Canvas

by Kevin Hoyt

Agenda

- Getting started
- Drawing
- Charting
- Images
- Interaction
- Animation

Getting Started

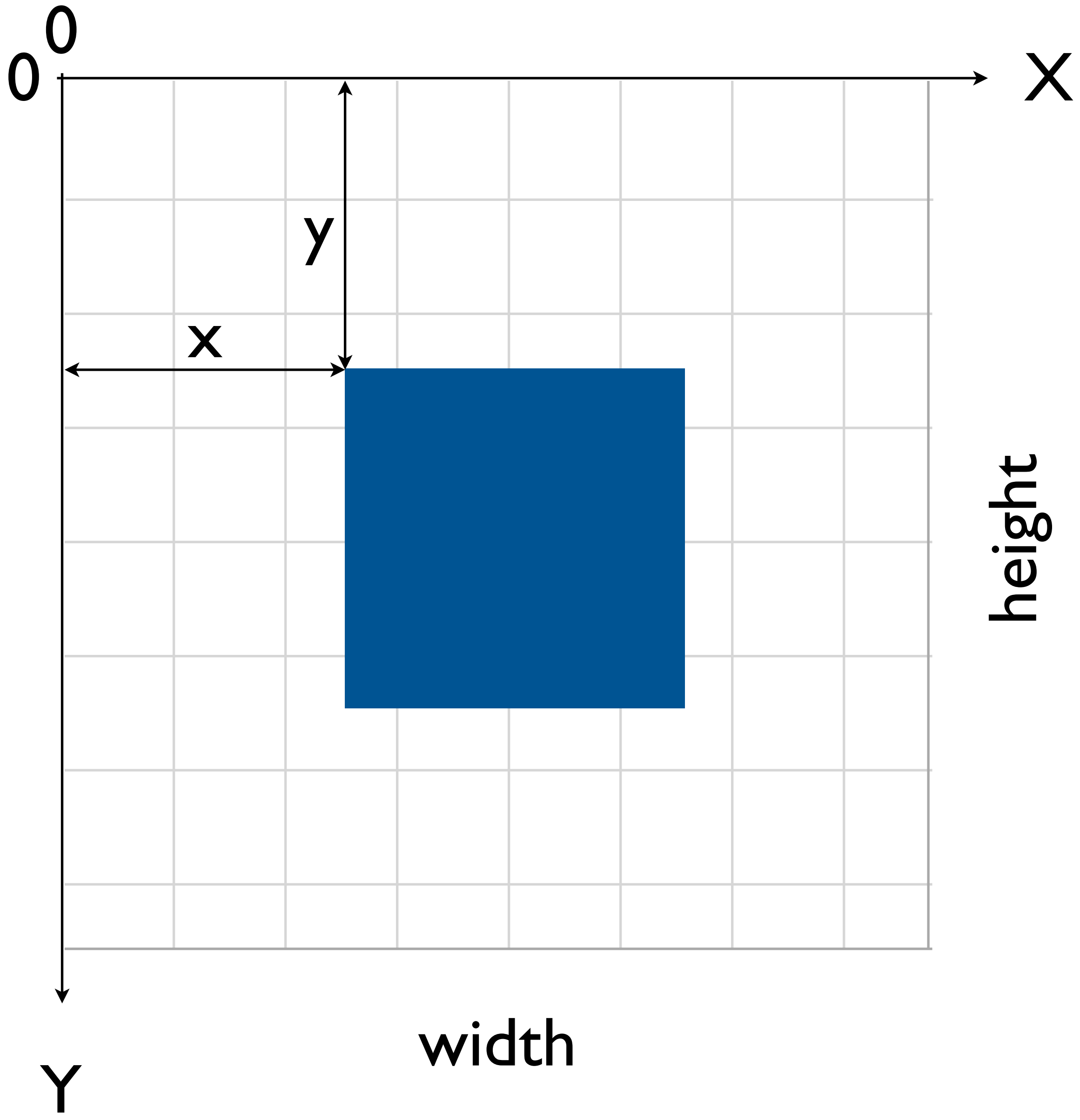
Using the Canvas Tag

```
<canvas id="chart" width="150" height="150">
    Current stock price: $3.15 +0.15
</canvas>

<canvas id="clock" width="150" height="150">
    <img
        src="imagesclock.png"
        width="150"
        height="150"/>
</canvas>
```

Rendering Context

```
var canvas = document.getElementById( 'canvas' );
var context = null;

if( canvas.getContext )
{
  context = canvas.getContext( '2d' );
  // Drawing code here
} else {
  // Unsupported code here
}
```

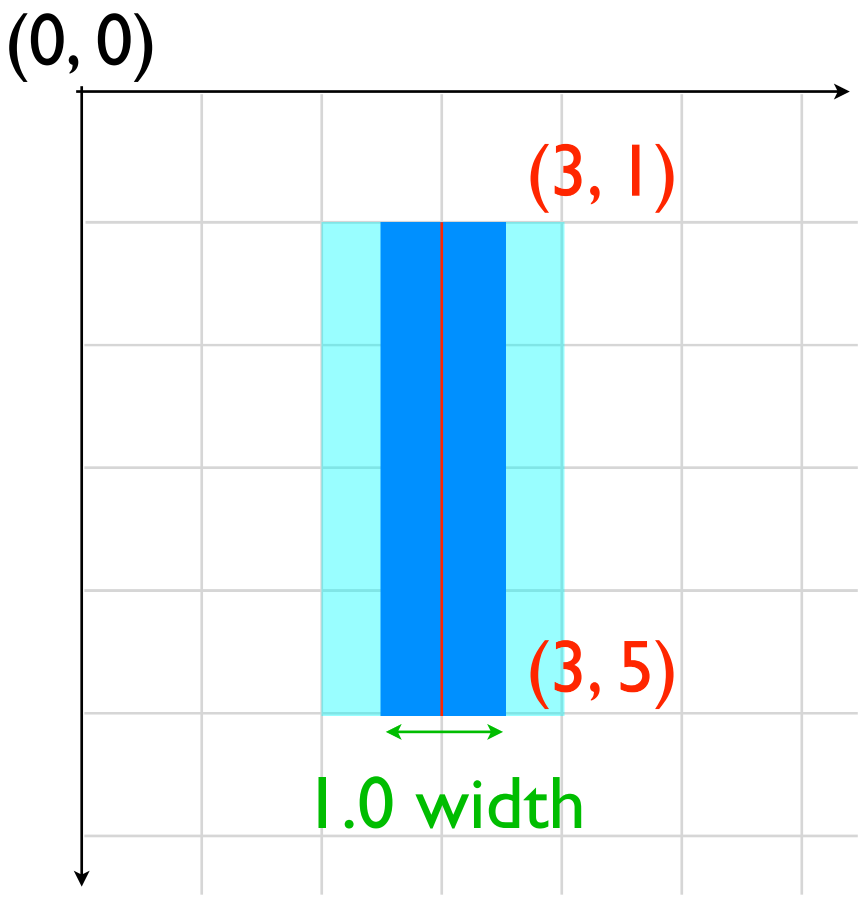
Coordinate Space

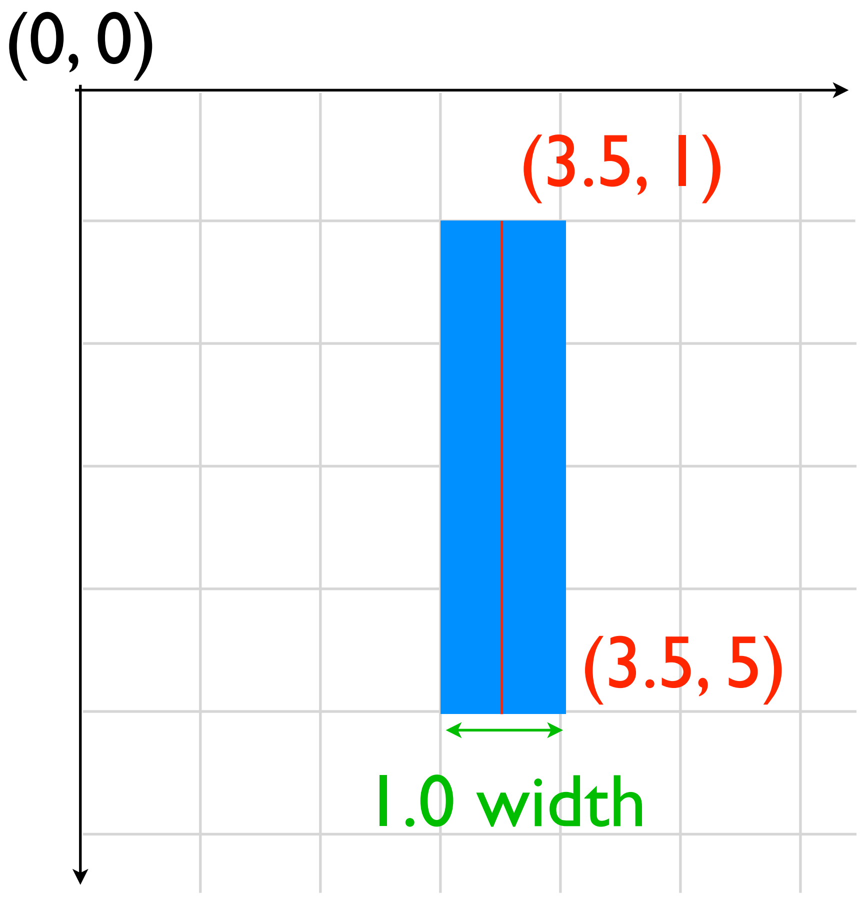


Drawing Paths

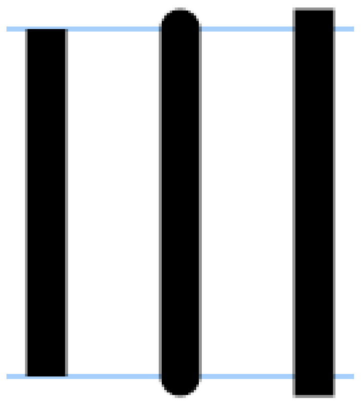
```
context.beginPath();

context.arc( 75, 75, 50, 0, Math.PI * 2, true );
context.moveTo( 110, 75 );
context.arc( 75, 75, 35, 0, Math.PI, false );
context.moveTo( 65, 65 );
context.arc( 60, 65, 5, 0, Math.PI * 2, true );
context.moveTo( 95, 65 );
context.arc( 90, 65, 5, 0, Math.PI * 2, true );

// context.closePath();
context.stroke();
// context.fill();
```

