March 19, 2012

7 – 9 PM CLASS

WELCOME

sKillfer

HOST: APPNEXUS

THANK YOU

appnexus

INTRODUCTION

INSTRUCTOR

ROBYN OVERSTREET

sKillfer

WHAT IS HTML5 CANVAS?

- HTML5 element, now part of HTML5 API
- Used for drawing and animating directly in HTML, with JavaScript scripting
- Originally developed by Apple for Dashboard widgets

sKillfer

WHAT CAN IT DO?

- Dynamically draw shapes, images and text
- Respond to user input – mouse, keyboard, touch, etc.
- Animation without Flash
- Create interactive charts and graphs

sKillfer

CANVAS IN ACTION

- Rough GuidesKEXP Archive Sketchpad Wired Mind NYC Restaurant Health Ratings Map

sKillfer

BROWSER SUPPORT

- Firefox
- Chrome
- Safari
- IE 9

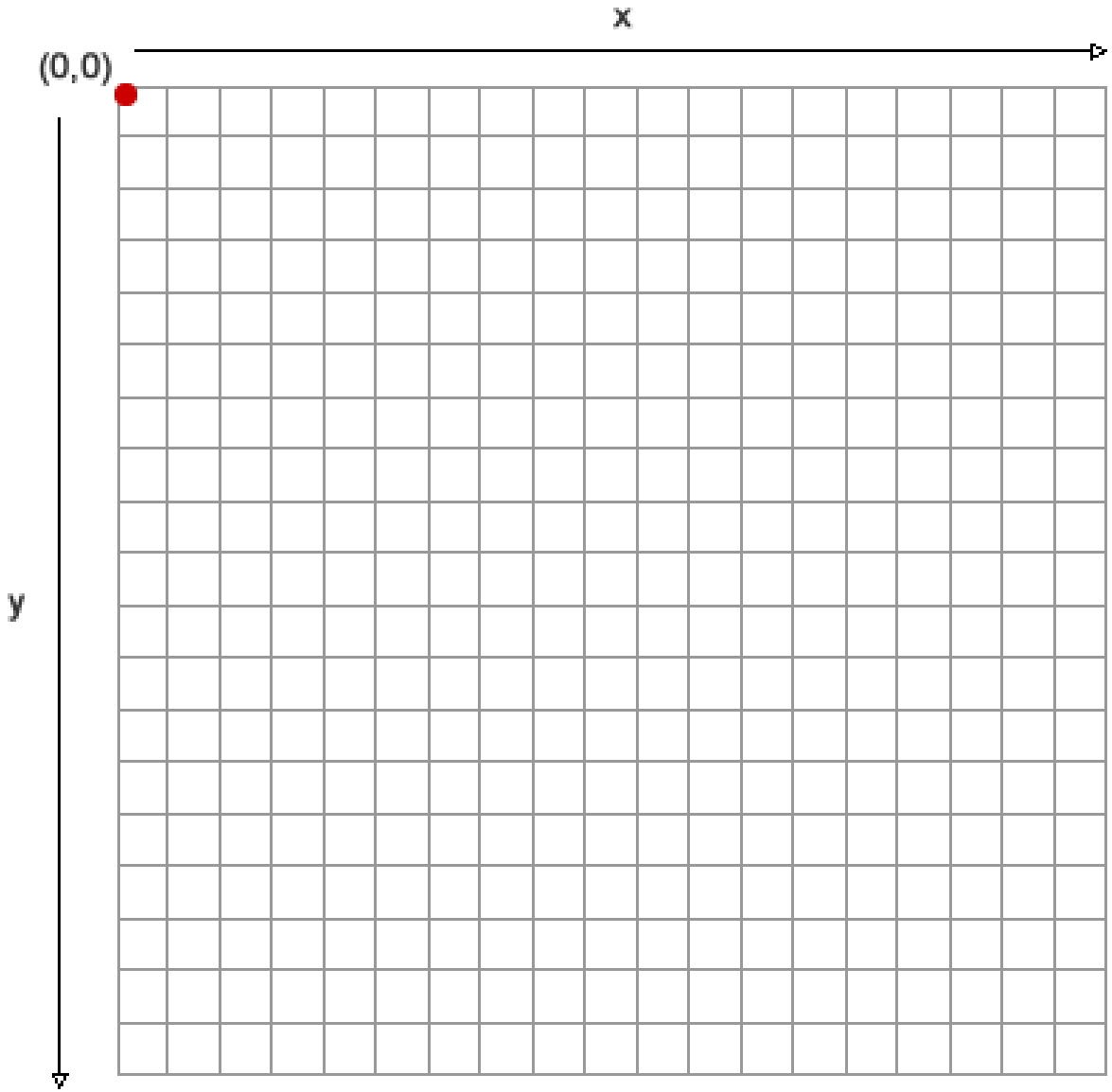
- See CanIUse.com for up-to-date support info

sKillfer

2D

DRAWING

sKillfer



THE GRID

sKillfer

SET-UP FOR DRAWING

grab the canvas element

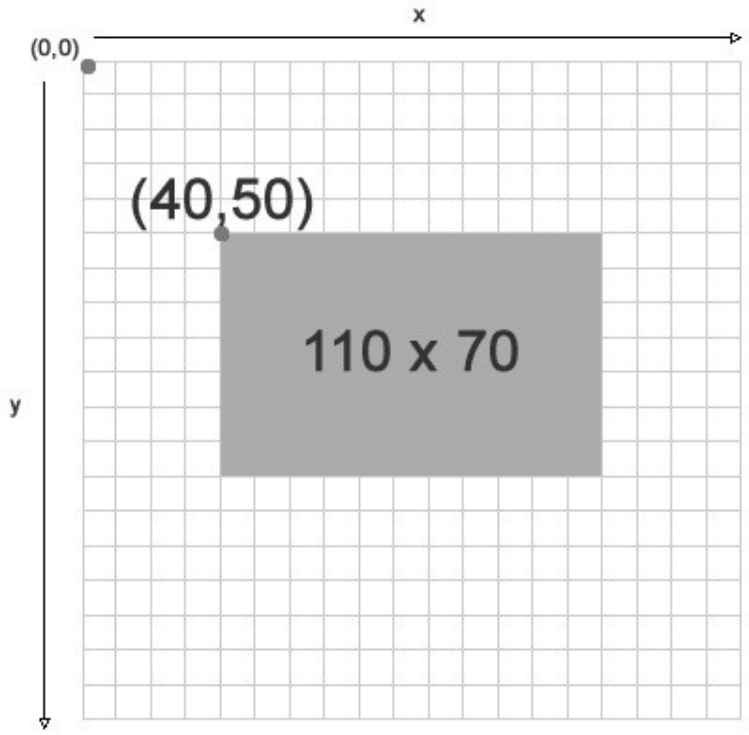
```
var mycanvas = document.getElementById("the_canvas");

var context = mycanvas.getContext("2d");
```

