Final Exam
Tuesday, May 15, 2:00–3:50 p.m.
Room 312, Warren Weaver Hall
Graphics on the Web
Graphics on the Web

Raster Graphics

GIF: Graphics Interchange Format
JPEG: Joint Photographic Experts Group
PNG: Portable Network Graphics
HTML5 Canvas (renders directly)
WebP: A new, lossless and lossy image compression format for the Web
Graphics on the Web
Vector Graphics

Flash (SWF files)
SVG: Scalable Vector Graphics
CSS (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.
Graphics on the Web
State of Flux

“The future is already here—it’s just not evenly distributed.”
—William Gibson
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.
Class 1
Introduction and Overview
Net Art

“Net art, like Process Art, Performance art and Happenings, is less an object for contemplation than an event or action that takes place over time.”

—Mark Tribe/Reena Jana
Simple Net Art Diagram

MTAA ca. 1997

The art happens here

http://www.mtaa.net/mtaaRR/off-line_art/snad.html
Net Art

Examples

Misconfigured ASCII Drawing 1995, Jodi

http://wwwwwwwwwww.jodi.org/
Net Art
Examples

Fukushima Sequence
2003/2015, Joshua Clayton

http://sequence.joshuaclayton.net/
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’s 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:

```html
<p>This is a paragraph.</p>
```

Start tag:

```html
<p>
```

Element content:

This is a paragraph.

End tag:

```html
</p>
```
Web Pages

HTML Documents

Web pages are plain text documents that contain HTML tags.

HTML documents are Web pages.

Recommended plain text editors:
Sublime Text, Atom, Brackets
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: Firefox, Chrome, Safari, Edge
This is where we will publish our projects during the semester.

A server is just a computer that is always on, ready to serve files when they are requested.

i6 is a secure, Unix-based Web server.

You will receive an email with your account information (if you don’t already have one).
i6 URL
i6.cims.nyu.edu/~netid
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)

i6 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
Net Art

Examples

All HTML
2011, Evan Roth

http://all-html.net/
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
CSS

Rule Set

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 semicolon

```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
In a web page, every element is rendered as a rectangular box. This box includes the following, changeable properties:

- Content
- Padding
- Border
- Margin
CSS
Cascading Style Sheets
There are two types of length units in CSS, relative and absolute.

Relative units of length include:
• `em` (relative to font size)
• `%` (relative to the containing element)

Absolute units of length include:
• `px` (pixels)

Alternatively specifications:
• `auto` (browser calculates length)
• `inherit` (from the parent element)
Elements in HTML are primarily “inline” or “block” elements.

- An inline element allows content to flow around its left and right sides.
- A block element fills the entire line and nothing is displayed on its left or right side.

The CSS display property allows you to specify the type of box used for an HTML element.
Page Layout

There are several ways to design the layout of a web page with CSS.

- CSS float property
- CSS positioning
- CSS flexible box
CSS Float Property

The CSS float property allows you to position block elements inline. This means that any element, block or inline, can be positioned alongside another element. The CSS float property is one of the main techniques of web page layout.
The CSS position property specifies the type of positioning used for an element on a page.

- **static**: Elements are rendered in order, as they appear in the document flow (this is the default).
- **absolute**: Element is positioned relative to its first positioned (not static) parent element.
- **fixed**: Element is positioned relative to the browser window.
- **relative**: Element is positioned relative to its normal position.
Flexible box, or flexbox, is a new layout mode in CSS3 that is becoming increasingly common on web pages.

Flexbox consists of flexible containers and flexible items within.

A flex container expands items to fill available free space or shrinks them to prevent overflow.

In practice, flexbox can accommodate different screen sizes and different display devices more easily than the CSS float property.
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
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.
SVG

Scalable

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
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.

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

Support for SMIL animation may be waning in favor of CSS animation.

Scalable Vector Graphics (SVG) 1.1
Concepts
www.w3.org/TR/SVG/animate.html
SVG
On the Web

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

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
CSS3 Animation

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.

W3Schools
CSS3 Animations
www.w3schools.com/css/css3_animations.asp
CSS3 Animation
Properties

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

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

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.
Net Art
Examples

into time .com
2010, Rafaël Rozendaal
http://www.intotime.com
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

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

Centralized VCS

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.
Drawing on the Web
CSCI-UA 380

Version Control
Project Management with Git

Pro Git
Scott Chacon
git-scm.com/book
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.
Version Control
Project Management with Git

Pro Git
Scott Chacon
git-scm.com/book
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.
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: modified, staged, and committed.

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.

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

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

Pro Git
Scott Chacon
git-scm.com/book
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.
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.
JavaScript
Front-End Language

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 as well as different JavaScript engines.
## 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 for web development.

We will use JavaScript and associated libraries for interactivity, animation, 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>
HTML Canvas

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

With the canvas, one can illustrate, make photo compositions, and create animation dynamically on the Web.

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

Today, the canvas is supported by all major web browsers.
HTML Canvas

Setup

The HTML canvas is a rectangular area on a web page, specified by the `<canvas>` element.

Typically, an id is specified along with width and height attributes to define the size of the canvas.

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

One HTML page can include multiple canvas elements.
All drawing to the canvas must be done with JavaScript.

Begin by referencing the canvas with a DOM query.

Then, use its `getContext()` method to specify a two-dimensional drawing.

The canvas is a Cartesian grid whose upper left corner has coordinate (0, 0).
**HTML Canvas**

**Methods**

Canvas only supports one primitive shape: rectangle

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

There are a variety of path methods in the canvas API which make it possible to compose shapes

These include methods for drawing line, ellipse, arc, text, color, and image
Since JavaScript is used to control the canvas, it's straightforward to use it for animation as well.

The basic steps are as follows:
1. Clear the canvas
2. Save the canvas state
3. Draw animated shapes
4. Restore the canvas state

The `requestAnimationFrame()` method is optimal for canvas animation.
HTML Canvas

Considerations

The canvas API opens up all sorts of possibilities for raster graphics in the web browser.

Note, however, that pixel data drawn to the canvas are not DOM elements.

That means regions of the canvas cannot be interacted with in the same way regions of an SVG image can.

Keep in mind how these technologies intersect as well as how they diverge.
Drawing on the Web
CSCI-UA 380

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

Material

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
Tuesday, May 15, 2:00–3:50 p.m.
Room 312, Warren Weaver Hall