ECS 89

5/23

Announcements

- Late assignments due tonight
- Next assignment up this weekend some time
- Will be due Tu June 3
- Will be game based on bouncing ball problem towards the end of Chapter 16 in Eloquent Javascript book.
- Follow link on class Web page to explore.

HTML5 <canvas>

- A region to draw in
- We'll use 2d version; 3d version is totally different (WebGL, a form of OpenGL)
- Insert the canvas tag, then draw into it using a Javascript progrm

```
<canvas width="400" height="400"></canvas>
```

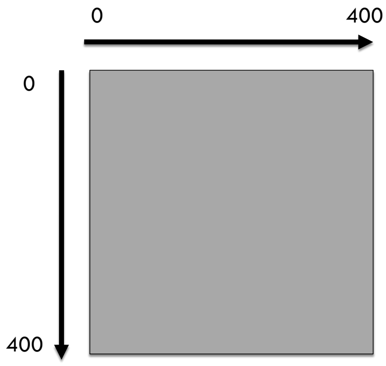
The drawing context object

```
var ctx = document.querySelector("canvas").getContext("2d");
```

- querySelector method can pick things by tag, id, or class
- Drawing context has a lot of attributes that control how things get drawn. Eg: color, rotations.
- Has a lot of methods that do actual drawing.

```
ctx.fillStyle = "rgb(160,170, 255)"; // attribute
ctx.fillRect(10, 10, 300, 300);         // method
```

A 400x400 canvas



Paths and strokes

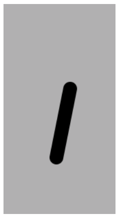
- A path is a line or a curve. Straight lines and circular arcs are the most popular.
- The path is defined, then actually drawn with the stroke method.

```
// a circle
ctx.beginPath();
ctx.arc(200, 200, 60 , 0, 2*Math.PI);
// center x, y, radius, start angle, end angle
ctx.stroke()
```

Stroke attributes

- Color is similar to css
- Line width is given in pixels
- Line caps can provide rounded corners

```
ctx.lineWidth=20;
ctx.lineCap="round";
ctx.beginPath();
ctx.moveTo(150,200);
ctx.lineTo(130,300);
ctx.stroke();
```



Fill method fills in path

- A circle drawing function

```
 function circle(cx, cy, rad) {
   ctx.beginPath();
   ctx.arc(cx,cy,rad,0,2*Math.PI);
   ctx.fill()
}
```

- Math is a built-in Javascript objects with useful attributes and methods

Use functions for drawing objects

- See code for drawing a sheep.

Moving things

- We could move the sheep by adding offsets to every x,y position when we draw it.
- But that would look pretty messy.
- And if we wanted to rotate it…oh boy.
- Instead, we can put translation and/or rotation into the drawing context, and then draw the sheep in its original position.

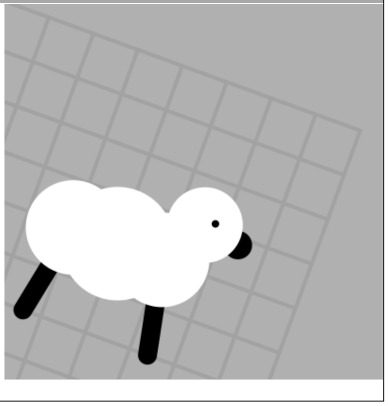
Transformations

```
ctx.save();  // save current context
ctx.translate(0,-15);  // move drawing context up
sheep(0,0)
ctx.restore()  // restore from saved context
```

- Translating y in the negative direction moves the sheep up towards zero
- Saving and restoring the context means any other objects you draw are not moved as well

Rotation

- Rotation is clockwise around (0,0)
- Often this is not what we want.



Rotation trick

- First translate (0,0) to center of sheep.
- The rotation is now around that point.
- But the sheep moved!

Rotation trick

- So then translate the sheep back.
- Ta da!