set up a 2D context

sKillfer

DRAWING A RECTANGLE

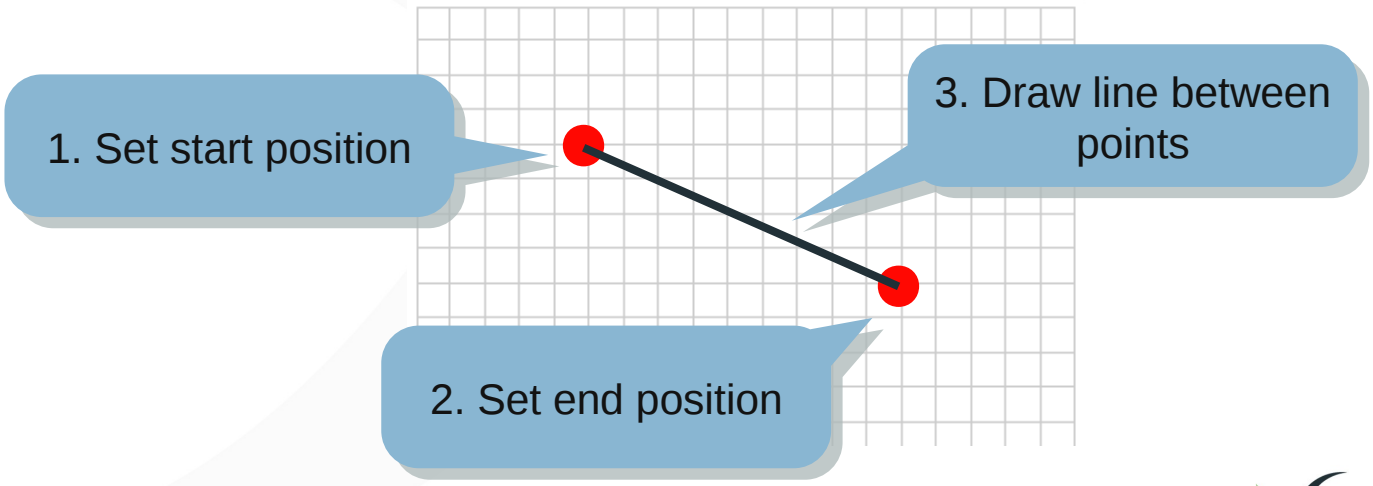
```
context.fillStyle = "rgba(0, 204, 204, 1)";

context.fillRect(40,50,110,70);
```



sKillfer

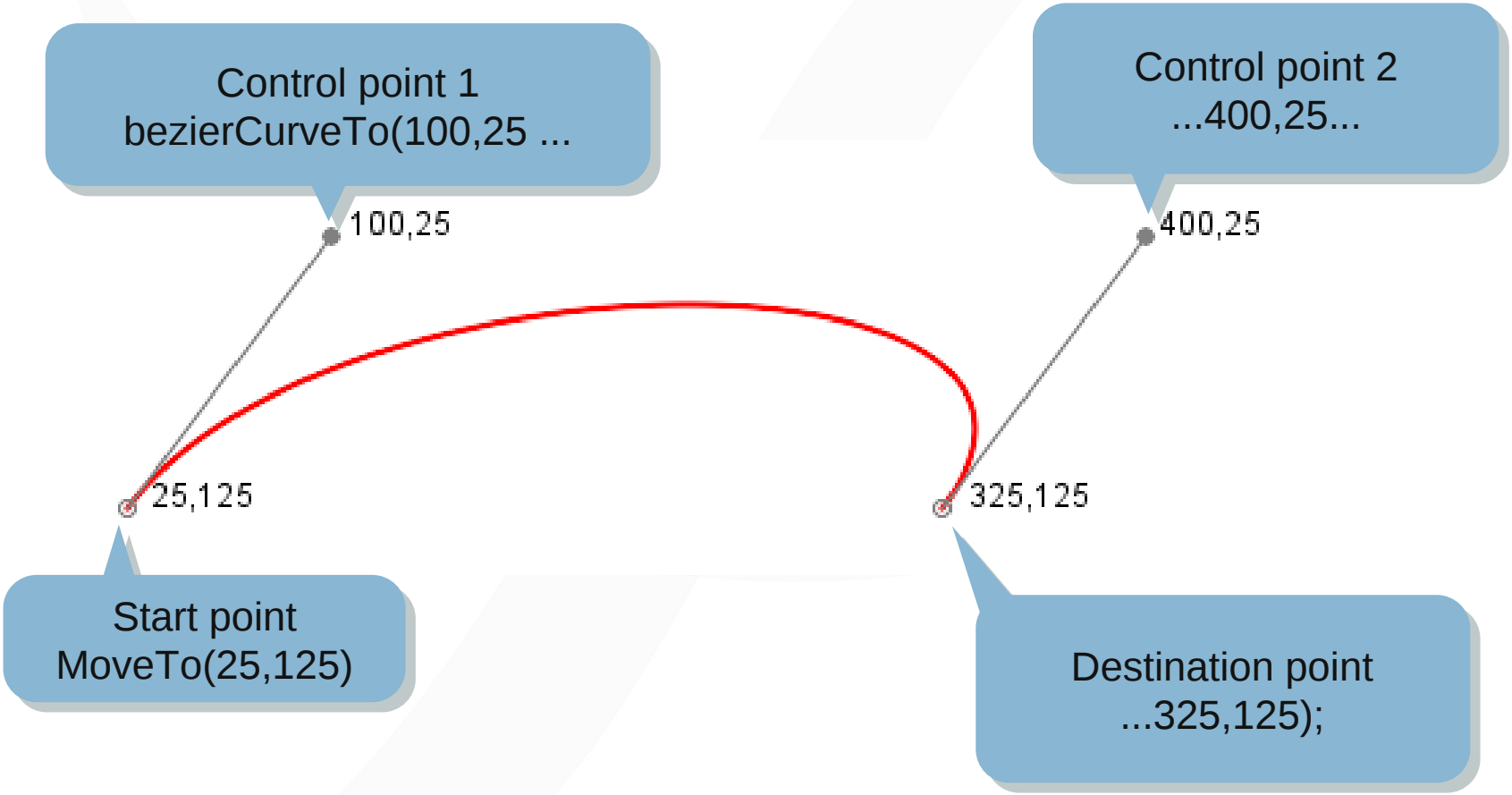
DRAWING LINES

```
context.beginPath(); //set up to draw a path
context.moveTo(x,y); //move to the start position
context.lineTo(x,y); //set the end
pointcontext.stroke(); //draw the line
```



