Where do I Draw?
(coordinate systems and 2D Transforms)

CS559 – Spring 2017
Lecture 2
January 19th, 2017
Key Idea #1

Work in convenient coordinate systems. Use transformations to get from where you want to be to where you need to be. Hierarchical modeling lets us build things out of pieces.
Today:
Let’s Draw a Line!
<html>
  <head>
    <meta name="description" content="Draw A Line"/>
  </head>
  <body>
    <canvas id="myCanvas" width="200" height="200"
      style="border:1px solid #000;">\</canvas>
  </body>
</html>

```javascript
var canvas = document.getElementById('myCanvas');
var context = canvas.getContext('2d');

// a fat purple line
context.lineWidth = 5;
context.strokeStyle = "#C08";

// the actual line
context.beginPath();
context.moveTo(50,50);
context.lineTo(100,100);
context.stroke();
```

http://jsbin.com/satunaromo/edit
Why this?

A little web programming
so you can do the assignment

Exposure to a graphics library (API)

Some important graphics concepts
What does this program do?
What does this program do?

It draws a line.
What does this program do?

It draws a thick red diagonal line.
What does this program do?

It puts a drawing space on the page called a canvas.

It causes a bunch of pixels to be changed from the background color to purple.
The HTML
You need some

Make a **canvas**
Give it an **id**

Not really graphics…
JSBin...
Sticks the JavaScript into the HTML body

<!DOCTYPE html>
<html>
<head>
  <meta name="description" content="Draw A Line">
  <canvas id="myCanvas" width="200" height="200"
    style="border:1px solid #000;"> </canvas>
</head>
<body>
  var canvas = document.getElementById('myCanvas');
  var context = canvas.getContext('2d');
  // a fat purple line
  context.lineWidth = 5;
  context.strokeStyle = "#C08";
  // the actual line
  context.beginPath();
  context.moveTo(50,50);
  context.lineTo(100,100);
  context.stroke();
The JavaScript

Warning:
read the tutorial on when code runs!
The JavaScript

Find the canvas

Get the context

Draw

```javascript
var canvas = document.getElementById('myCanvas');
var context = canvas.getContext('2d');

// a fat purple line
context.lineWidth = 5;
context.strokeStyle = "#C08";

// the actual line
context.beginPath();
context.moveTo(50,50);
context.lineTo(100,100);
context.stroke();
```
Context?
An API with state

Set the style for the next drawing commands

```javascript
var canvas = document.getElementById('myCanvas');
var context = canvas.getContext('2d');

// a fat purple line
context.lineWidth = 5;
context.strokeStyle = "#C08";

// the actual line
context.beginPath();
context.moveTo(50,50);
context.lineTo(100,100);
context.stroke();
```
Context
What pen and paper are we using?
The state needed to know how to draw
where are we drawing?
any mode information
properties to use for next commands
other info
Object oriented APIs
receieves drawing commands
Drawing – immediate mode

```javascript
var canvas =
document.getElementById('myCanvas');
var context = canvas.getContext('2d');

// a fat purple line
context.lineWidth = 5;
context.strokeStyle = '#C08';

// the actual line
context.beginPath();
context.moveTo(50, 50);
context.lineTo(100, 100);
context.stroke();
```

Drag the pen along the path

Now
Immediate vs. Retained

2 types of APIs

Immediate: draw now
Retained: store the object
gather them all up
draw them all at once
Drawing – immediate mode

The line never “exists”

Only gets turned into pixels

```javascript
1
2 var canvas =
   document.getElementById('myCanvas');
3 var context = canvas.getContext('2d');
4
5 // a fat purple line
6 context.lineWidth = 5;
7 context.strokeStyle = '#C08';
8
9 // the actual line
10 context.beginPath();
11 context.moveTo(50,50);
12 context.lineTo(100,100);
13 context.stroke();
14
15
```
Immediate Mode

Pros:
- easy – does what you want
- control over timing

Cons:
- can’t change things (need to redraw)
- can’t analyze collection
- can’t send things in bulk
Pen and Path Model

Move pen around

Stroke or Fill in current style

```javascript
var canvas = document.getElementById('myCanvas');
var context = canvas.getContext('2d');

// a fat purple line
context.lineWidth = 5;
context.strokeStyle = "#C08";

// the actual line
context.beginPath();
context.moveTo(50,50);
context.lineTo(100,100);
context.stroke();
```
Not for today...

How do we convert from primitives (the line) to pixels?

Need to determine which pixels to set to which colors.

rasterization

rendering
Coordinates and Coordinate Systems

What does 50,50 mean?

```javascript
var canvas = document.getElementById('myCanvas');
var context = canvas.getContext('2d');

// a fat purple line
context.lineWidth = 5;
context.strokeStyle = '#C08';

// the actual line
context.beginPath();
context.moveTo(50, 50);
context.lineTo(100, 100);
context.stroke();
```
Interpreting coordinates with coordinate systems

Need to know what it means

Want to use convenient numbers

Need to be able to translate between systems

Manipulate systems to move objects
A tuple (ordered list) of numbers

50,50 – is just a list of numbers

Vectors – a movement, a displacement, …
  Represented in a tuple / array

Point / position – specific place
  Represented in a tuple / array