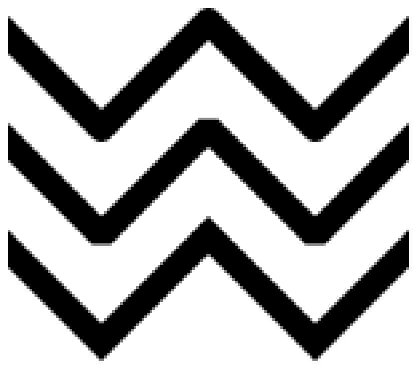
Understanding Lines

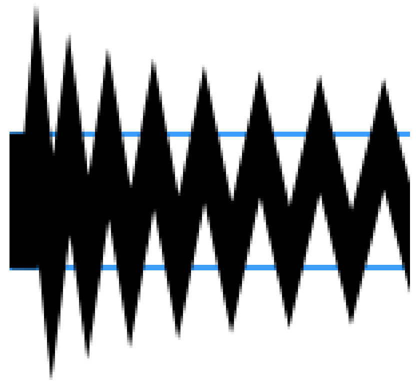




Caps, Joints and Miters





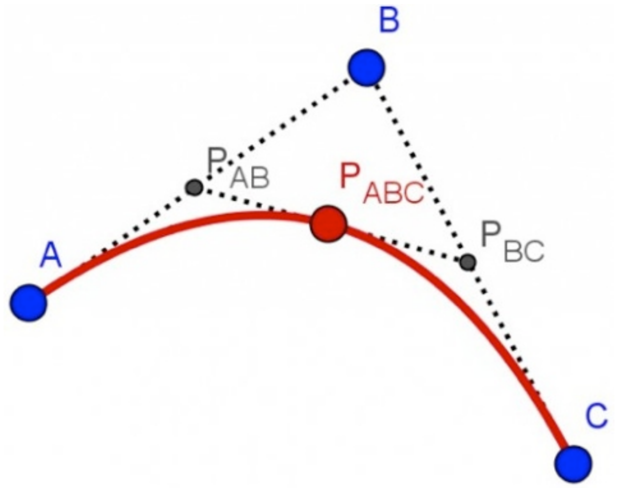


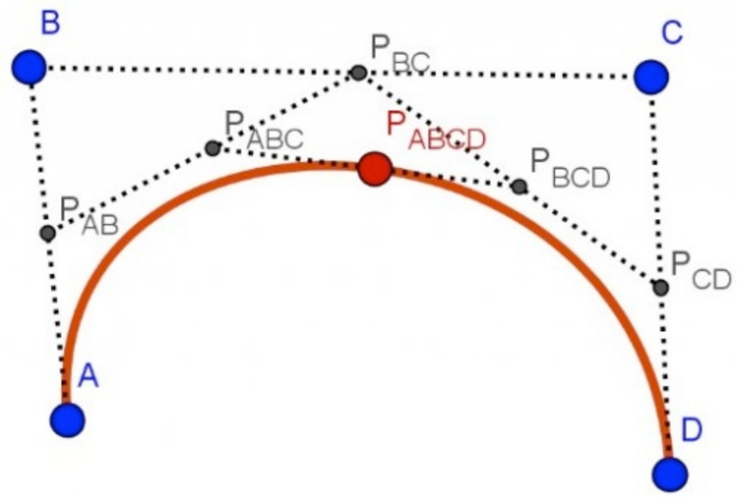
butt
round
square

round
bevel
miter

width / 2
x limit
from joint

Curves





Quadratic Curve

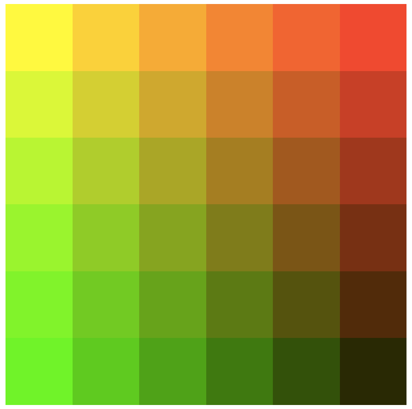
$$\mathbf{B}(t) = (1-t)^2\mathbf{P}_0 + 2(1-t)t\mathbf{P}_1 + t^2\mathbf{P}_2 \ , \ t \in [0,1].$$

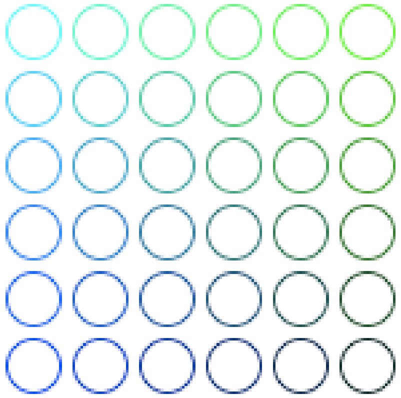
Cubic Curve

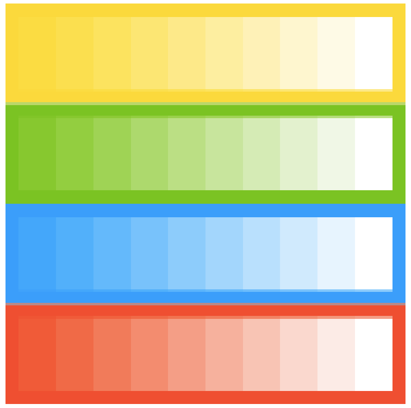
$$\mathbf{B}(t) = (1-t)^3\mathbf{P}_0 + 3(1-t)^2t\mathbf{P}_1 + 3(1-t)t^2\mathbf{P}_2 + t^3\mathbf{P}_3 \ , \ t \in [0,1].$$

Styles and Colors

```
fillStyle
strokeStyle
```







```
orange
#FFA500
rgb( 255, 165, 0 )
rgba( 255, 165, 0, 1 )
```

Gradients

```
// Start point and stop point
// Controls angle/direction of gradient
var fill = context.createLinearGradient( 0, 0, 0, 50 );

// 0 - 1 relative to gradient range
fill.addColorStop( 0, ‘#FFFFFF’ );
fill.addColorStop( 0.5, ‘#FFA500’ );
fill.addColorStop( 1, ‘#FFFFFF’ );

// Apply the fill style
context.fillStyle = fill;
context.fillRect( 0, 0, 50, 50 );
```

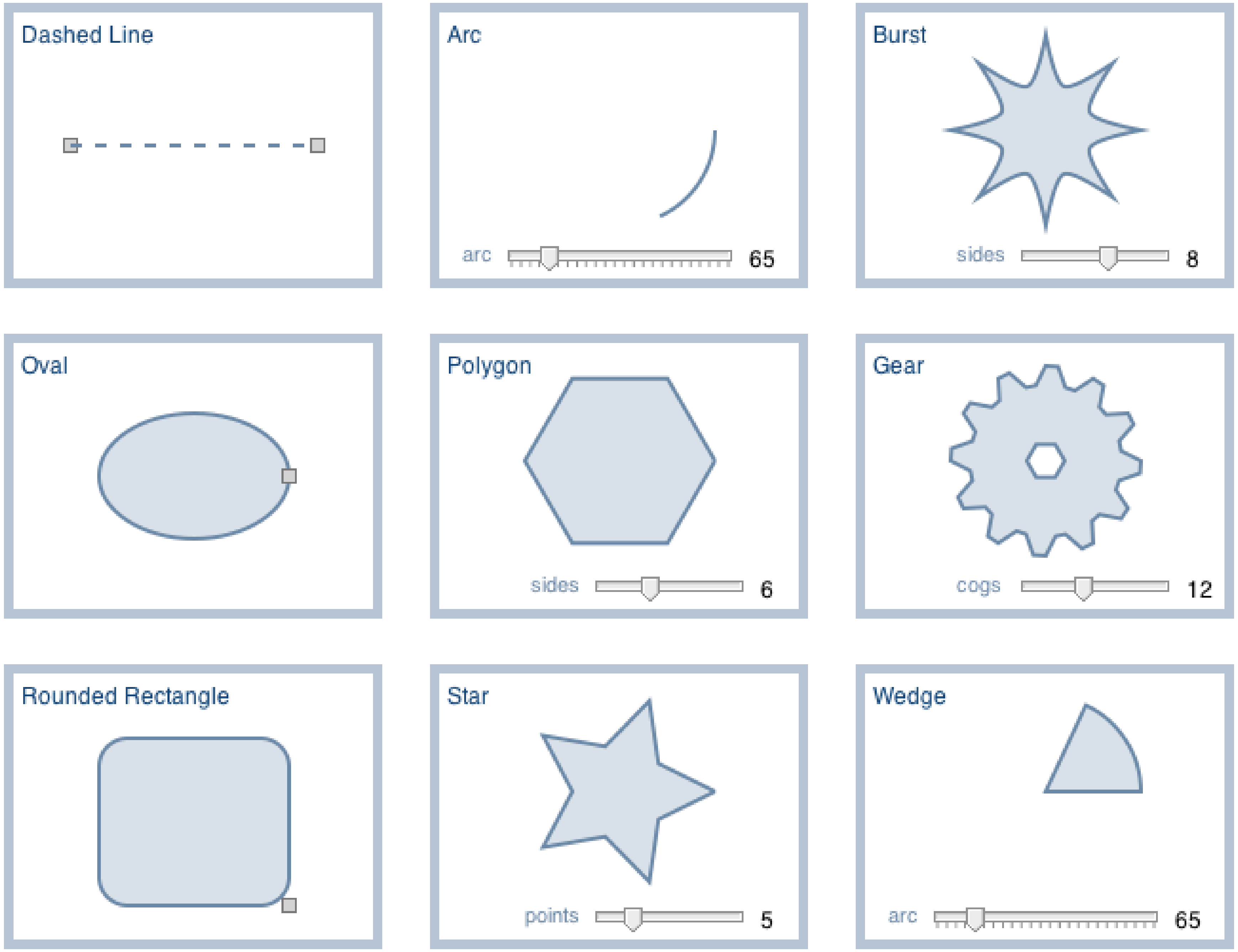
Gradients

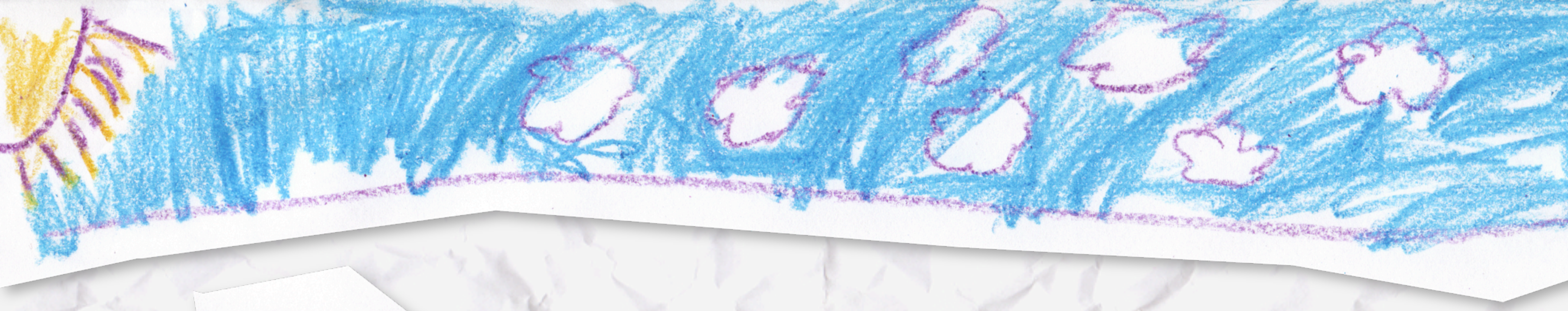
```
// Inner circle point and radius
// Outer circle point and radius
// Not necessarily a single point
var fill = context.createRadialGradient(
    95, 15, 15,
    102, 20, 40
);


// 0 - 1 relative to gradient range
fill.addColorStop( 0, ‘#FFFFFF’ );
fill.addColorStop( 0.5, ‘#FFA500’ );
fill.addColorStop( 1, ‘#FFFFFF’ );

// Apply the fill style
context.fillStyle = fill;
context.fillRect( 0, 0, 50, 50 );
```

