PhiloGL

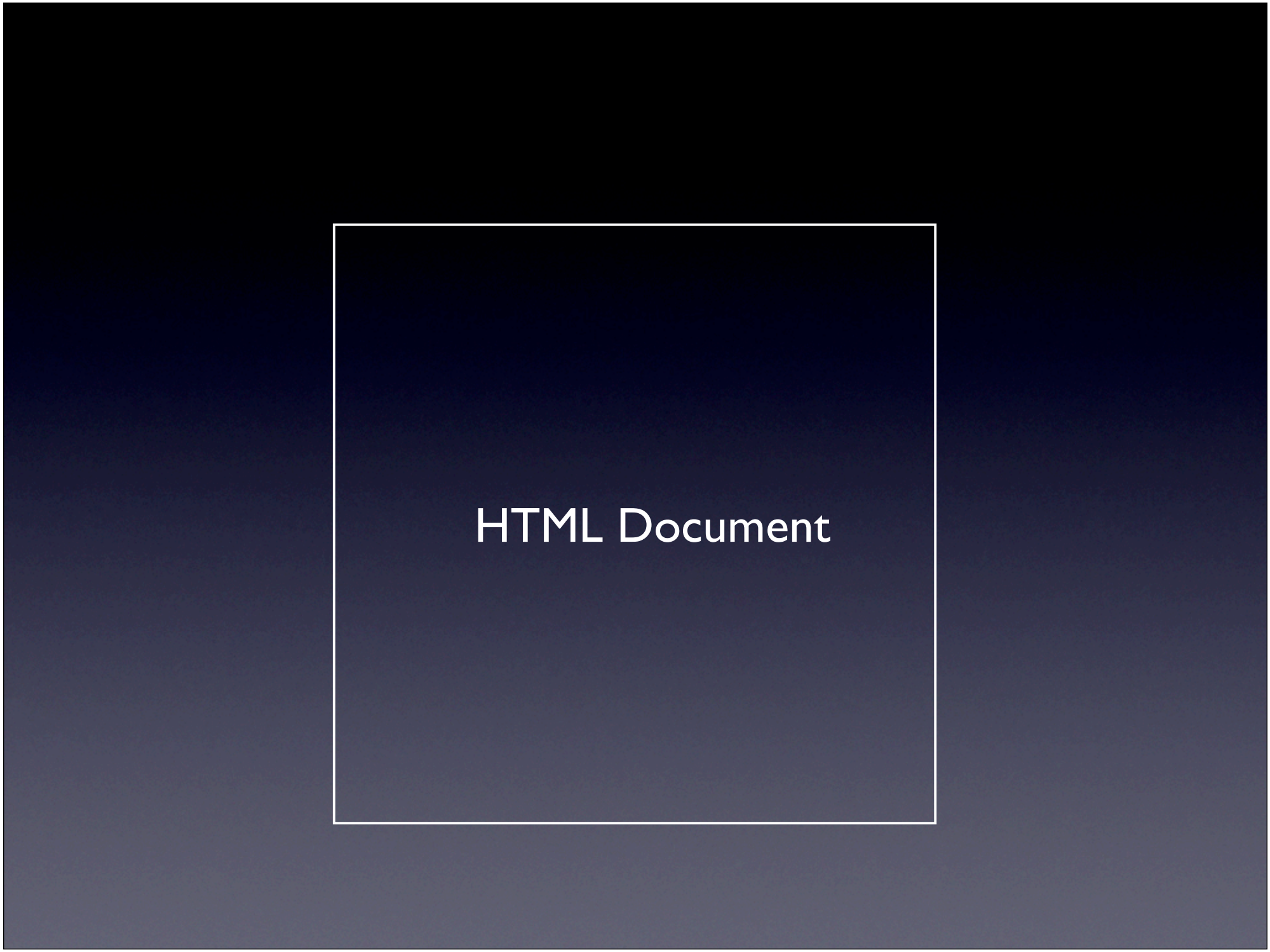
A WebGL Framework for
data visualization, creative coding
and game development