sKillfer

DRAWING CURVES



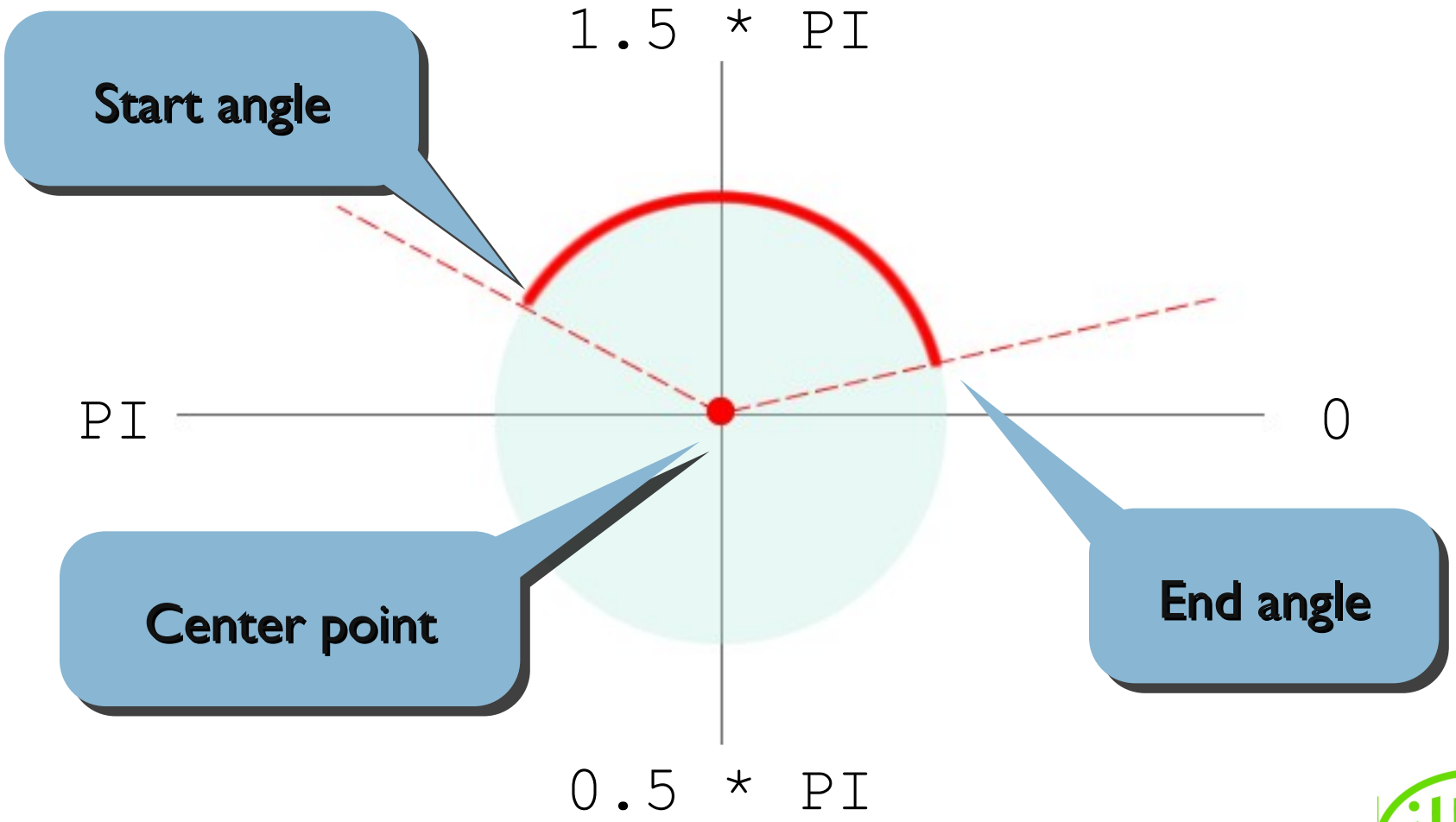
sKillfer

CODING CURVES

```
var controlPt1 = {x:110,y:30};
var controlPt2 = {x:130,y:80};
var startPt = {x:75,y:140};
ctx.beginPath();  //prepare path
ctx.moveTo(startPt.x,startPt.y);
ctx.bezierCurveTo(
      controlPt1.x,controlPt1.y,
      controlPt2.x,controlPt2.y,
      startPt.x,startPt.y
);
ctx.stroke();
```

sKillfer

DRAWING ARCS



Killfer

DRAWING ARCS & CIRCLES

- Circles are types of arcs
- Angles are in radians (need to calculate between degrees and radians)

```
ctx.beginPath();ctx.arc(x, y, radius, 0, Math.PI*2,
true);ctx.closePath();ctx.fill();
```

Start angle

End angle

sKillfer

TEXT

- Text is "drawn" to the canvas

```
context.fillText("Hello world", 10, 50);
```

- Style text in CSS syntax with .font property

```
context.font = "20pt Arial";
```