Text and Shadows

```
var canvas = document.getElementById( 'canvas' );
var ctx = null;

if( canvas.getContext )
{
    ctx = canvas.getContext( '2d' );

    ctx.shadowOffsetX = 2;
    ctx.shadowOffsetY = 2;
    ctx.shadowBlur = 2;
    ctx.shadowColor = 'rgba( 0, 0, 0, 0.5 )';

    ctx.font = '20px Times New Roman';
    ctx.fillStyle = 'black';
    ctx.fillText( 'Sample String', 5, 30 );
}
```

Shapes


Dashed Line
Arc
arc
65
Burst
sides
8
Oval
Polygon
sides
6
Gear
cogs
12
Rounded Rectangle
Star
points
5
Wedge
arc
65

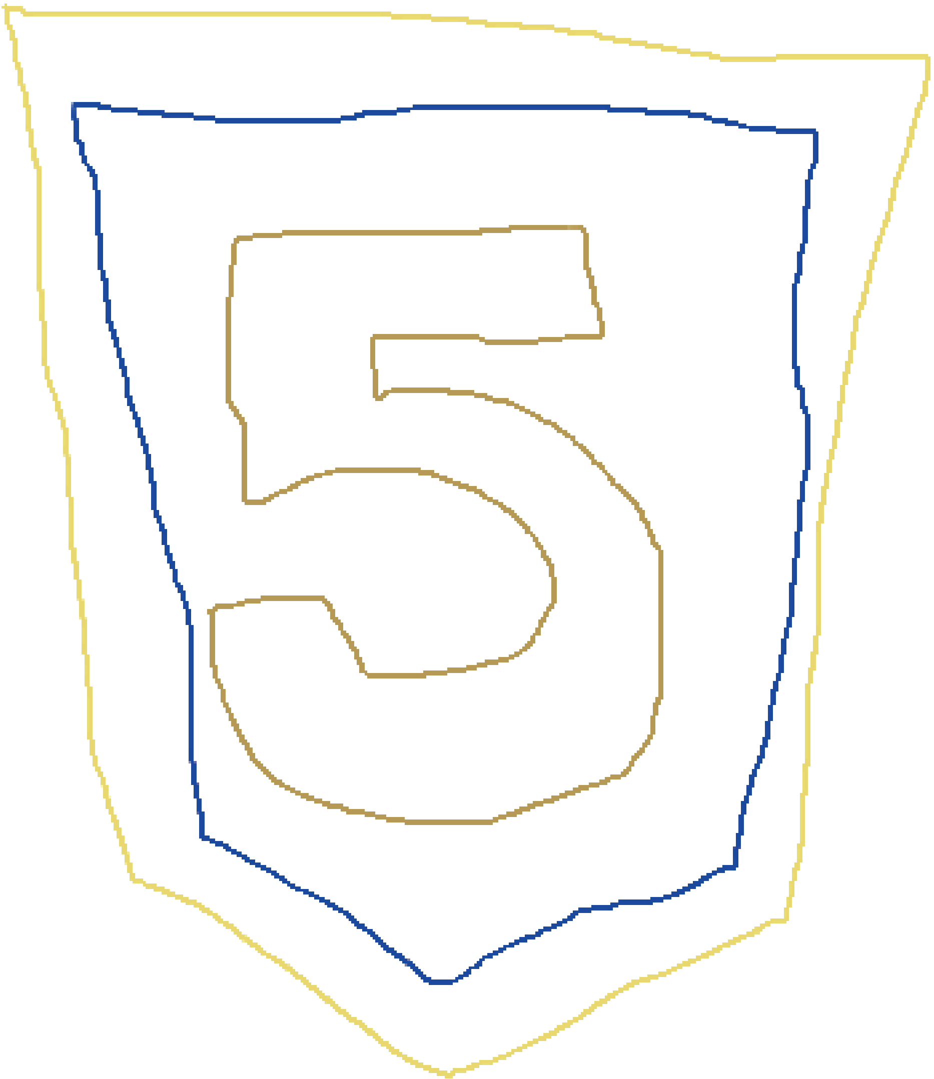


Drawing

Follow the Mouse

```
$( '#canvas' ).mousedown( function( evt ) {
  var coord = coordinate( evt, this );

  context.strokeStyle = color();
  context.moveTo( coord.x, coord.y );
  context.beginPath();

  $( '#canvas' ).mousemove( function( evt ) {
    var coord = coordinate( evt, this );

    context.lineTo( coord.x, coord.y );
    context.stroke();
  } ).mouseup( function( evt ) {
    $( '#canvas' ).unbind( 'mousemove' );
    $( '#canvas' ).unbind( 'mouseup' );
  } );
} );
```

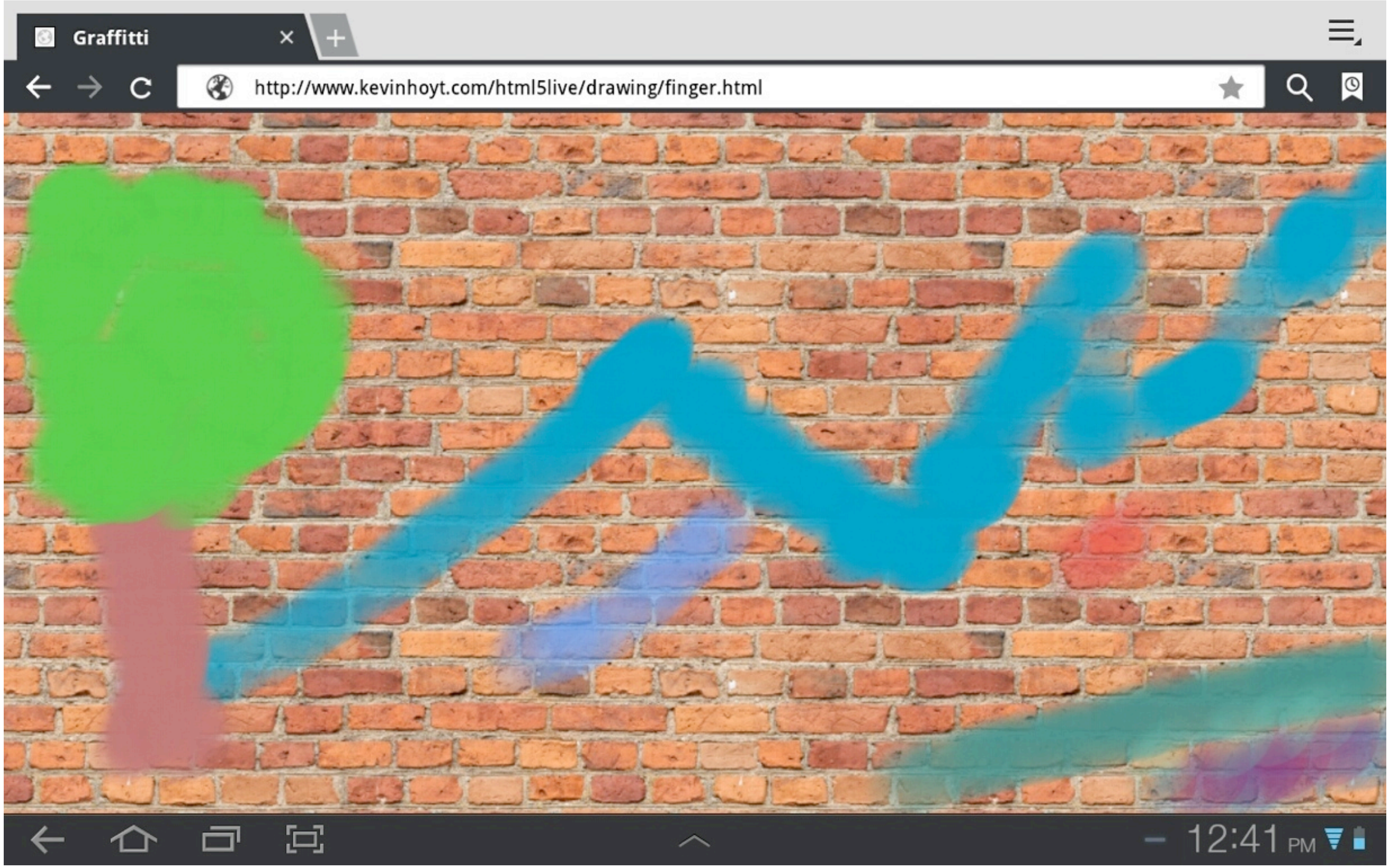
Follow the Mouse



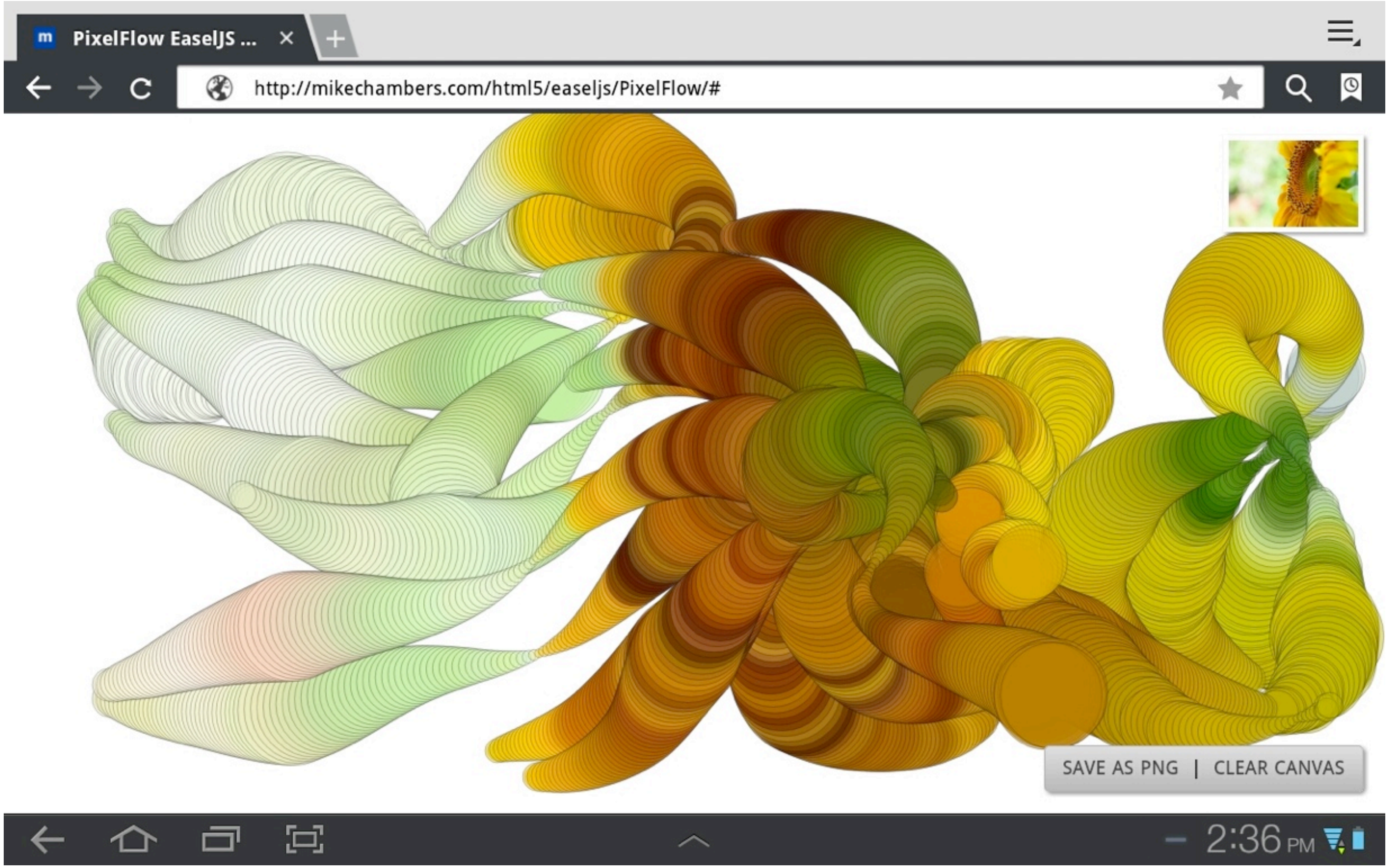
Follow the Finger

```
$( '#canvas' ).bind( ‘touchstart’, function( evt ) {
  evt.preventDefault();
  fill = color();

  $( '#canvas' ).bind( ‘touchmove’, function( evt ) {
    var touch = evt.originalEvent.touches[0];

    evt.preventDefault();
    context.fillStyle = radial(
      context, touch.clientX, touch.clientY );
    context.strokeStyle = 'rgba( 0, 255, 0, 0 )';
    context.beginPath();
    context.arc(
      touch.clientX, touch.clientY, 40, 0, Math.PI * 2, true );
    context.stroke();
    context.fill();
  } ).bind( ‘touchend’, function( evt ) {
    $( '#canvas' ).unbind( 'touchmove' );
    $( '#canvas' ).unbind( 'touchend' );
  } );
} );
```

