States

Git has three main states that your files can reside in: committed, modified, and staged.

Committed means that the data is safely stored in your local database.

Modified means that you have changed the file but have not committed it to your database yet.

Staged means that you have marked a modified file in its current version to go into your next commit snapshot.

Pro Git
Scott Chacon
git-scm.com/book
Local Operations

- working directory
- staging area
- git directory (repository)

- checkout the project
- stage files
- commit

Pro Git
Scott Chacon
git-scm.com/book
Git

Workflow

1. Modify files in your working directory
2. Stage the files, adding snapshots of them to your staging area
3. Commit changes, which takes the files as they are in the staging area and stores that snapshot permanently to your Git directory
GitHub is a web-based hosting service that uses the Git VCS.

The site also provides social networking functionality such as feeds, followers, wikis, and statistics.

The company was founded in 2008 and is located in San Francisco.

In addition to computer programmers, architects, musicians, municipal governments, and academics are among its users.
You can think of a web page as consisting of three layers: structure, presentation, and behavior.

- HTML is the structure layer
- CSS is the presentation layer
- JavaScript is the behavior layer

JavaScript is a programming language for creating interactivity and functionality in web browsers.
JavaSript

Background

JavaScript was introduced by Netscape in 1995.

At that time, the Java language was ascendant and the name “JavaScript” was an attempt to ride this popularity.

Eventually, browsers other than Netscape began to support JavaScript functionality, calling it “ECMAScript”.

Today, JavaScript is not only a lingua franca of the Web but a basis for many other computational media projects.
As with CSS, JavaScript targets HTML elements to do something with them.

There are three ways you can apply JavaScript to HTML:

- Inline JavaScript
- Embedded JavaScript
- External JavaScript

External and embedded JavaScript are preferable for their separation of content and behavior.
Like HTML and CSS, JavaScript is rendered in the web browser.

Because it’s rendered in the browser rather than on a server, JavaScript is considered a “front-end language”.

A browser’s “rendering engine” does the work of what you see on screen.

There are different rendering engines for different browsers—this is why we sometimes use CSS vendor prefixes.
# Rendering Engines

<table>
<thead>
<tr>
<th>Browser</th>
<th>Engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome</td>
<td>Blink</td>
</tr>
<tr>
<td>Firefox</td>
<td>Gecko</td>
</tr>
<tr>
<td>Internet Explorer</td>
<td>Trident</td>
</tr>
<tr>
<td>Opera</td>
<td>Blink</td>
</tr>
<tr>
<td>Safari</td>
<td>Webkit</td>
</tr>
</tbody>
</table>
### JavaScript Engines

<table>
<thead>
<tr>
<th>Browser</th>
<th>Engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome</td>
<td>V8</td>
</tr>
<tr>
<td>Firefox</td>
<td>Spidermonkey</td>
</tr>
<tr>
<td>IE and Edge</td>
<td>Chakra</td>
</tr>
<tr>
<td>Opera</td>
<td>V8</td>
</tr>
<tr>
<td>Safari</td>
<td>Nitro</td>
</tr>
</tbody>
</table>
JavaScript
Capability

Computationally speaking, there isn’t much JavaScript can’t do; it’s a robust programming language.

Core functionality includes modifying HTML and CSS, communicating with the server, and storing data.

We will use JavaScript and associated libraries for interactivity, drawing on the HTML5 canvas, and rendering in 3D.

As with any technology, it’s good to consider when to—and not to—use it.
<canvas>
HTML5 Canvas

The HTML5 canvas tag is used to draw raster graphics in the browser using JavaScript.

With the canvas, you can illustrate, make photo compositions, and create animation.

Canvas was first introduced by Apple for the Mac OS X Dashboard and later implemented in Safari and Chrome.

The canvas is supported by all major web browsers.
**HTML5 Canvas**

Create

A canvas is a rectangular area on an HTML page, and it is specified with the `<canvas>` element.

By default, the canvas element has no border and no content.

Always specify an ID attribute as well as width and height attributes to define the size of the canvas.

You can have multiple canvas elements on one HTML page.