- Get the dimensions of a text area

```
textObj = ctx.measureText(d);
width = textObj.width;
```

sKillfer

TRANSFORMATIONS

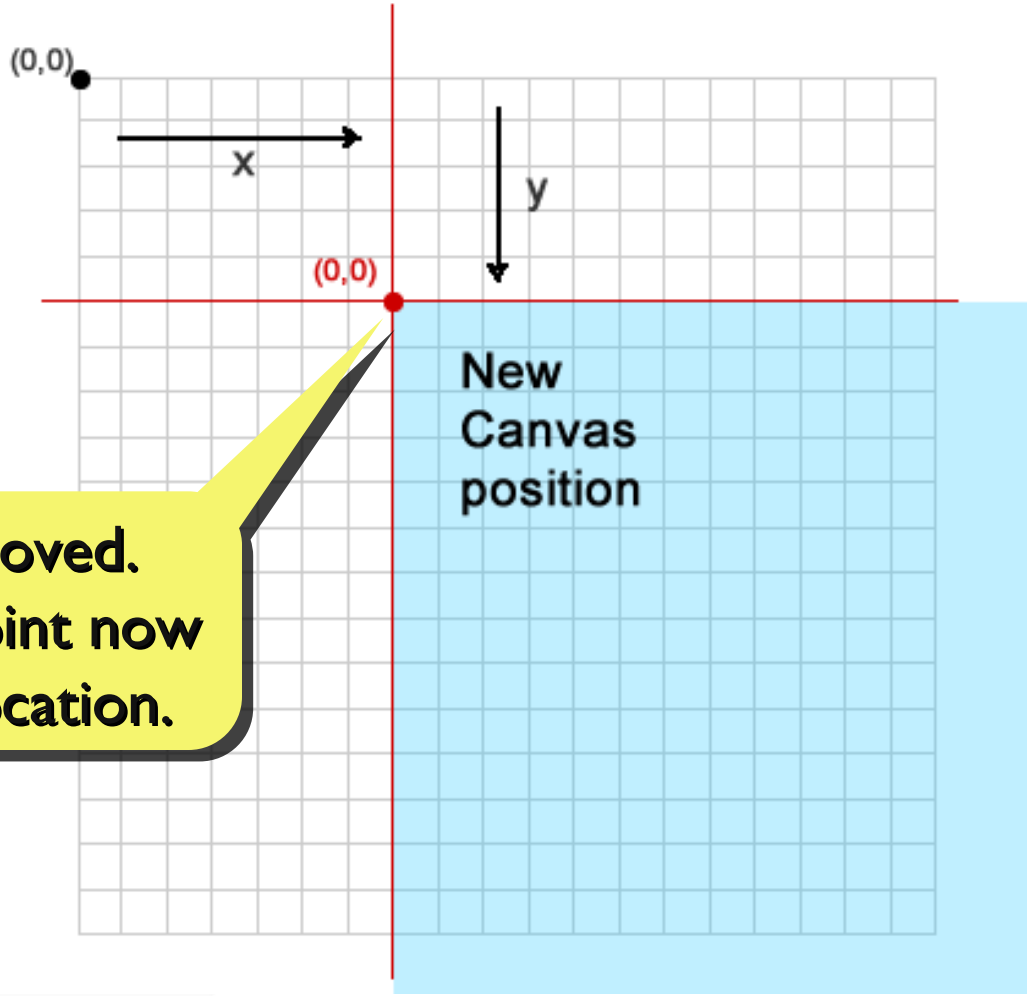
sKillfer

TRANSFORMATIONS

- To change the orientation of one element on the canvas, you must shift the entire canvas
- Translate: move the canvas and its origin to a different point in the grid
- Rotate: rotate the canvas around the current origin

sKillfer

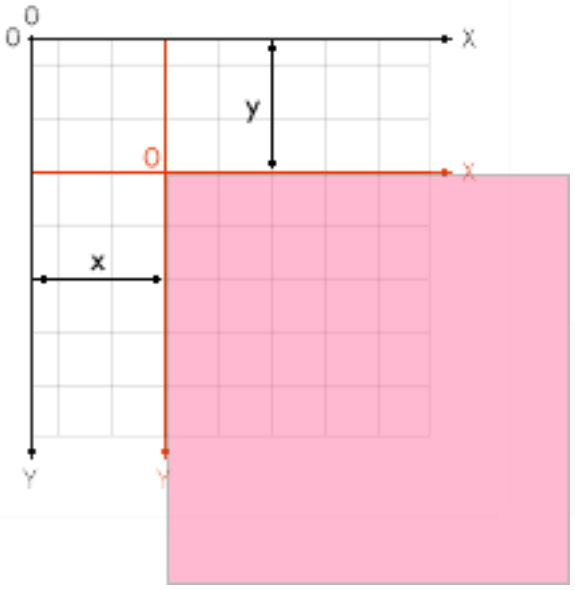
TRANSLATE CONTEXT



Context moved.
The origin point now
has a new location.