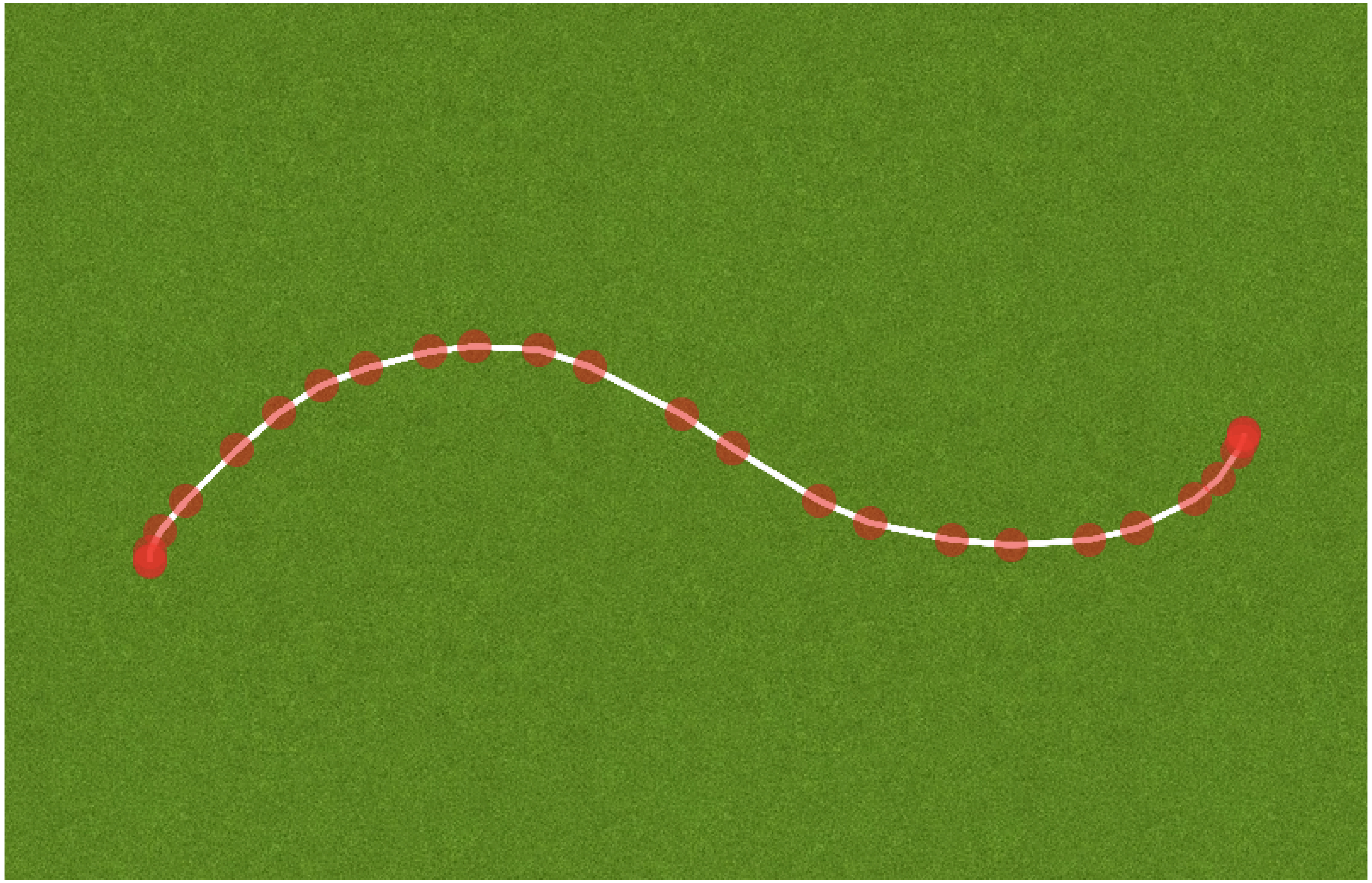
Follow the Finger



Follow the Finger


PixelFlow EaselJS ...
http://mikechambers.com/html5/easeljs/PixelFlow/#
SAVE AS PNG | CLEAR CANVAS
2:36 PM