Nicolas Garcia Belmonte - @philogb - August 2011 - SIGGRAPH WebGL BOF


Sencha

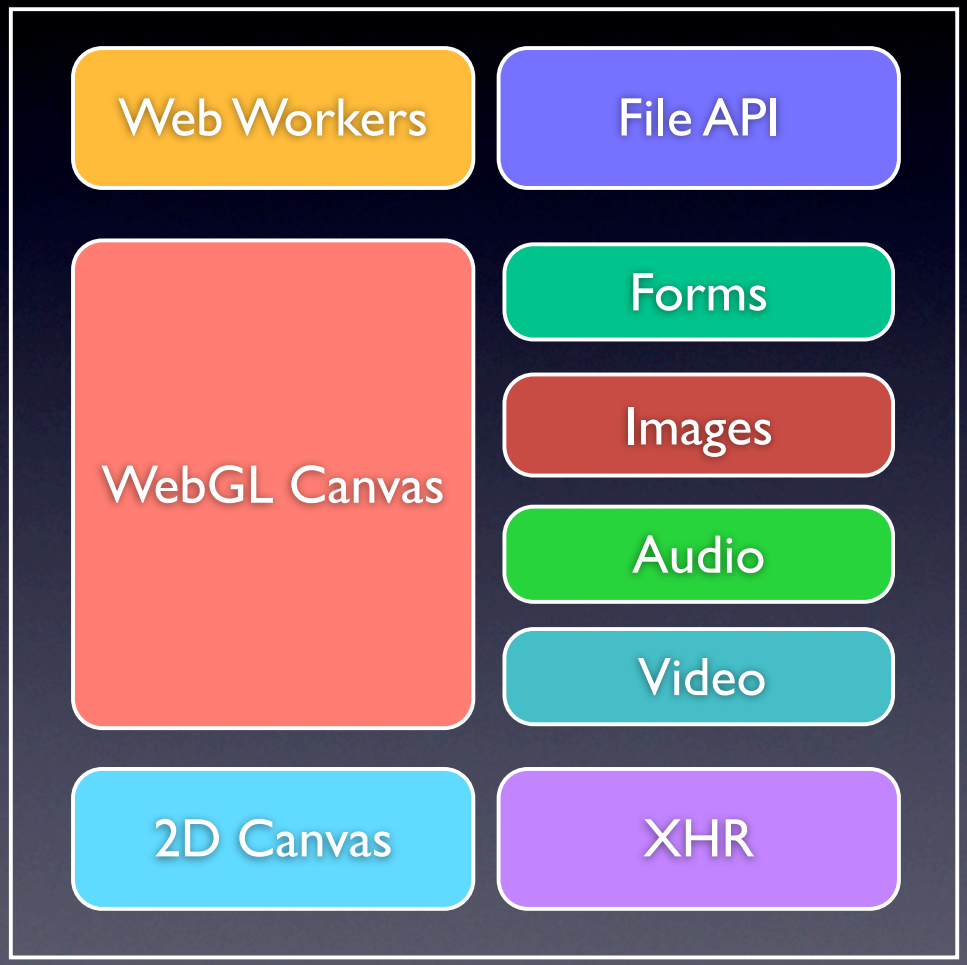
Data Visualization

JavaScript

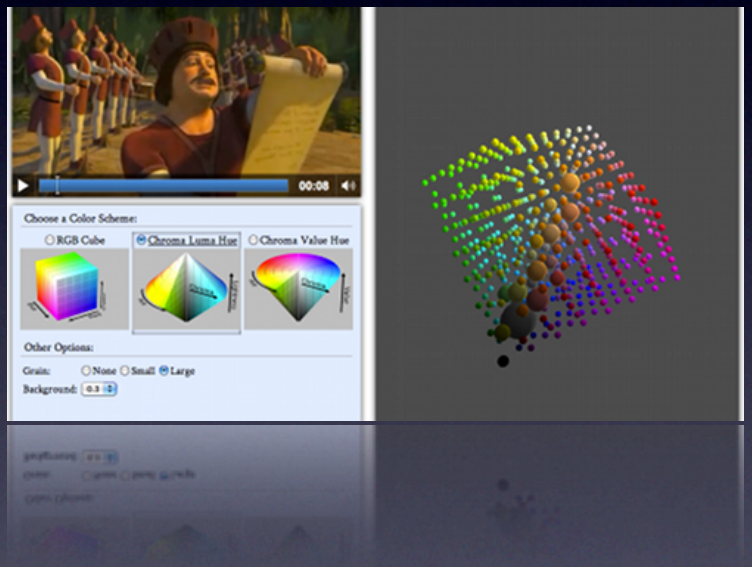

HTML Document

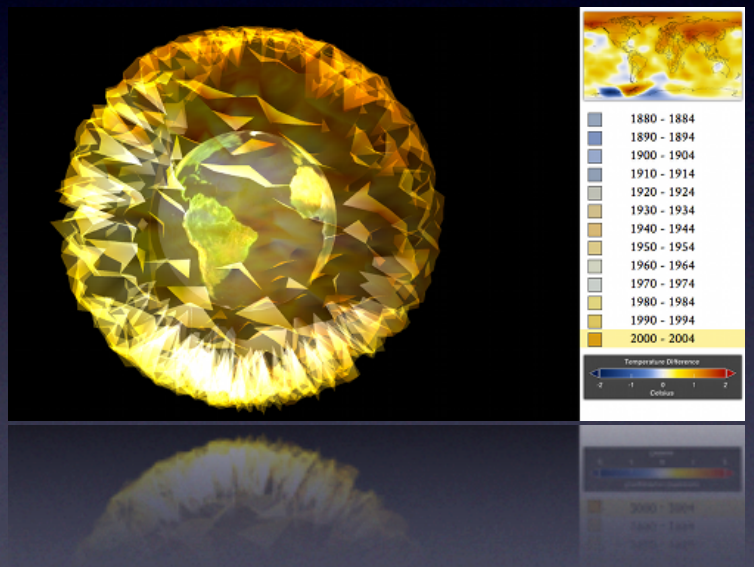
How most people see a WebGL app:

How I see a WebGL app:



Examples


00:08
Choose a Color Scheme:
RGB Cube
Chroma Luma Hue
Chroma Value Hue
Other Options:
Grain:
None
Small
Large
Background:


1880 - 1884
1890 - 1894
1900 - 1904
1910 - 1914
1920 - 1924
1930 - 1934
1940 - 1944
1950 - 1954
1960 - 1964
1970 - 1974
1980 - 1984
1990 - 1994
2000 - 2004
Temperature Difference
Celsius

PhiloGL

- Idiomatic JavaScript
- Rich Module System
- Flexible and Performance Focused

Idiomatic JavaScript

Concise & Expressive

Idiomatic JavaScript

Uniforms

Idiomatic JavaScript

?!

```
gl.uniform1i
gl.uniform2i
gl.uniform3i
gl.uniform4i
gl.uniform1f
gl.uniform2f
gl.uniform3f
gl.uniform4f
```

```
gl.uniformMatrix1fv
gl.uniformMatrix2fv
gl.uniformMatrix3fv
gl.uniformMatrix4fv
```

```
gl.uniform1iv
gl.uniform2iv
gl.uniform3iv
gl.uniform4iv
gl.uniform1fv
gl.uniform2fv
gl.uniform3fv
gl.uniform4fv
```

Idiomatic JavaScript

```
program.setUniform(name, value);
```

Idiomatic JavaScript

Buffers

Idiomatic JavaScript

```
//setup part...
var positionLocation = gl.getAttribLocation(program,
'position');
gl.enableVertexAttribArray(positionLocation);
var positionBuffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, positionBuffer);
var vertices = [
   0.0,  1.0,  0.0,
  -1.0, -1.0,  0.0,
   1.0, -1.0,  0.0
];
gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(vertices),
gl.STATIC_DRAW);

//render part...
gl.bindBuffer(gl.ARRAY_BUFFER, position);
gl.vertexAttribPointer(positionLocation, 3, gl.FLOAT,
false, 0, 0);
```

Idiomatic JavaScript

```
//setup part...
app.setBuffer('position', {
  size: 3,
  value: vertices
});

//render...
app.setBuffer('position', true); //bind
app.setBuffer('position', false); //unbind
```