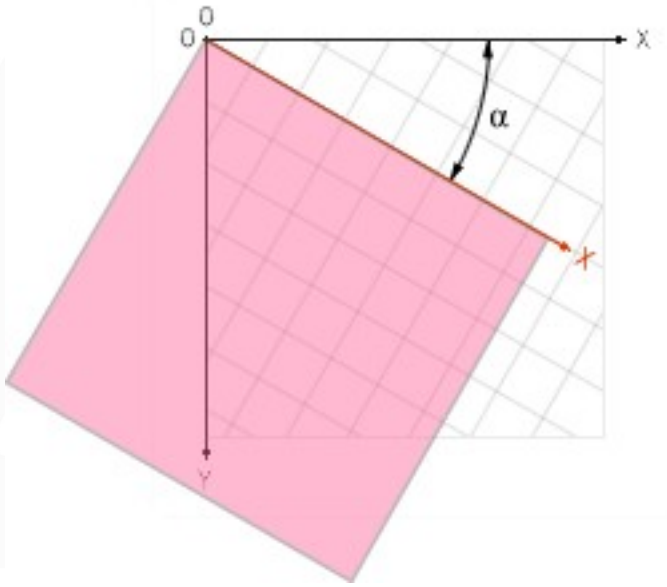
sKillfer

TRANSLATE & ROTATE



Translate

`translate(x, y)`

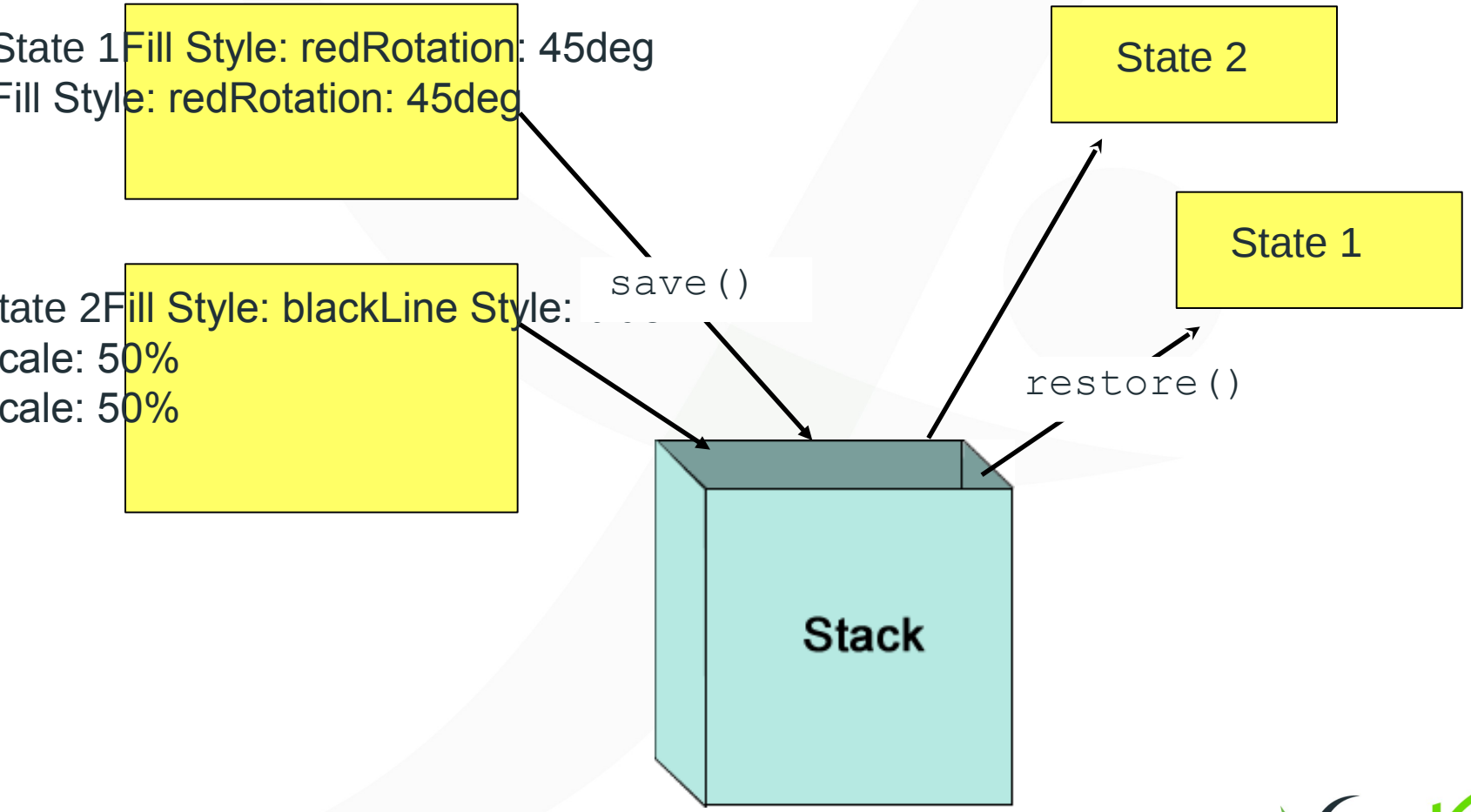


Rotate

`rotate(angle)`

sKillfer

THE STATE STACK



skillfer

SAVE & RESTORE

```
ctx.fillRect(0,0,150,150); //Draw with default settings
ctx.save(); // Save the default state    ctx.fillStyle =
'#09F'; //Change the fill color
ctx.fillRect(15,15,120,120); // Draw with new settings
ctx.save(); //Save the current state

ctx.fillStyle = '#FFF'; //Change fill again
ctx.fillRect(15,15,120,120); // Draw with new settings

ctx.restore(); //Restore to last saved state
ctx.fillRect(45,45,60,60); //Draw with restored settings
```

state 1

state 2

back to state 2

sKillfer

IMAGE

MANIPULATION

sKillfer

DRAW IMAGE

```
var img = new Image();   //Create an image objectimg.src =
'dog.png'; //Set source
img.onload = function(){ //Wait for the image to load
      context.drawImage(img,x,y); //Draw image to canvas
}
```

sKillfer

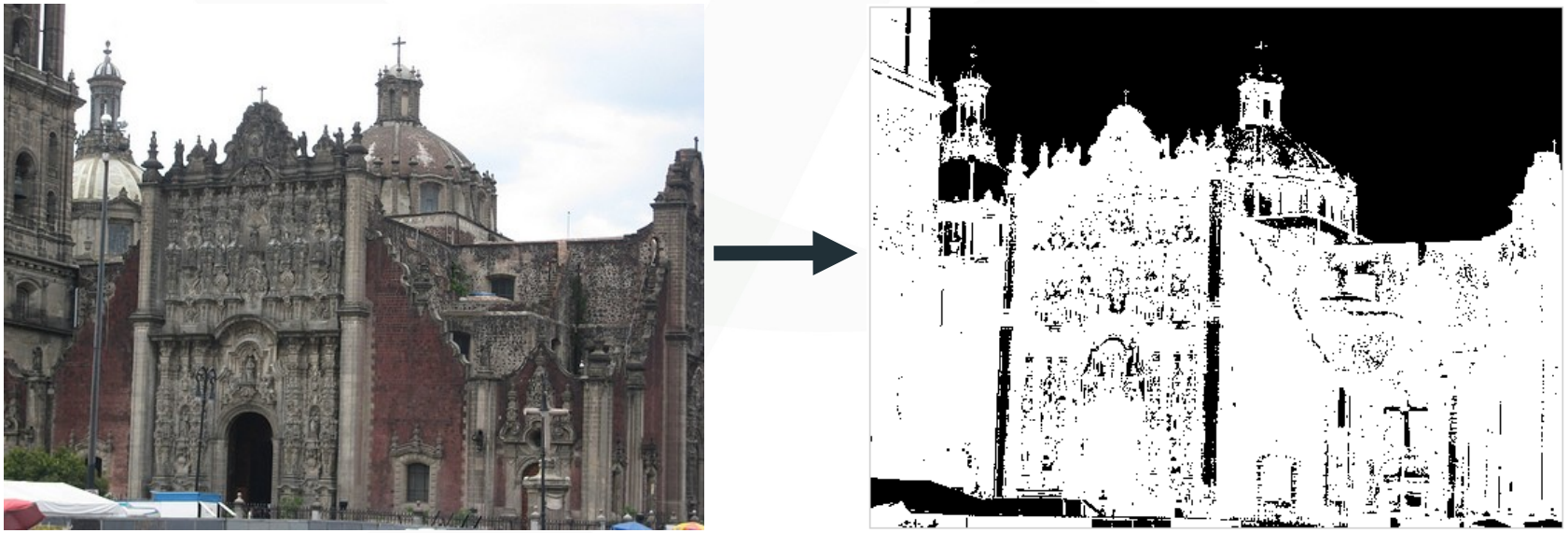
REDRAW IMAGE

```
//clear the stage
clearRect(0,0,canvas.width,canvas.height);
speed = 10;
if (previous_x < mouse_x){
direction = 1; //moving left to right
} else {
direction = -1; //moving right to left
}
var new_x = previous_x + (speed * direction);
context.drawImage(img,new_x,y);
```