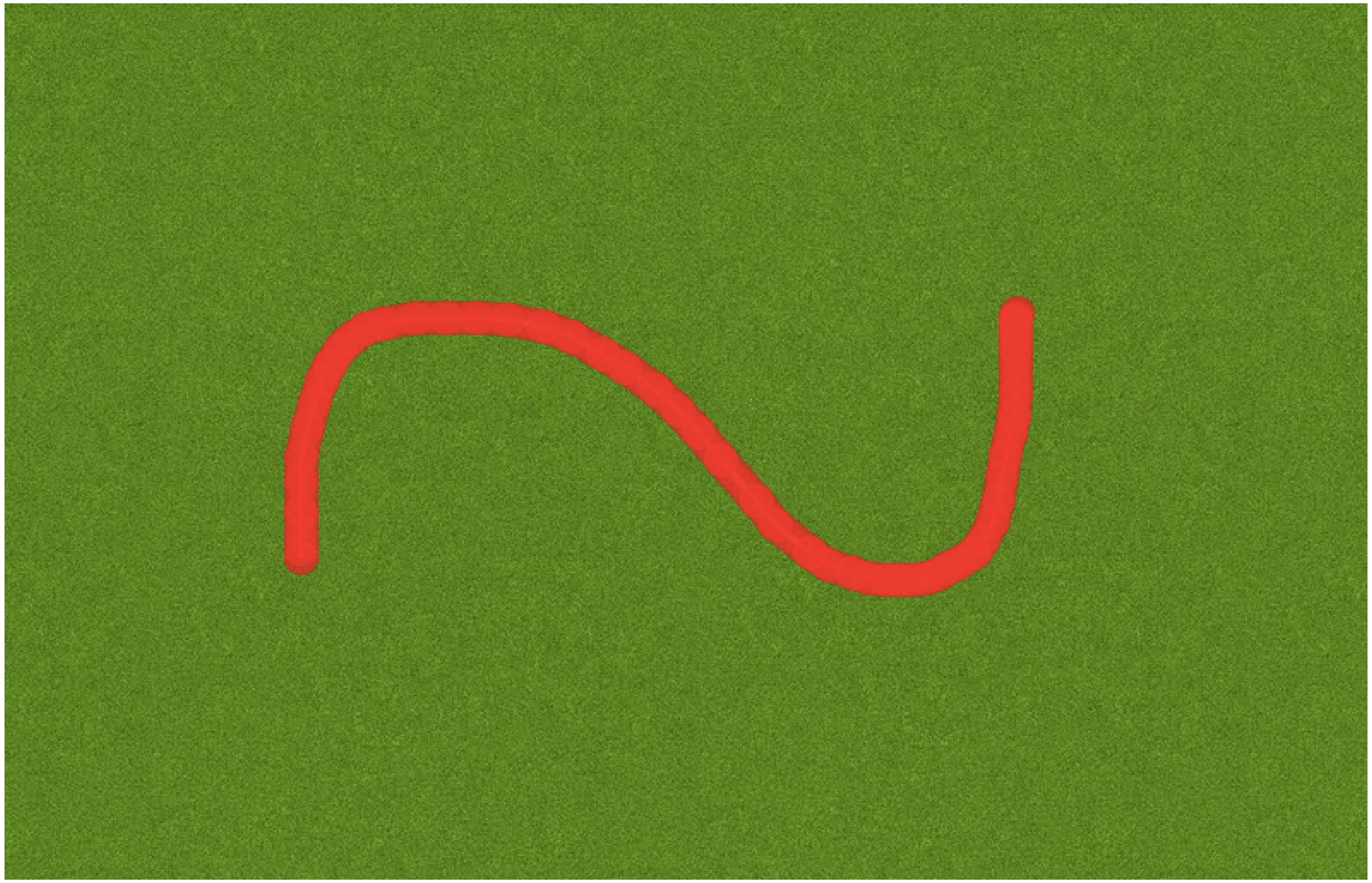
Line Interpolation



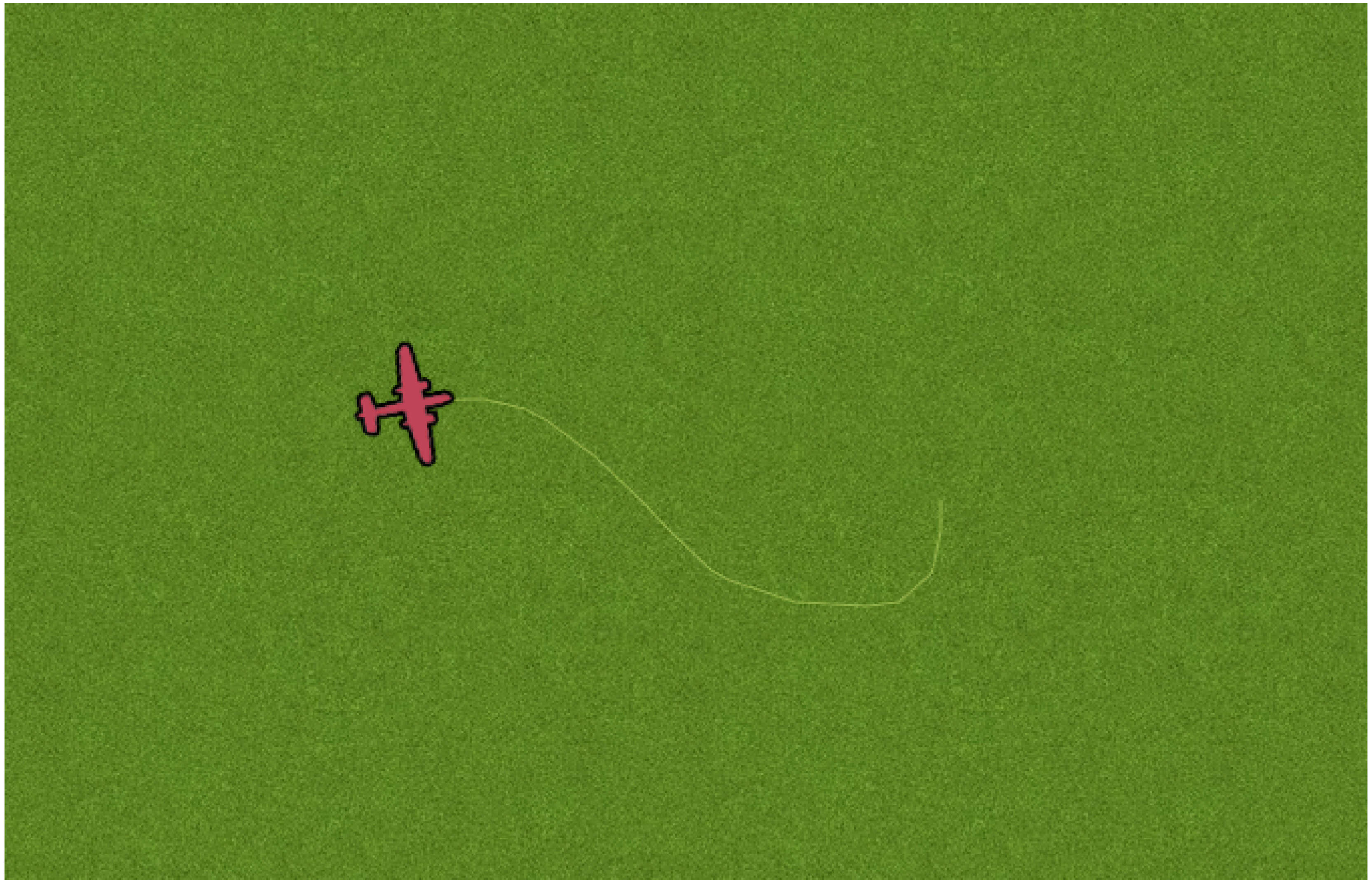
Line Interpolation

```
var xabs = Math.abs( point1.x - point2.x );
var yabs = Math.abs( point1.y - point2.y );
var xdiff = point2.x - point1.x;
var ydiff = point2.y - point1.y;

var length = Math.sqrt( (
  Math.pow( xabs, 2 ) + Math.pow( yabs, 2 ) ) );
var steps = length / distance;
var xstep = xdiff / steps;
var ystep = ydiff / steps;

var newx, newy = 0;
var result = new Array();

for( var s = 0; s < steps; s++ ) {
  newx = point1.x + ( xstep * s );
  newy = point1.y + ( ystep * s );

  result.push( {x: newx, y: newy} );
}
```

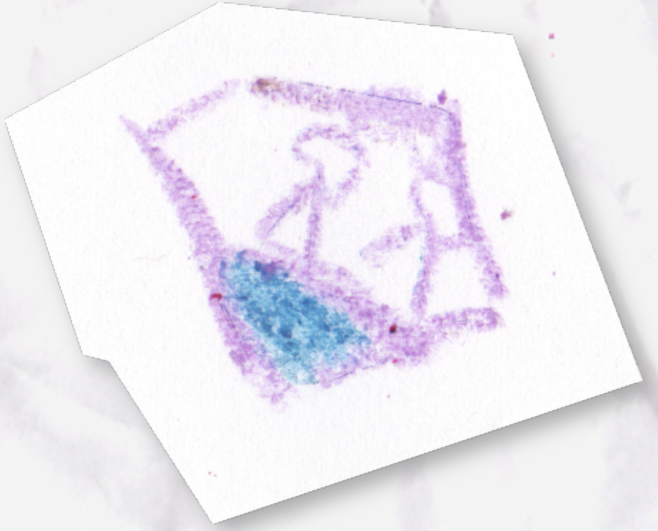
Line Interpolation



Line Interpolation



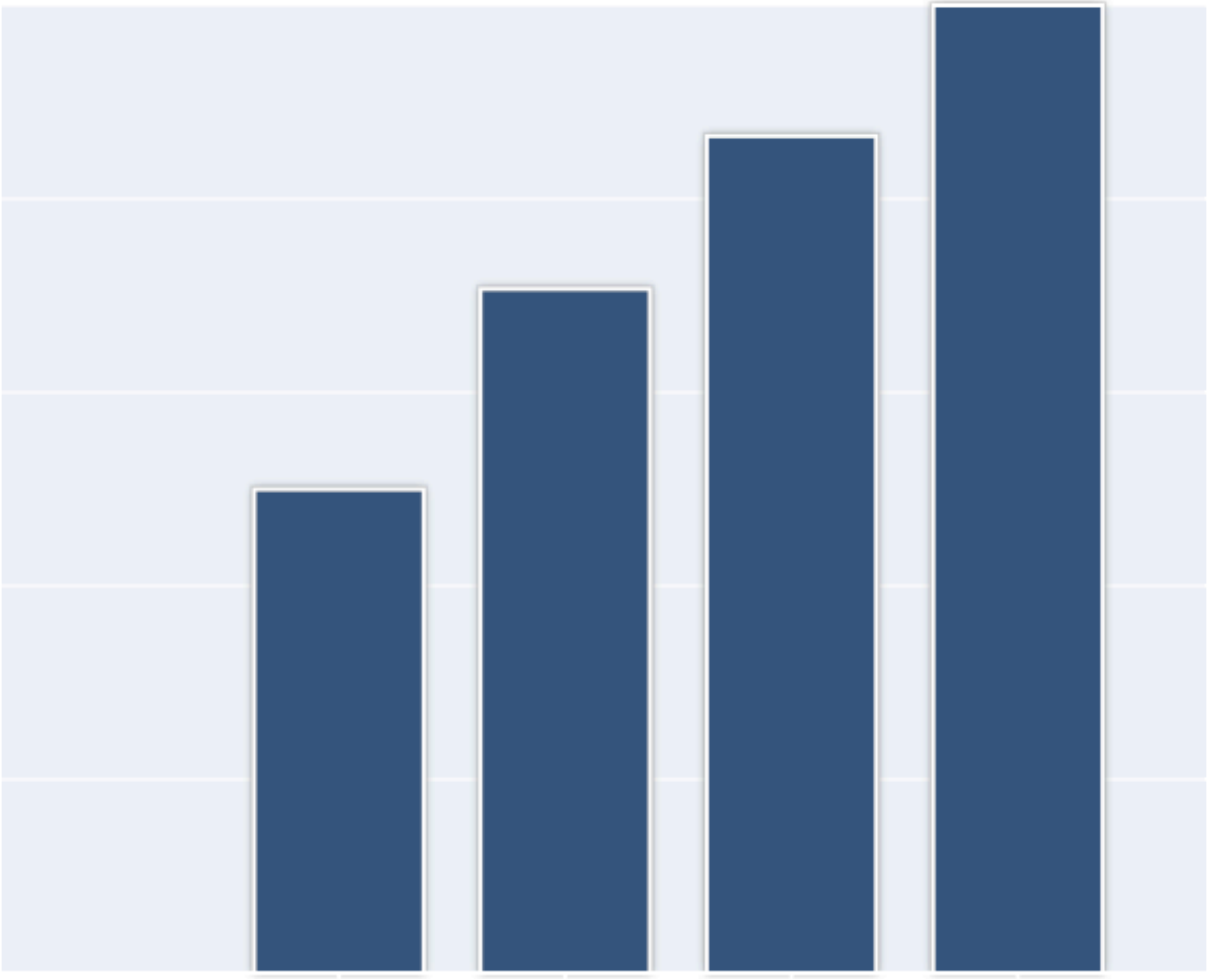
Charting



PlotKit

```
MochiKit.DOM.addLoadEvent( function() {
  var canvas = MochiKit.DOM.getElement( 'chart' );
  var layout = new PlotKit.Layout( 'bar', {} );
  var plotter = new PlotKit.SweetCanvasRenderer(
    canvas,
    layout, {
    padding: {
      left: 40,
      right: 25,
      top: 25,
      bottom: 30
    },
  } );

  layout.addDataset( 'sqrt', [
    [0, 0], [1, 1], [2, 1.414], [3, 1.73], [4, 2]
  ] );
  layout.evaluate();
  plotter.render();
} );
```