Idiomatic JavaScript

Textures

Idiomatic JavaScript

```
//setup...
var texture = gl.createTexture();
var img = new Image();
img.onload = function () {
  gl.bindTexture(gl.TEXTURE_2D, texture);
  gl.pixelStorei(gl.UNPACK_FLIP_Y_WEBGL, true);
  gl.texImage2D(gl.TEXTURE_2D, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE, img);
  gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MAG_FILTER, gl.NEAREST);
  gl.texParameteri(gl.TEXTURE_2D, gl.TEXTURE_MIN_FILTER, gl.NEAREST);
  gl.bindTexture(gl.TEXTURE_2D, null);
};
img.src = "nehe.gif";

//bind...
gl.activeTexture(gl.TEXTURE0);
gl.bindTexture(gl.TEXTURE_2D, texture);
```

Idiomatic JavaScript

```
PhiloGL.IO.Textures({
  src: [‘image1.png’, ‘image2.png’, ...],
  onComplete: function() {
    app.setTexture(‘image1.png’, true); //bind.
  }
});
```

Idiomatic JavaScript

Programs

Idiomatic JavaScript

```
function getShader(gl, id) {
    var shaderScript = document.getElementById(id);
    if (!shaderScript) {
        return null;
    }

    var str = "";
    var k = shaderScript.firstChild;
    while (k) {
        if (k.nodeType == 3) {
            str += k.textContent;
        }
        k = k.nextSibling;
    }

    var shader;
    if (shaderScript.type == "x-shader/x-fragment") {
        shader = gl.createShader(gl.FRAGMENT_SHADER);
    } else if (shaderScript.type == "x-shader/x-vertex") {
        shader = gl.createShader(gl.VERTEX_SHADER);
    } else {
        return null;
    }
    gl.shaderSource(shader, str);
    gl.compileShader(shader);
    if (!gl.getShaderParameter(shader, gl.COMPILE_STATUS)) {
        alert(gl.getShaderInfoLog(shader));
        return null;
    }
    return shader;
}

var shaderProgram;
function initShaders() {
    var fragmentShader = getShader(gl, "shader-fs");
    var vertexShader = getShader(gl, "shader-vs");
    shaderProgram = gl.createProgram();
    gl.attachShader(shaderProgram, vertexShader);
    gl.attachShader(shaderProgram, fragmentShader);
    gl.linkProgram(shaderProgram);
    if (!gl.getProgramParameter(shaderProgram, gl.LINK_STATUS)) {
        alert("Could not initialise shaders");
    }
    gl.useProgram(shaderProgram);
}
```

```
Program.fromShaderSources
```

```
Program.fromShaderURIs
```

```
Program.fromShaderIds
```

Idiomatic JavaScript

Declarative

Idiomatic JavaScript

```
//Create application
PhiloGL('canvasId', {
  program: {
    from: 'uris',
    vs: 'shader.vs.glsl',
    fs: 'shader.fs.glsl'
  },
  camera: {
    position: {
      x: 0, y: 0, z: -50
    }
  },
  textures: {
    src: ['arroway.jpg', 'earth.jpg']
  },
  events: {
    onDragMove: function(e) {
      //do things...
    },
    onMouseWheel: function(e) {
      //do things...
    }
  },
  onError: function() {
    alert("There was an error creating the app.");
  },
  onLoad: function(app) {
    /* Do things here */
  }
});
```

Idiomatic JavaScript

```
app.gl
app.canvas
app.camera
app.scene
app.events
app.program
app.textures
app.framebuffers
app.renderbuffers
```

Module System

- Core
- Math
- WebGL
- Program
- Shaders
- O3D
- Camera
- Scene
- Event
- Fx
- IO
- Workers

Module System

Math classes have generic methods

```
var v1 = new Vec3(0, 1, 2),
    v2 = new Vec3(1, 2, 3);

v1.add(v2);

//or...
Vec3.add(v1, v2);

//May just be [x, y, z]
```