sKillfer

IMAGES AT THE PIXEL LEVEL

Get image data and loop through pixel by pixel



sKillfer

LOOPING THROUGH PIXELS

```
c.drawImage(img,0,0);var imgData =
c.getImageData(0,0,canvas.width,canvas.height);
for(n=0; n<data.width*data.height; n++) {var index =
n*4; //each pixel has 4 data points
//rgb values from 0-255
red = imgData.data[index];green = imgData.data[index+1];blue
= imgData.data[index+2];
alpha = imgData.data[index+3];
//to reset pixels, change the values in the data array
imgData.data[index+2] = 255; //change the blue value
}
}
```

sKillfer

ANIMATION

sKillfer

ANIMATION

Canvas doesn't track objects on the screen –

You must redraw the stage for each movement

Use event listeners or timers

sKillfer

FOLLOW THE MOUSE

- Draw the image on the canvas
- Track the position of the mouse (event listener for every mouse move)
- Redraw the image every n seconds based on the x position of the mouse

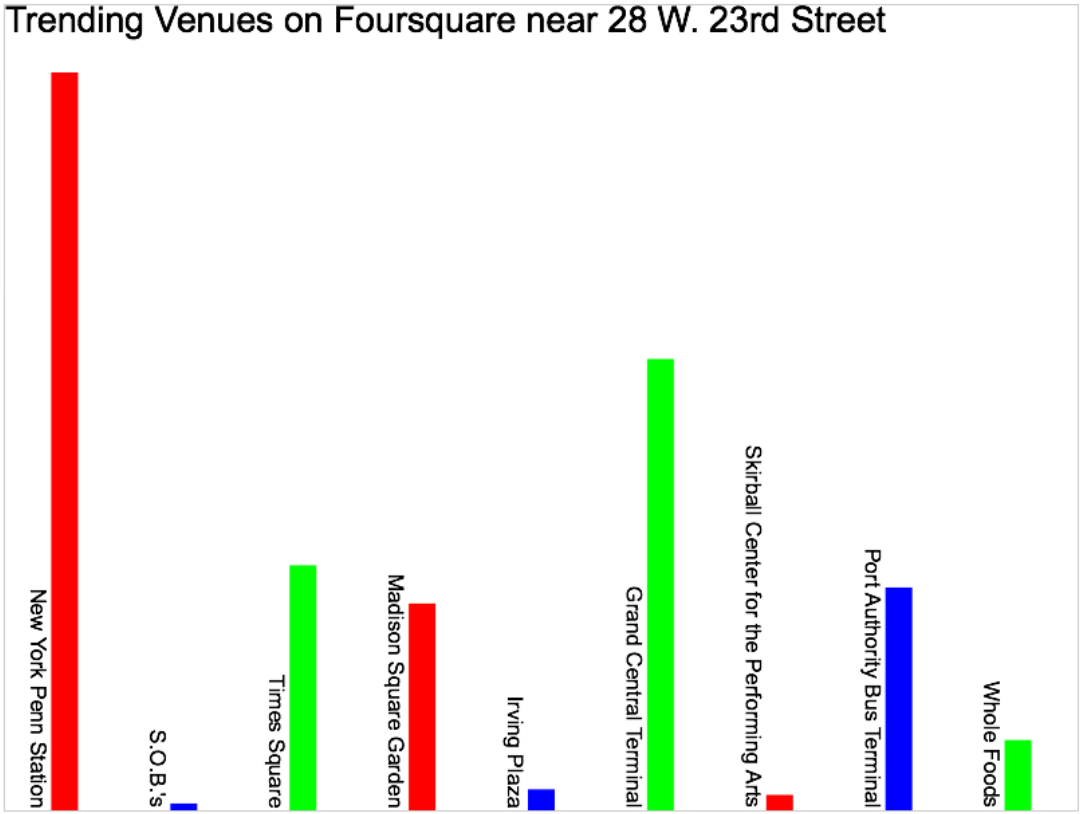
skillfer

VISUALIZE DATA

USE JSON DATA TO CREATE DYNAMIC DRAWINGS

CONNECT TO A WEB SERVICE (TWITTER, FOURSQUARE, ETC)

BAR CHART



sKillfer

MAP DATA TO CANVAS

```
//the data could also come in as a JSON object
var data = {jane: 231,julie: 325,ben: 276}; //js object
var highest_value = 325; //find programmatically
var pixel_units = canvas.height/highest_value;
for (user in data){ //loop through data points
var bar_height = data[user] * pixel_units;
var bar_title = user;
}
//now it's time to draw!
```

sKillfer

COOL EXAMPLE

Each bubble represents a tweet containing the word "water". Popping the bubble displays the tweet.

sKillfer

GRAPHING LIBRAIRIES

- jQuery VisualizeAwesomeChartJS ... many more
- ... many more

sKillfer

MOBILE TIPS

- Javascript touch events instead of mouse events
- PhoneGap Framework to compile to native iOS and Android apps