W3Schools: HTML5 Canvas

www.w3schools.com/html/html5_canvas.asp
All drawing to the canvas must be done with JavaScript.

First, reference the canvas by its ID.

Then, call its `getContext()` method and pass the string, “2d” to it.

The canvas is a two-dimensional grid whose upper-left corner has coordinate (0, 0).
HTML5 Canvas Methods

Canvas only supports one primitive shape: rectangles

All other shapes must be created by combining one or more paths.

There are a variety of methods which make it possible to compose shapes

These include methods for drawing lines, arcs, text, color, and images

W3Schools: HTML5 Canvas
www.w3schools.com/html/html5_canvas.asp
HTML5 Canvas

p5.js

A JS client-side library for creating graphic and interactive experiences

Based on the core principles of Processing

p5.js has a full set of drawing functionality for the HTML canvas

It also has add-on libraries for text, input, video, webcam, and sound

p5.js was originally created by Lauren McCarthy in 2013 and its ongoing development is collaborative

www.p5js.org
Drawing on the Web
CSCI-UA 380

WebGL
Drawing in 3D with Three.js
3D Graphics

We experience the world in three dimensions and, increasingly, screen-based media is rendered in 3D.

3D graphics have been around since the 1960s but required advanced computers with special software.

Today, 3D processing hardware is integrated with virtually every new computer and mobile device.

Moreover, software for rendering 3D is now ubiquitous: the web browser.
3D Graphics

Coordinate System

3D graphic data is represented in a Cartesian coordinate system.

In addition to “x” and “y” values, an additional “z” value describes depth.

The primary difference between 2D Canvas (and CSS) coordinates and WebGL is that “y” starts at the bottom of the window, increasing to the top.

This distinction stems from historical convention of WebGL being y-up and HTML being y-down.
3D Graphics

Mesh

The most common way to draw 3D graphics is with a mesh.

A mesh is composed of one or more polygon shapes, constructed of vertices \((x, y, z)\), defining coordinate positions in space.

The polygons most often used in meshes are triangles and rectangles.

3D meshes are often referred to as “models”.

Programming 3D Applications with HTML5 and WebGL
Tony Parisi
Image source:
wikipedia.org/wiki/File:Dolphin_triangle_mesh.png
Material is generally used to describe the surface attributes of a mesh. This can be as simple as a solid color or more complex, such as a shiny, reflective surface. Surface information can also be applied using bitmap images to create textures.
3D Graphics

Light

Light defines how a surface is illuminated.

Without a light source, it is difficult to perceive 3D attributes such as depth.

3D graphics can have one or more light sources in a given scene.
3D Graphics

Camera

3D scenes require a point of view from which to experience them. The “camera” defines where, relative to the scene, a viewer is positioned. Additional camera properties include field of view, which defines perspective. The final 3D image is rendered into a 2D “viewport”—the window or canvas.
Drawing on the Web
CSCI-UA 380

WebGL
Drawing in 3D with Three.js

Image source: obviam.net
3D Graphics

Shader

Your computer’s graphics hardware understands vertices and textures, but that’s about it.

Other aspects of a 3D scene—material, light, cameras—need to be interpreted.

A “shader” is the part of your program that gets the pixels for a mesh onto the screen.

Shaders are very powerful but we will rely on a JavaScript library for this part.
WebGL is the standard 3D graphics API for the Web, initiated by Mozilla engineer, Vladimir Vukićević, in 2006. It allows developers to utilize the power of a computer’s 3D rendering hardware from within the browser, using JavaScript.

WebGL is supported in all current, major, desktop browsers; increasingly, WebGL is also supported in mobile browsers. WebGL is a low-level drawing API.
Three.js is a JavaScript toolkit for WebGL that provides higher-level access to the API.

It was initially released via GitHub in 2010 by Ricardo Cabello, also known as “Mr.doob” and is actively maintained with additional three.js authors.

Three.js is governed with an MIT free software license for use and reuse within diverse types of projects.

threejs.org
Final Exam
Thursday, May 12, 4:00–5:50 p.m.
Room 317, Warren Weaver Hall