Module System

Workers - Divide & Conquer

```
var workerGroup = new WorkerGroup('worker.js', 10);

workerGroup.map(function(i) {
    //return a worker config
});

workerGroup.reduce({
  reduceFn: function(a, b) {
    //merge worker results
  }
});
```

Module System

Rich and mobile-ready event system

```
onClick
onRightClick
onTouchStart
onTouchMove
onTouchEnd
onMouseWheel
...
```

Module System

XHR and JSONP

```
new IO.XHR({
  url: 'http://some/query/',

  onError: function() {
    alert('There was an error');
  },

  onProgress: function(e) {
    if (e.total) {
        alert(e.loaded / e.total);
    }
  },

  onSuccess: function(data) {
    //handle data
  }

}).send();
```

```
IO.JSONP({
  url: 'http://some/query/',

  callbackKey: 'fn',

  onComplete: function(json) {
    //handle data
  }

});
```

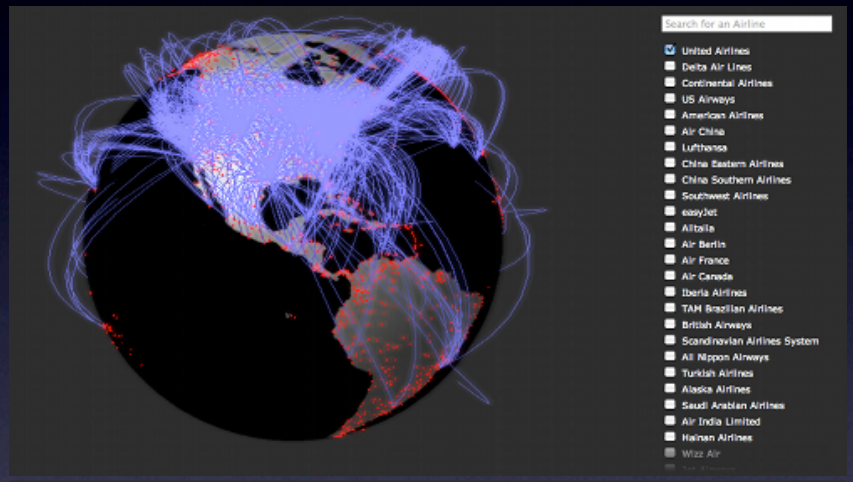
Module System

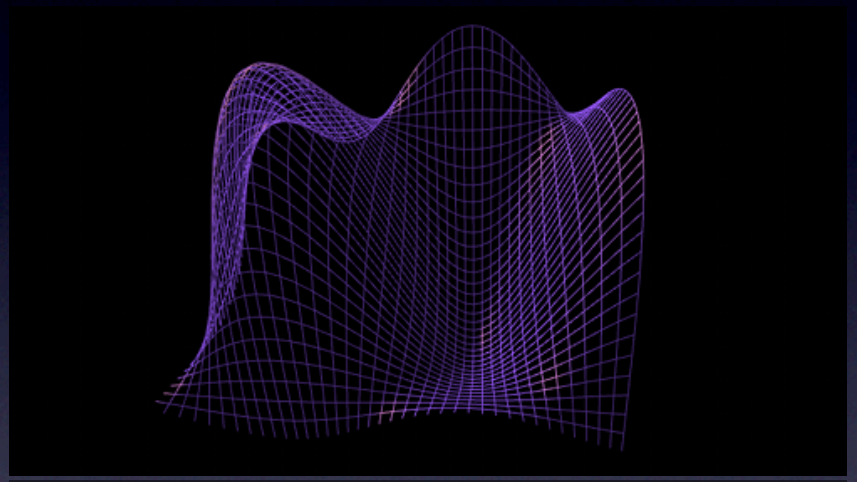
Tween

```
var fx = new Fx({
  duration: 1000,
  transition: Fx.Transition.Back.easeOut,

  onCompute: function(delta) {
    obj.height = Fx.compute(from, to, delta);
  },

  onComplete: function() {
    alert('done!');
  }
});
```

Fx.requestAnimationFrame

Other Examples





The most complete documentation and examples.

Thanks :)

PhiloGL

Project page:

http://senchalabs.github.com/philogl/

Twitter: @philogb