sKillfer

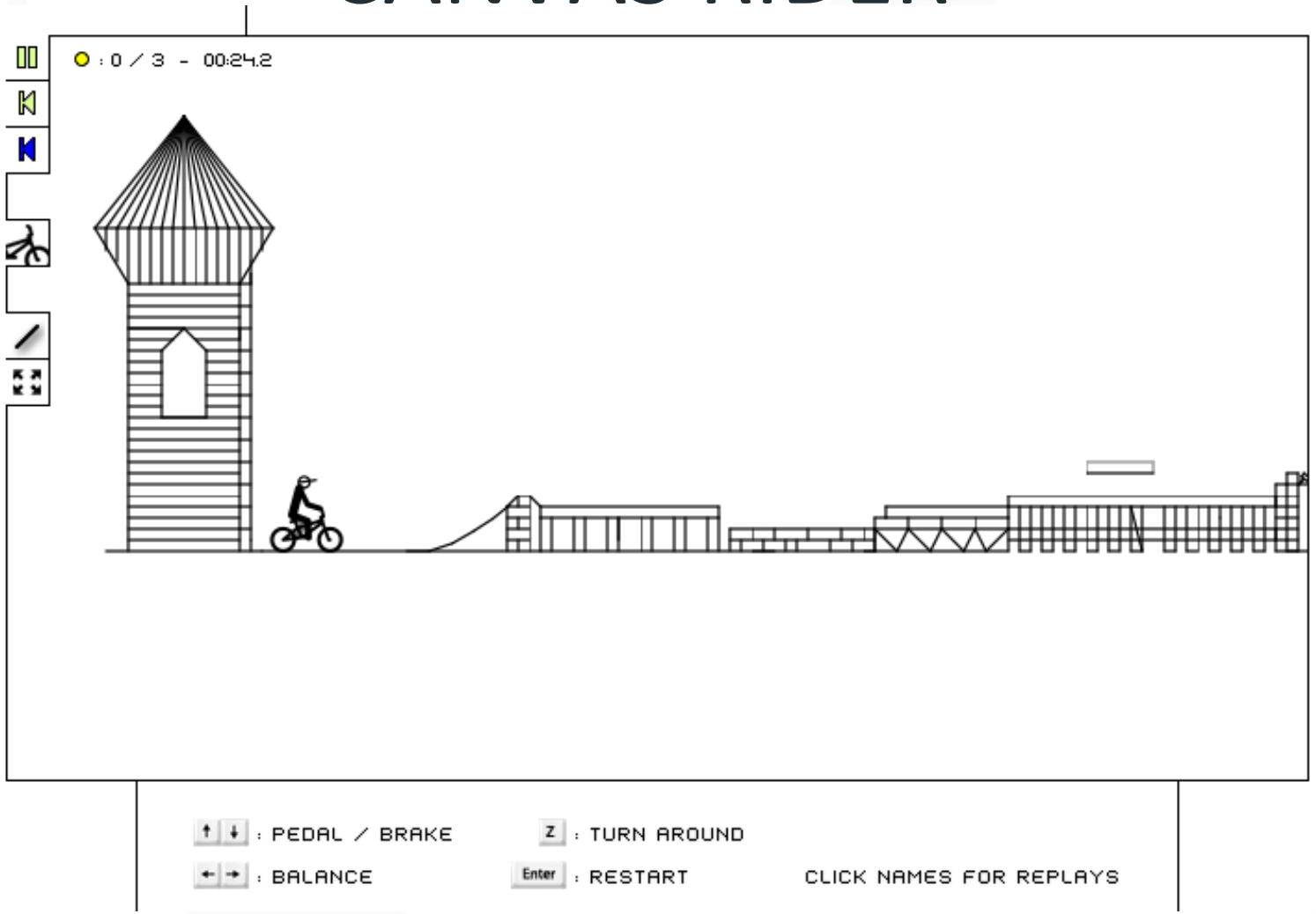
CANVAS GAMES

GAMES EXAMPLES

- Alex the Alligator http://alex4.tapjs.com
- CATcher http://wpsystem.com.br/catcher/
- Catch the goblin http://www.lostdecadegames.com/how-to-make-a-simple-html5-canvas-game/
- Magician Fairy Rescue http://www.relfind.com/game/magician.html
- Canvas Rider, bike game http://canvasrider.com/