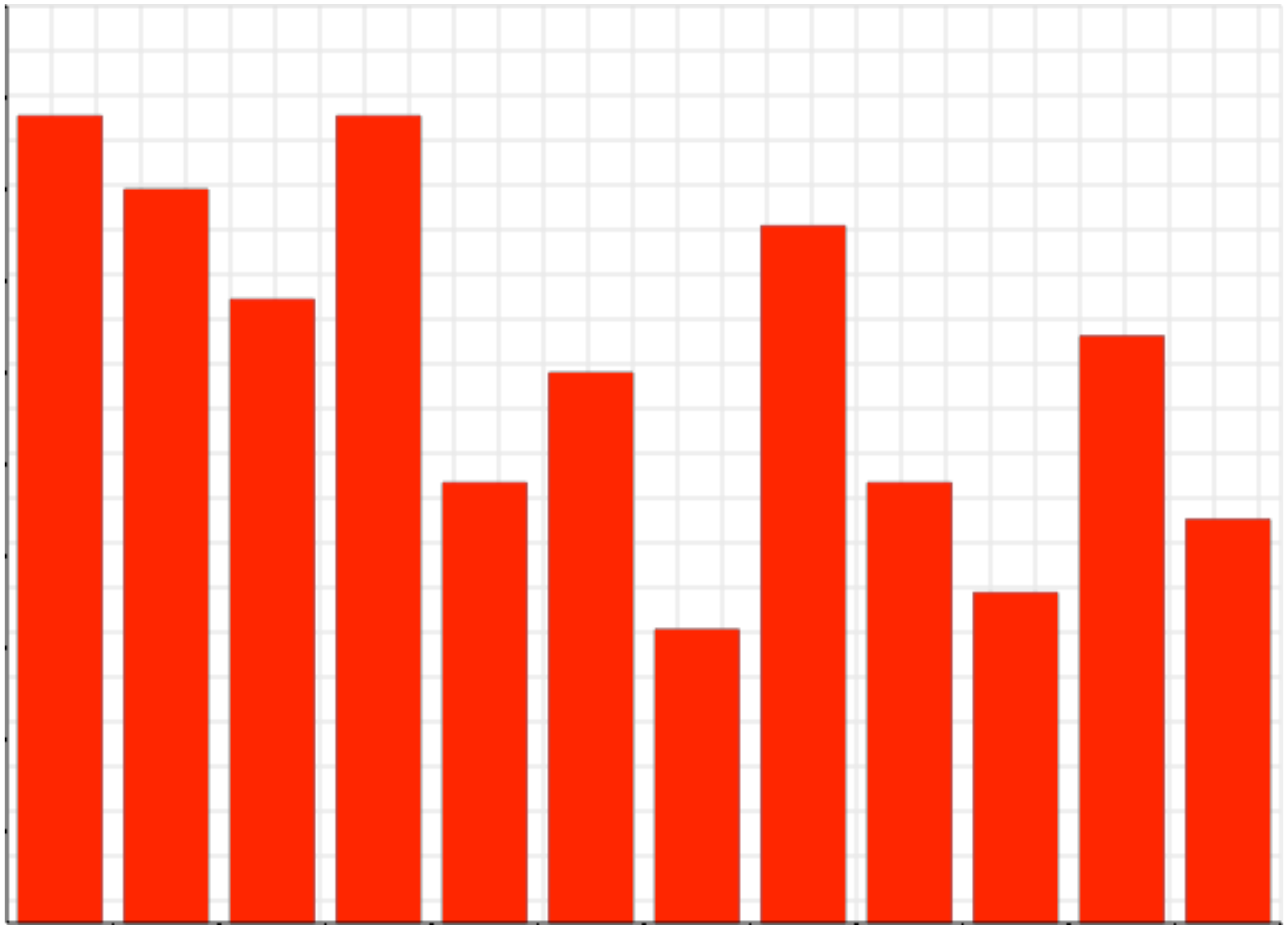
PlotKit



RGraph

```
$( document ).ready( function() {
  var bar = null;
  var data = [
    280, 45, 133, 166, 84,
    259, 266, 960, 219, 311];

  bar = new RGraph.Bar( 'chart', data );
  bar.Set( 'chart.labels', [
    'Richard', 'Alex', 'Nick', 'Scott', 'Kjnell',
    'Doug', 'Charles', 'Michelle', 'Mark', 'Alison'
  ] );
  bar.Set( 'chart.gutter.left', 45 );
  bar.Set( 'chart.background.barcolor1',
    'rgba( 255, 255, 255, 1 )' );
  bar.Set( 'chart.background.barcolor2',
    'rgba( 255, 255, 255, 1 )' );
  bar.Set( 'chart.background.grid', true );
  bar.Set( 'chart.colors', ['rgba( 255, 0, 0, 1 )'] );
  bar.Draw();
} );
```

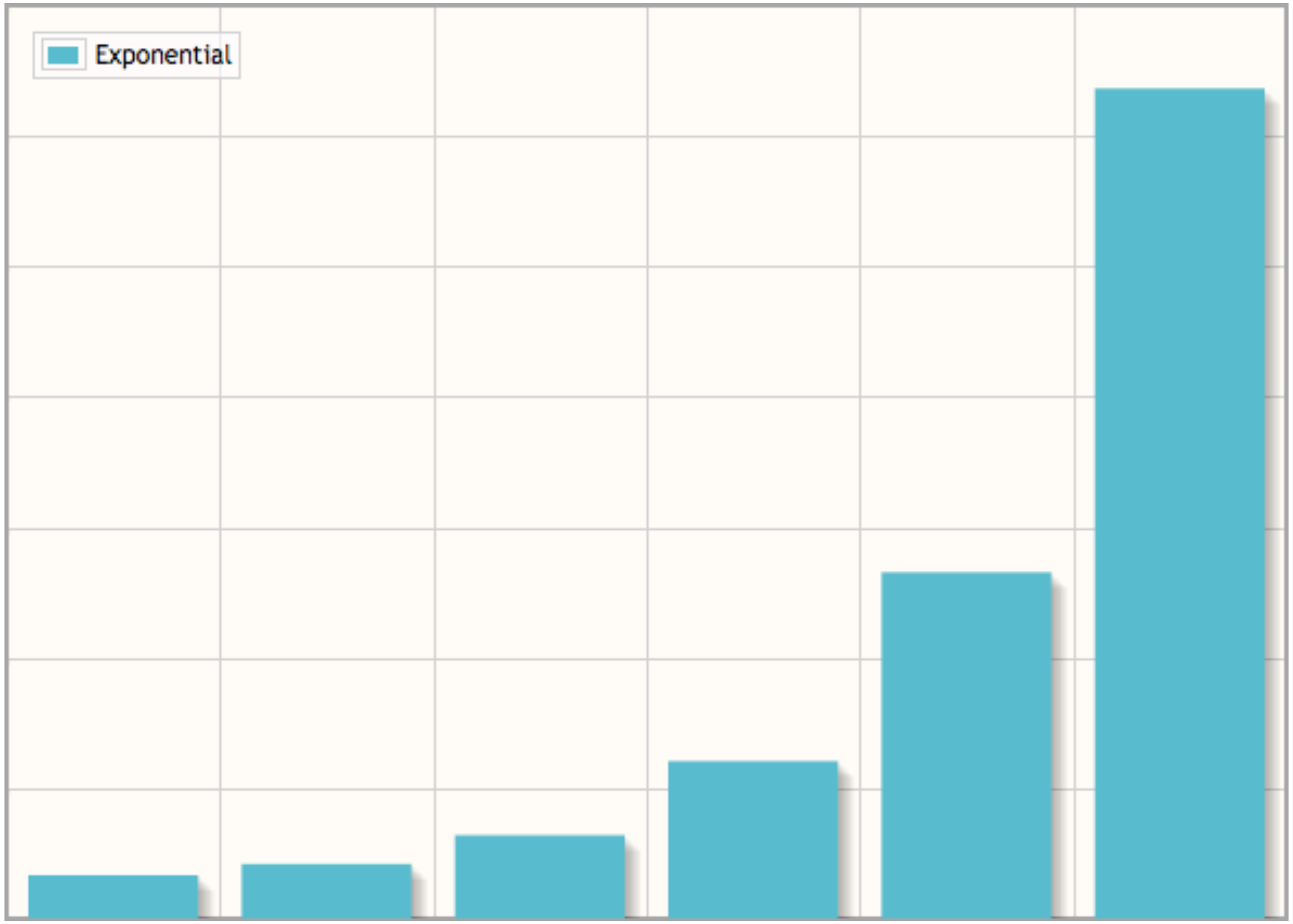
RGraph



jqPlot

```
var bar = null;
var data = new Array();
var value = null;

$.jqplot.config.enablePlugins = true;

$.jqplot( 'chart', [data], {
  legend: {show: true, location: 'nw'},
  title: 'Bar Chart',
  series: [
    {label: 'Random Numbers', renderer: $.jqplot.BarRenderer}
  ],
  axes: {
    xaxis: {renderer: $.jqplot.CategoryAxisRenderer},
    yaxis: {min: 0}
  }
} );
```

