WebGL

The HTML5/JavaScript 3D Computer Graphics API

Phillip Harmon
Principal, Consultant
Xelltech, Inc
http://www.xelltech.com

Co-Founder, CTO
CoutureMafia, Inc
http://www.couturemafia.com
Phillihp Harmon

Read my Tech Blog @ http://www.phillihp.com

Some of my work:
WebGL

Seminar Prerequisites
● HTML
● JavaScript
● Interest in 3D Graphics

This session is intended as demo and introduction to WebGL for an audience with basic programming experience or interested in beginning programming or 3D graphics.

Agenda
● Introduction
● What is WebGL?
● How to Learn WebGL
● Demonstration of WebGL Capabilities
● Future of 3D Web Graphics
● Create your own 3D Web Application by example
● Code explore WebGL Tetris

But, we will get Complex! Please ask questions.
What is WebGL?

OpenGL-based Ecosystem

Desktop Visual Computing
OpenGL and OpenCL have direct interoperability. OpenCL objects can be created from OpenGL Textures, Buffer Objects and Renderbuffers.

Mobile Visual Computing
Compute, graphics and AV APIs interoperate through EGL.

Roadmap Convergence
OpenGL and OpenGL ES are both streamlined, programmable pipelines. GL and ES working groups are working on convergence. WebGL is a positive pressure for portable 3D content on all platforms.

OpenCL
OpenMAX IL
EGL
OpenGL ES
OpenGL 3 Revolution Solution
WebGL

Mobile Visual Computing
Compute, graphics and AV APIs interoperate through EGL.
What is WebGL?

**WebGL:**
- JavaScript's 3D computer graphics API
- No need for plugins
- Must be browser supported
- WebGL is based on OpenGL ES
- WebGL's syntax is nearly identical to OpenGL

**Ecosystem:**
- **OpenGL** - Cross platform 2D/3D computer graphics
- **OpenGL ES** - OpenGL for Embedded Systems (iPhone, iPad, Android, BlackBerry)
- **OpenMax** - Audio, Video, Images
- **OpenIL** - OpenMax Integration Layer
- **EGL** - High performance graphics context management
- **OpenCL** - Parallel computing: CPU, GPU, Processor API
The Web Environment

Where does WebGL fit into the Web?
What is WebGL?

Q: Is WebGL HTML5?
A: Not exactly, WebGL extends HTML5, utilizing the HTML5 Canvas element.

Q: Where did WebGL come from?
A: WebGL came out of Canvas 3D experiments by Vladimir Vukićević from Mozilla, prototyped in 2006.

Q: How is WebGL managed?
A: WebGL is managed by Khronos Group, a working group, chaired by Ken Russel. The group includes Apple, Google, Mozilla, and Opera; although the invitation is open, not Microsoft.
WebGL in your Browser! Chrome!
WebGL Compatible Browsers

Desktop:
Google Chrome 9.0+
Mozilla Firefox 4.0+
Safari 5.1+ (*disabled*)
Opera 12.0+ (*planned*)
IE 10.0+ (*unknown*)

Mobile:
iOS Safari
Opera Mini
Opera Mobile
Android 2.3
Firefox Fenec (*beta*)

Please update Chrome or Firefox for future slides
WebGL Compatible Video Cards

- WebGL extends OpenGL
- Any Video Card that supports OpenGL 2.0
  - Adobe Photoshop CS4, CS5 utilize OpenGL

- Two Competing Advanced GPU Chipsets
  - ATI Radeon Series
  - NVIDIA GeForce, Quadro Series
  - Supported

- Integrated Graphics
  - Intel GMA 500+ Chipset
  - Supported
The WebGL Community

Giles Thomas
- !! to have programming background: Python, Java, Perl, C
- Not a JavaScript / OpenGL expert
- Describes WebGL as almost-finalized
- http://learningwebgl.com which he derived from http://nehe.gamedev.net/

Aleksandar Rodic
- Never learned C++
- Never learned OpenGL
- First programming language was JavaScript
- Focuses on Shading Technology
- http://aleksandarrodic.com/

Building on top of each others work
Enable WebGL for Firefox

● In the address bar, type in "about:config"
● Click [I'll be Careful]
● Filter for "webgl"
● Make sure the settings are as follows:
Enable WebGL for Chrome

Make sure that you have Google Chrome 9.0+
Demonstration of WebGL

3D Shapes
- http://webgl.pythonocc.org/
- http://web.chemdoodle.com/demos/molgrabber-3d

Google Search Volume by Language
- http://data-arts.appspot.com/globe-search

Google Cow
- http://bodybrowser.googlelabs.com/body.html#m=2

Let's try on Firefox
- http://www.silexars.com/demo/web/creation/
Time to get dirty!
WebGL Components