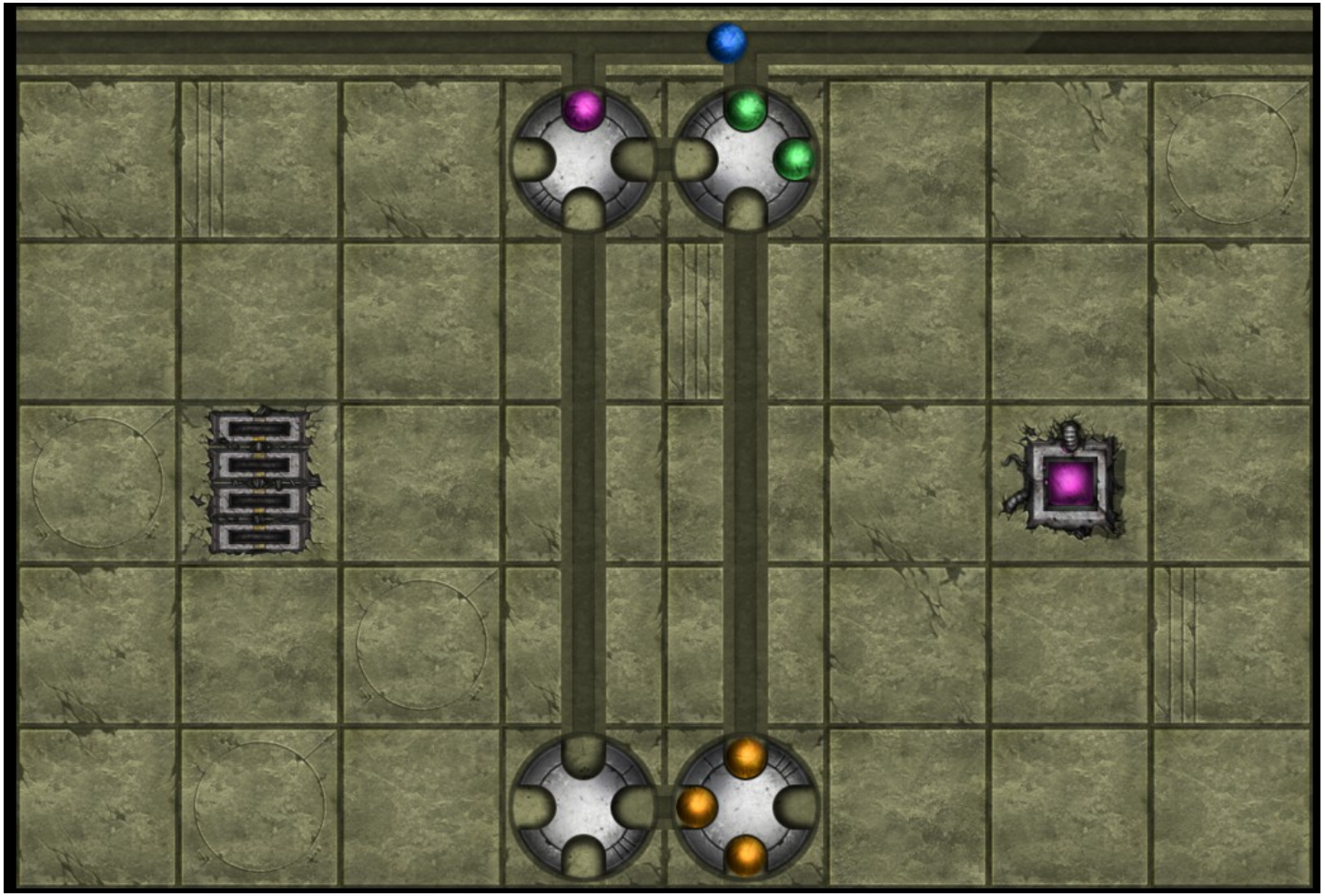
sKillfer

CANVAS RIDER



sKillfer

ORBIUM



sKillfer

SIMPLE GAME EXAMPLE

sKillfer

CODING GOBLINS

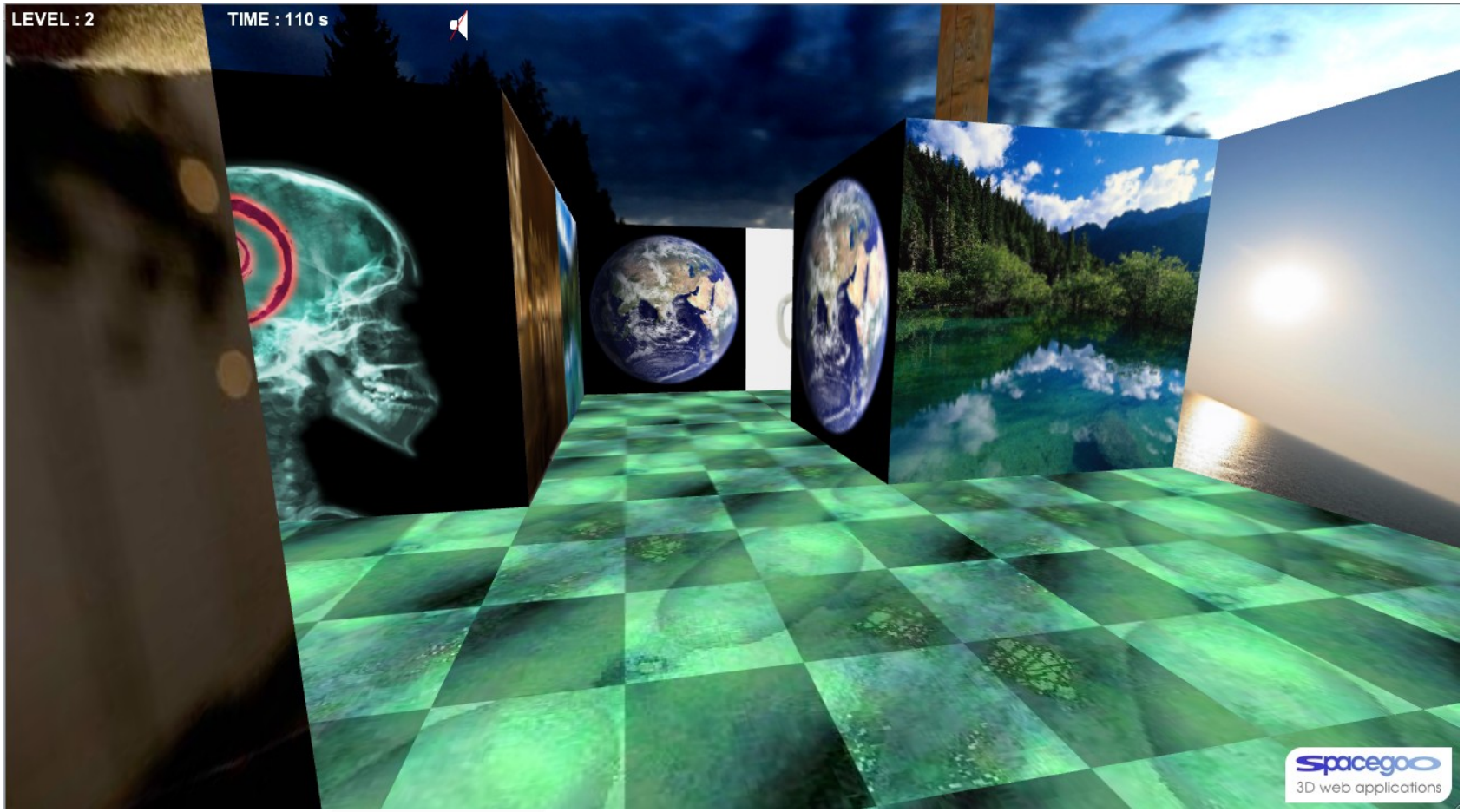
1. Draw images for background, player, and goblin
2. Listen for arrow keys and change player coordinates accordingly
3. Detect hit by comparing monster's coordinates to player's coordinates
4. Redraw as frequently as possible

sKillfer

3D DRAWING

sKillfer

3D IN CANVAS



sKillfer

3D

- A few options:
- Canvas 3D context
- Canvas WebGL context
- Simulated 3D in 2D context
- Libraries ... three.js, c3dl, k3d, Sylvester

sKillfer

WebGL CONTEXT

```
var canvas = document.getElementById("glcanvas");
// Use standard WebGL context or experimental if it's not
available
gl = canvas.getContext("webgl") ||
     canvas.getContext("experimental-webgl");

// Define initial WebGL settings
if (gl) {
    gl.clearColor(0.0, 0.0, 0.0, 1.0);
    gl.enable(gl.DEPTH_TEST);
    gl.depthFunc(gl.LEQUAL);
    gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
}
// It gets more complex from here!
```

sKillfer

LIBRARIES

- KineticJS Canvas drawing & animation library
- Three.JS 3D library
- Processing.js Port of Processing environment
- GLGE WebGL library
- melonJS game engine
- ImpactJS game engine

sKillfer

PROCESSING.js

- Reads files from Processing programming environment
- Uses processing-like code in Javascript
- More flexibility with objects -- collisions, mouseover, etc

sKillfer

KINETIC JS

- Dynamically creates a series of canvas elements as "layers"
- Provides a Stage class and code similar to Actionscript
- Allow easy object moving, dragging, and tracking

sKillfer

LEARNING RESOURCES

BEST LIBRARIES
TUTORIALS
& TOP EXAMPLES

Q&A

THANK YOU!

Robyn Overstreet & Jovena Whatmoor