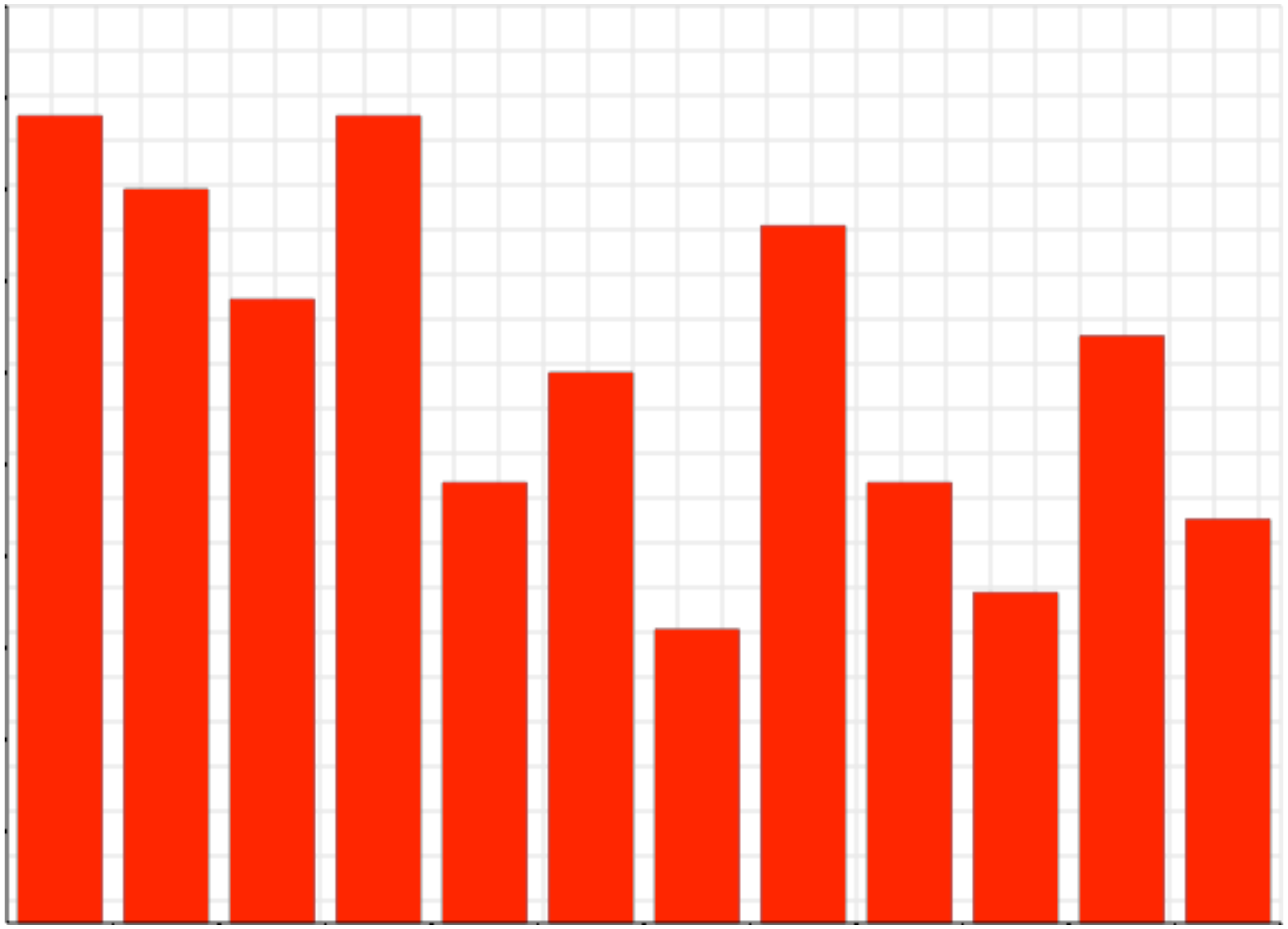
jqPlot



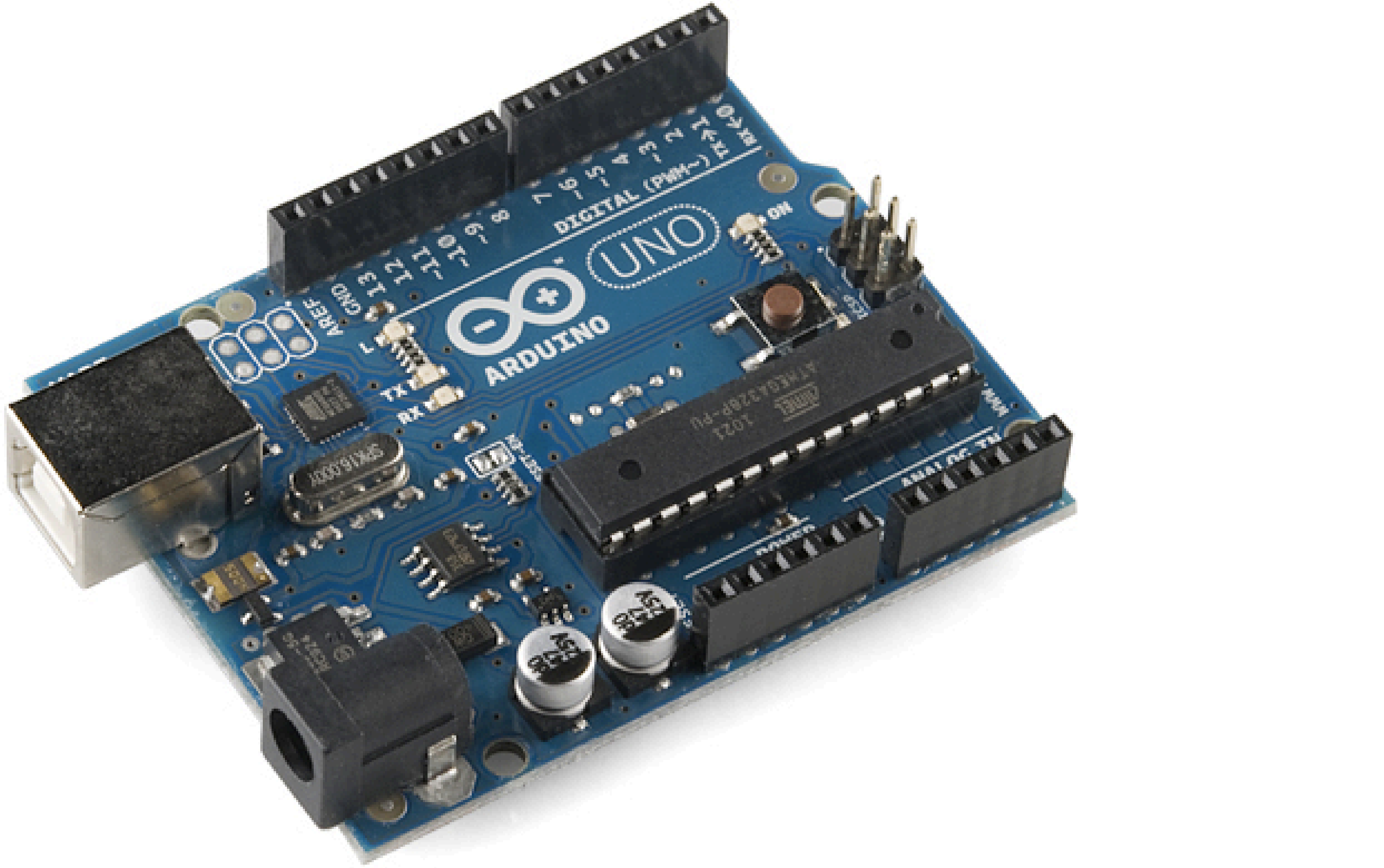
Real-Time

```
setInterval( function() {
  data.splice( 0, 1 );
  data.push( Math.floor( Math.random() * 25 ) );

  RGraph.Clear( canvas );
  bar.data = data;
  bar.Draw();
}, 500 );
```

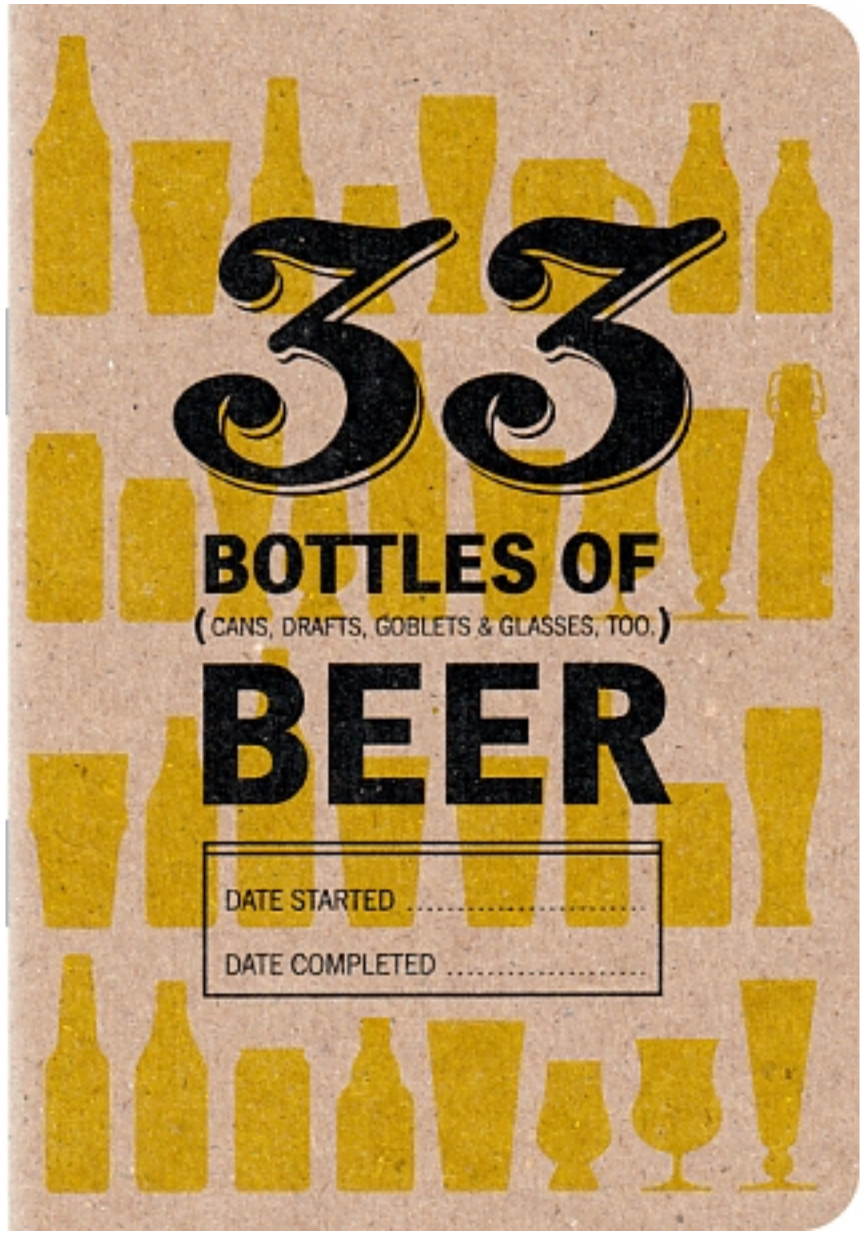
Real-Time

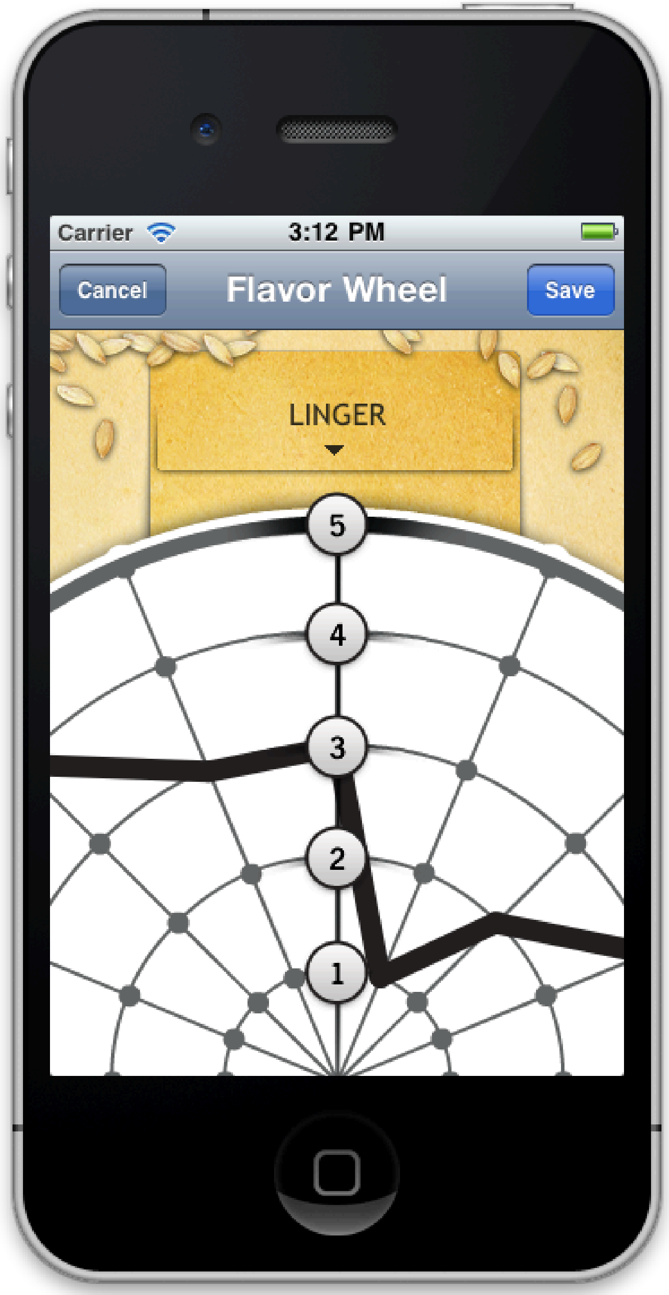


Real-Time


ARDUINO
UNO
DIGITAL (PWM~)
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
ON
L
TX
RX
ANALOG IN
POWER

Roll Your Own





Images

Load From Server

```
var canvas = document.getElementById( 'canvas' );
var ctx = canvas.getContext( '2d' );
var img = null;

$( '#make' ).click( function( evt ) {
    img = new Image();
    img.onload = function() {
        ctx.drawImage( img, 0, 0 );
    };
    img.src = 'images/backdrop.png';
} );
```

Other Canvas

```
var canvas = document.getElementById( 'lines' );
var ctx = canvas.getContext( '2d' );

ctx.beginPath();
ctx.moveTo( 30, 96 );
ctx.lineTo( 70, 66 );
ctx.lineTo( 103, 76 );
ctx.lineTo( 170, 15 );
ctx.stroke();

canvas = document.getElementById( 'canvas' );
ctx = canvas.getContext( '2d' );

$( '#lines' ).click( function( evt ) {
    ctx.drawImage( this, 0, 0 );
} );
```