HTML DOM:
- JavaScript
- Canvas
  - HTML5
- WebGL JavaScript API

WebGL / OpenGL:
- Transforms and Viewport
- Matrix Manipulation
- Textures, Shading, Buffers
  - Store vertex data in RAM
- Identity (2D/3D)
- Rendering Pipeline
Understanding Transforms & Views

- 3D Graphics has 3 vertex points: x, y, z
- Where is my Camera?
  - 0, 0, 0
- How can I Transform my Viewport?
  - Translate(x, y, z)
  - Rotate(angle, x, y, z)
- Perspective

- Pong:
  - [http://pointmarker.com/webgl/test2.html](http://pointmarker.com/webgl/test2.html)
Matrix Pushing and Popping

**Important:** The purpose of the Matrix stack is to programmatically navigate around your 3D world.

**How?**
- `mvPushMatrix()`
  // Actions
- `mvPopMatrix()`

**What?**

![Diagram showing push and pop operations on a stack]
Understanding Textures

- Use Graphics to make your objects look nice
- Load the Graphic
- Bind (i.e. Attach in RAM)
- That first JavaScript looks ok.

```javascript
var neheTexture;

function initTexture() {
  neheTexture = gl.createTexture();
  neheTexture.image = new Image();
  neheTexture.image.onload = function() {
    handleLoadedTexture(neheTexture)
  }
  neheTexture.image.src = "nehe.gif";
}

function handleLoadedTexture(texture) {
  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, texture.image);
  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);
}
Understanding Textures

- Create Buffer
- Bind Buffer
- Setup Face Coordinates

```javascript
const cubeVertexTextureCoordBuffer = gl.createBuffer();
gl.bindBuffer(gl.ARRAY_BUFFER, cubeVertexTextureCoordBuffer);
var textureCoords = [
  // Front face
  0.0, 0.0,
  1.0, 0.0,
  1.0, 1.0,
  0.0, 1.0,

  // Back face
  1.0, 0.0,
  1.0, 1.0,
  0.0, 1.0,
  0.0, 0.0,

  // Top face
  0.0, 1.0,
  0.0, 0.0,
  1.0, 0.0,
  1.0, 1.0,

  // Bottom face
  1.0, 1.0,
  0.0, 1.0,
  0.0, 0.0,
  1.0, 0.0,

  // Right face
  1.0, 0.0,
  1.0, 1.0,
  0.0, 1.0,
  0.0, 0.0,

  // Left face
  0.0, 0.0,
  1.0, 0.0,
  1.0, 1.0,
  0.0, 1.0,
];
gl.bufferData(gl.ARRAY_BUFFER, new Float32Array(textureCoords), gl.STATIC_DRAW);
cubeVertexTextureCoordBuffer.itemSize = 2;
cubeVertexTextureCoordBuffer.numItems = 24;
```
Understanding Shading

- Shading Technology deals with calculation on how pixels are rendered inside your OpenGL world.
- Shader's are complex calculations, literal programs that are written for Graphics Processor Units.
- Pipelines
- Variable Variables
- Vertex, Geometric, Pixel
HTML5 Sounds

*Wait, this looks kind of familiar... better than that GL stuff*

```html
<audio>
<source></source>
</audio>

var el = document.getElementById('soundtoplay');
el.play();
el.pause();
el.volume = 1;
```

*WebGL is still just JavaScript. Utilize the environment, just the same.*
Create your own 3D Web Application

http://content.phillihp.com/

http://www.learningwebgl.com
Lessons: 2, 4, 13, 16
My WebGL Tetris project for fun and learning.

WebGL 0.2: http://phillihp.com/2011/06/11/webgl-tetris-v0-2-out/
Tetris Components

The Tetris Board

- Utilize a Multi-Dimensional Array for cube locations and colors.
- Below is a programmatic representation of a manual initialization.

```javascript
var mat = new Array();
mat = [
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
];
```
Tetris Components

The Objects Themselves:
- There are seven pieces used in Tetris
- All mapped using an array

```javascript
var objs = new Array();
objs = [
  [0, 0],
  [0, 1],
  [1, 0],
  [1, 1],
  [0, 0],
  [0, 1],
  [1, 0],
  [1, 1],
];
```
Easy Transforms to Draw Cubes

Drawing the Object:

```javascript
function drawObj(x, y, obj) {
  for (i = 0; i < 4; i++) {
    mvPushMatrix();
    newY = objs[curType][curPos][i][1] + y;
    newX = objs[curType][curPos][i][0] + x;
    if (newY < 20) {
      mvTranslate([newX * 2.0, newY * 2.0, 0.0]);
      drawCube(obj);
    }
    mvPopMatrix();
  }
}
```

