Getting started with WebGL and Three.js

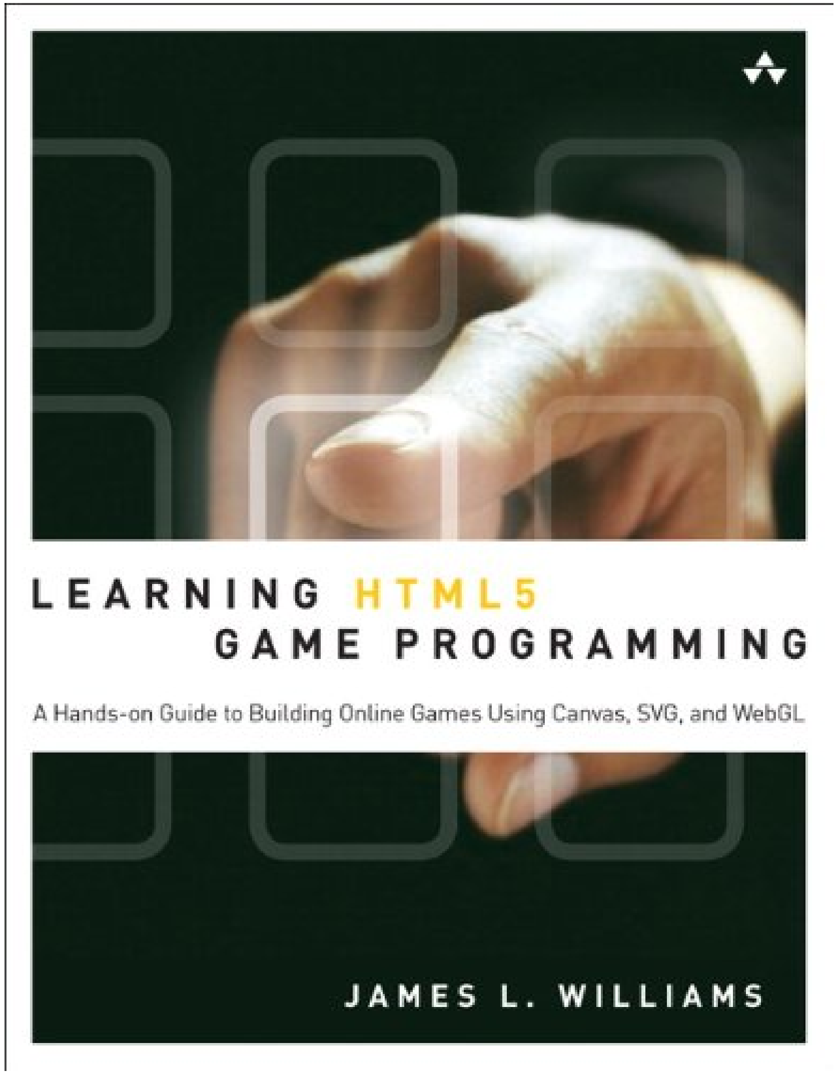
+James Williams @ecspike

TakeOff Conference (Lille, France)
January 2013

About Me

- Former student at Lille III - Univ. CDG
- Author of Learning HTML5 Game Programming
- Writing a new book on Three.js

Buy my book


LEARNING HTML5
GAME PROGRAMMING
A Hands-on Guide to Building Online Games Using Canvas, SVG, and WebGL
JAMES L. WILLIAMS

Buy my book ... in Chinese!

Three.js By Example

- LeanPub book
- Name your price
- Readers get updates as soon as they are pushed
- http://leanpub.com/threejsbyexample
- Github: http://github.com/jwill/threejs-by-example

Agenda

- What is WebGL/Three.js?
- Creating Meshes
- Lighting and Shading
- Demos

What is WebGL?

- Low-level 3D graphics context
- Uses canvas tag
- Hardware-accelerated
- Syntax based on OpenGL ES 2.0

Why aren't we using raw WebGL?

- Higher barrier to entry
- Lots of code
- Requires directly managing data structures

What is Three.js?

- 3D scenegraph library
- Abstracts the nastiness away from WebGL
- Renders to Canvas 2D, WebGL, and CSS3D*
- Can import models from popular 3D modeling apps

Camera

- Eye Point
- Field of Vision
- Near/Far Planes
- Target (LookAt) Point

- Up Vector

```
camera = new THREE.PerspectiveCamera(FOV, ASPECT, NEAR, FAR, [target]);
```

Creating Meshes

- Consists of two parts:
  - Geometry
  - Material

Built-in Geometries

- SphereGeometry
- PlaneGeometry
- CylinderGeometry
- CubeGeometry
- TextGeometry
- and several others

What is GLSL?

- Targets the GPU and graphics pipeline
- High level language with C-like syntax
- Passed around as strings
- Can be generated and compiled at run-time
- Referred to as programs (the combination of a vertex and fragment shader)

Vertex Shaders

- Run once per vertex in a mesh
- Can alter color, position, or texture coordinates

Example vertex shader

```
<script id="shader-vs" type="x-shader/x-vertex">
    #ifdef GL_ES
    precision highp float;
    #endif

    void main(void) {
        gl_Position = projectionMatrix * modelViewMatrix * vec4(position,
 1.0);
    }
</script>
```

Frament(Pixel) Shaders

- Run on every pixel in a mesh
- Can produce effects such as bump mapping and shadowing
- Only knows* about the pixel it is working on

Example fragment shader

```
<script id="shader-vs" type="x-shader/x-vertex">
    #ifdef GL_ES
    precision highp float;
    #endif

    void main(void) {
          gl_FragColor = vec4(0.0, 1.0, 0.0, 1.0);
    }
</script>
```

Shader Toy

Website enabling you to play around with GLSL shaders

http://www.iquilezles.org/apps/shadertoy/

WebGL Inspector

- Allows you to incrementally step through rendering
- View texture assets and GLSL programs
- Permits capturing individual frames
- Can be embedded or installed as a Chrome/Webkit extension

Github: http://benvanik.github.com/WebGL-Inspector/

Questions ?