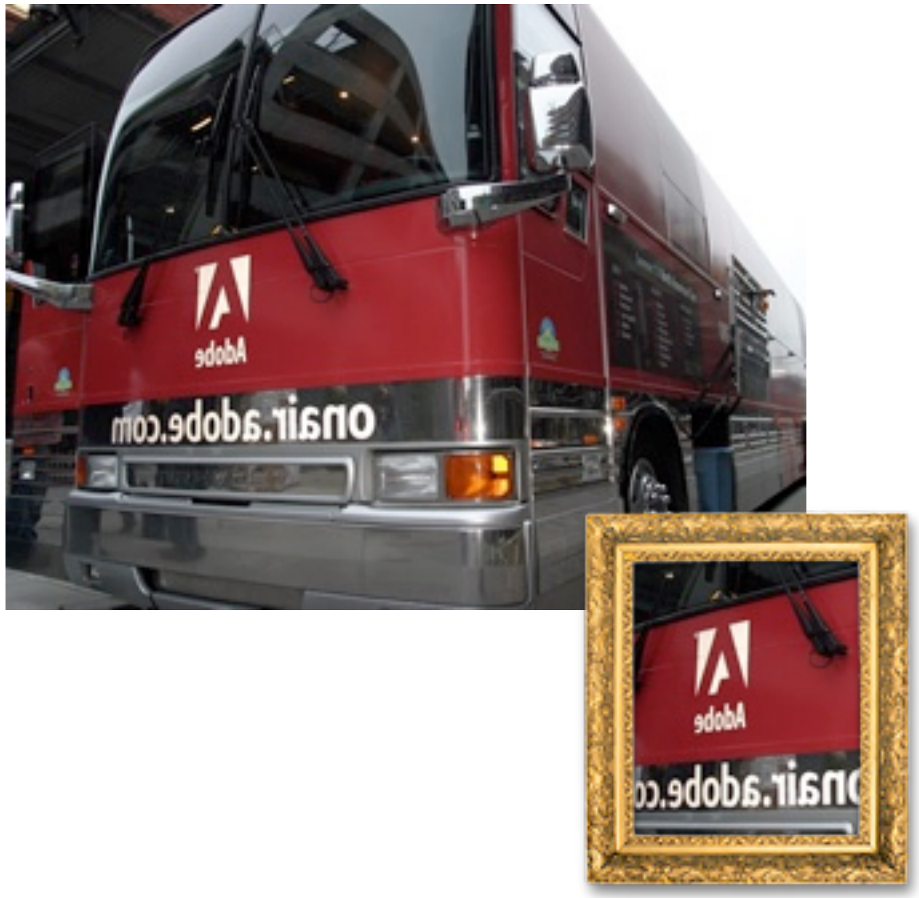
Embedded Data

```
var canvas = document.getElementById( 'canvas' );
var ctx = canvas.getContext( '2d' );
var img = null;

$( '#embed' ).click( function( evt ) {
    img = new Image();
    img.src = 'data:image/png;base64,iVB...';
    ctx.drawImage( img, 0, 0 );
} );
```

Pixel Pushing

```
context.drawImage(
     myImg,
     frame.x,
     frame.y,
     frame.width,
     frame.height );
```

Pixel Slicing

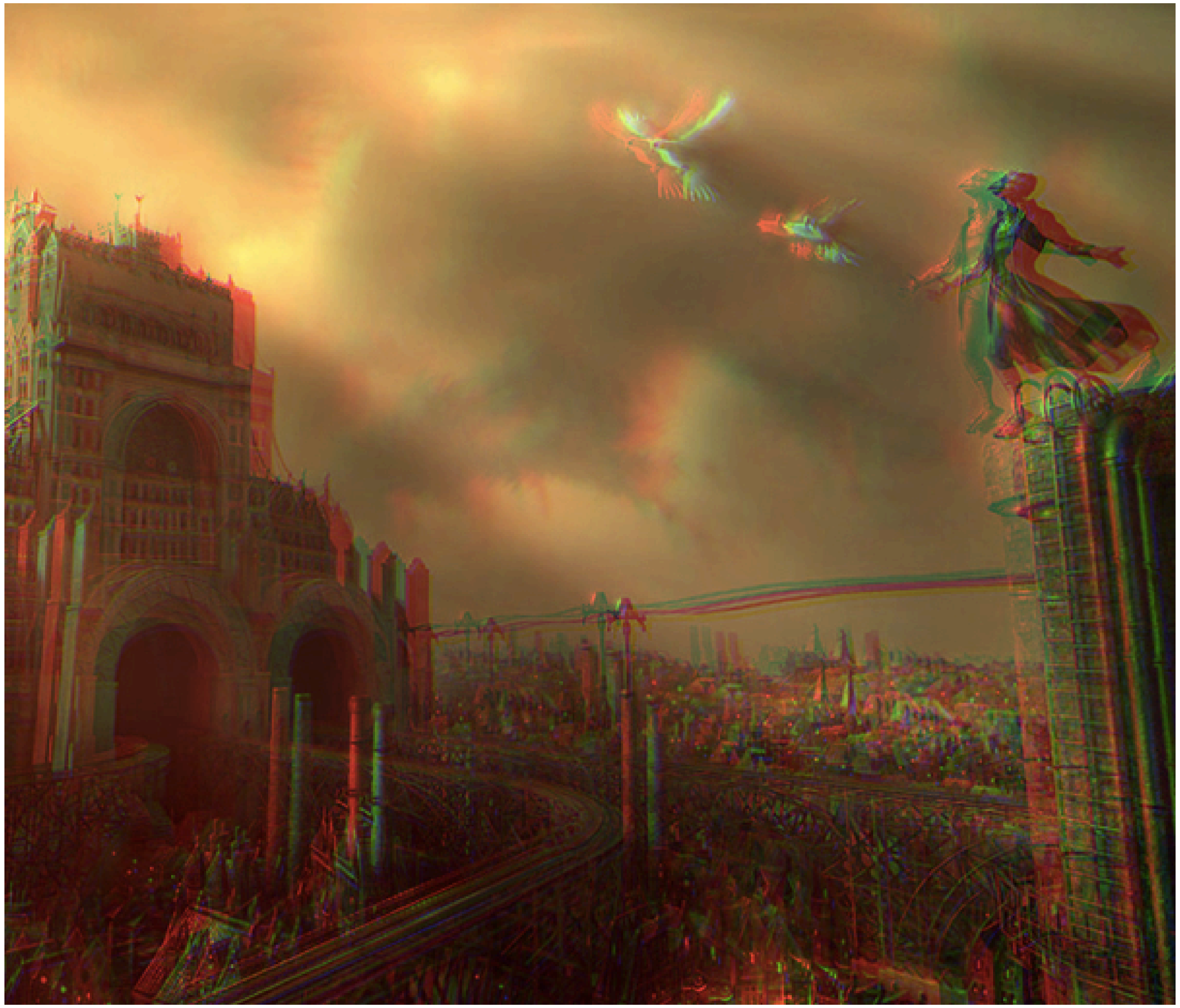


```
context.drawImage(
    myImg,
    frame.x,
    frame.y,
    frame.width,
    frame.height,
    22,
    21,
    frame.width,
    frame.height );
```

Pixel Dissecting

```
pixels = context.getImageData(
  0,
  0,
  640,
  480 );

for( var p = 0; p < pixels.data.length; p += 4 )
{
  red = pixels.data[p + 0];
  green = pixels.data[p + 1];
  blue = pixels.data[p + 2];
}
```

Pixel Dissecting



Beyond Pixels

```
var reader = new FileReader();

reader.onload = function( evt ) {
  var exif = new ExifInfo( evt.target.result );

  $( '<img src="data:image/jpeg;base64,' +
    btoa( exif.thumbnail ) +
    '"/>' )
    .appendTo( '#output' );
};

reader.readAsBinaryString( this.files[0] );
```

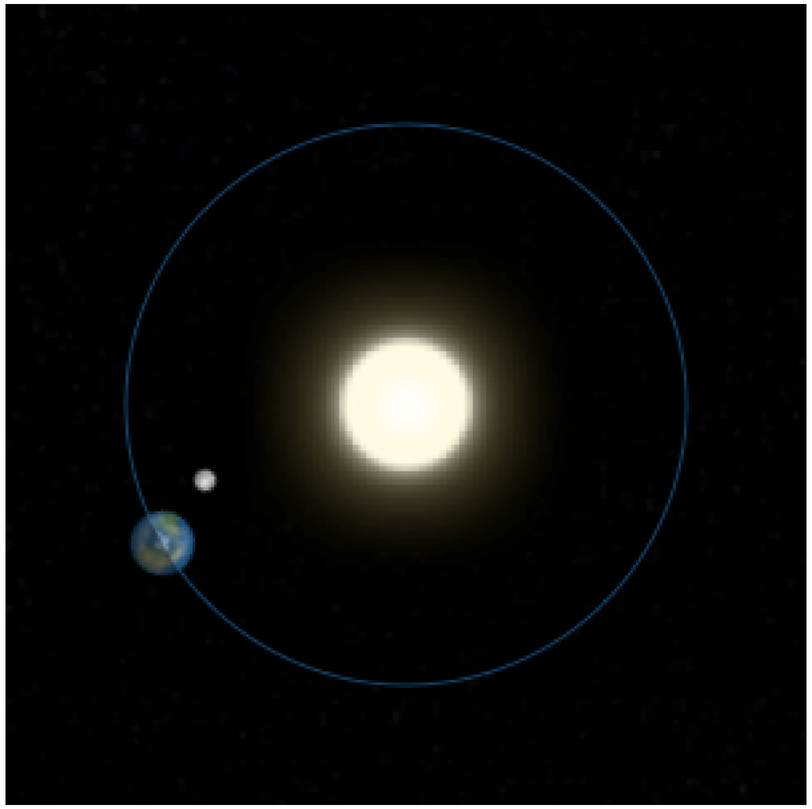
Beyond Pixels

Animation

Update on Interval

```
setInterval( function() {
  var time = new Date();

  context.clearRect( 0, 0, 300, 300 );

  context.rotate( ( ( 2 * Math.PI ) / 60 ) *
    time.getSeconds() +
    ( ( 2 * Math.PI ) / 60000 ) *
    time.getMilliseconds() );
  context.translate( 105, 0 );
  context.fillRect( 0, -12, 50, 24 );
  context.drawImage( earth, -12, -12 );
}, 100 );
```

Update on Interval

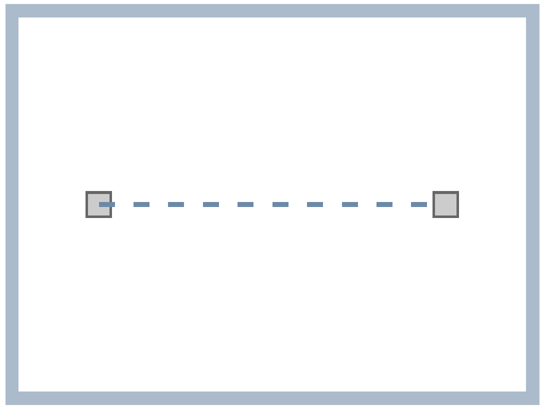


Tweening

```
JSTweener.addTween( position, {
    time: 3,
    transition: 'easeOutExpo',
    x: end.x,
    y: end.y,
    onUpdate: function() {
        context.clearRect( 0, 0, 640, 480 );
        // Draw updates to sprites
    },
    onComplete: function() {
        position = null;
        start = null;
        end = null;
    }
} );
```

Interaction

Tracking Points



What's clickable?
How do you know?
Canvas vs DOM...

Easel JS

```
var canvas = document.getElementById( 'game' );

stage = new Stage( canvas );

hole = new Bitmap( document.images[0] );
hole.x = 380;
hole.y = 120;
stage.addChild( hole );

ball = new Bitmap( document.images[1] );
ball.x = 20;
ball.y = 129;
stage.addChild( ball );

stage.update();

Ticker.setFPS( 24 );
Ticker.addListener( this );
```

Easel JS

Got Questions?

Kevin Hoyt
khoyt@adobe.com
Twitter: @krhoyt
AIM, YIM: parkerkrhoyt
http://blog.kevinhoyt.com