```javascript
function drawCube(col) {
  gl.bindBuffer(gl.ARRAY_BUFFER, cubeVertexPositionBuffer);
  gl.vertexAttribPointer(shaderProgram.vertexPositionAttribute, cubeVertexPositionBuffer.itemSize, gl.FLOAT, false, 0, 0);
  gl.bindBuffer(gl.ARRAY_BUFFER, cubeVertexColorBuffer[col]);
  gl.vertexAttribPointer(shaderProgram.vertexColorAttribute, cubeVertexColorBuffer[col].itemSize, gl.FLOAT, false, 0, 0);
  gl.bindBuffer(gl.ELEMENT_ARRAY_BUFFER, cubeVertexIndexBuffer);
  setMatrixUniforms();
  gl.drawElements(gl.TRIANGLES, cubeVertexIndexBuffer.numItems, gl.UNSIGNED_SHORT, 0);
}
```
Tetris Events (Hey, JS Events!)

Setup your key events:

Left and Right move the object
Up flips the object
Down moves the object

Remember to `preventDefault()` and to add your `EventListener`.

And Yes, jQuery is fine with WebGL. Any Framework is fine.
The Refresh Frames

How often to redraw the content on the page.

```javascript
var lastTime = 0;

function animate() {
    var timeNow = new Date().getTime();
    if (lastTime !== 0) {
        var elapsed = timeNow - lastTime;
        rCube -= (75 * elapsed) / 1000.0;
    }
    lastTime = timeNow;
}

var lastDrop = 0;

function dropObject() {
    var timeNow = new Date().getTime();
    if (lastDrop !== 0) {
        var elapsed = timeNow - lastDrop;
        if (elapsed > dropRate) {
            moveObj("down");
            lastDrop = timeNow;
        } else {
            lastDrop = timeNow;
        }
    }
}

function tick() {
    drawScene();
    animate();
    dropObject();
}

function webGLStart() {
    var canvas = document.getElementById("the-canvas");
    initGL(canvas);
    initShaders();
    initBuffers();
    gl clearColor(1.0, 1.0, 1.0, 1.0);
    gl clearDepth(1.0);
    gl enable(gl.DEPTH_TEST);
    gl depthFunc(gl.LEQUAL);
    setInterval(tick, 15);
}

window.addEventListener("keydown", doKeyDown, true);
$function() {
    webGLStart();
});
```
Future for WebGL Tetris

- Score Count
- Next Piece
- Textures
- Score Keeping
- Multi-player via HTML5 Web Sockets
WebGL Competition

Dionysis Zindros

http://dionyziz.kamibu.com/3d/tetris/
WebGL is not Perfect

- **Gile's Thomas**, "Almost Finished 3D Graphics Standard"
- **WebGL**:
  - rendering does not completely match native Graphics Processing loads
  - is interpreted
  - has high processing requirements
- **When will WebGL Match OpenGL?**
  - compilation vs interpretation
  - top-side overhead on initial data transfer (internet)
Microsoft's Security Concern

The Mechanics of a WebGL Attack

1. A user visits a site where malicious WebGL script is present
2. The WebGL component uploads the specified 3D geometry and code to the graphics card
3. The geometry or code exploits issues in buggy or unpatched graphics drivers
4. The graphics hardware can be attacked causing the entire system to freeze or crash
Cause of WebGL

How?
- WebGL was a Mozilla experiment in 2006

Future:
- WebGL is driving future technologies to think outside the box
- Adobe Flash must compete
- Microsoft Silverlight in constant flux

Symptoms:
- JavaScript (ECMA) development is proving to be more powerful than ever thought before.
- WebGL proves true, non-compiled interpreted scripts to be "almost" as effective
- Server-Side JavaScript becoming very popular
- Extended ability to test, try, and learn
Future of 3D Web Graphics

Future:
● Web Served Video Games
● Online 3D Cad
● Highly Advanced Web Design

Competition:
● Flash Stage 3D vs WebGL
  ○ Code Security
  ○ Plugin Adaptation

Concerns:
● iOS5 not Supported
● IE10 not Supported
● Security

Known Games:
● Quake2
● Battlestar Galactica
Quake2 Compiled in WebGL

- Choppy
- Slow texture loading.
- Notice no sounds
Taste of Procedural Animation

The Origin of Mass
Final Demonstration

Aleksandar Rodic

- http://vimeo.com/20875622
- http://chrysaora.com/
Resources

Phillihp's Tech Blog:  
http://phillihp.com

Learn OpenGL/WebGL:  
http://learningwebgl.com  
http://nehe.gamedev.net/

WebGL Camp:  
http://webglcamp.com  
3 sessions as of August, 2011

Join the WebGL Public Mailing List:  
http://www.khronos.org/webgl/public-mailing-list/

Kronos Group:  
http://www.khronos.org/webgl/  
http://www.khronos.org/webgl/wiki/

Mozilla Involvement:  