UEMCON 2022

Aspects of VR
WebXR

Phillip G. Bradford

University of Connecticut

Outline

Setup

What is WebXR?

WebXR API

Setup

Ensure Python3 is installed – download and install

https://www.python.org/

```
Windows> mkdir workshop
Windows> cd workshop
Windows> python3 -m http.server 9000
```

What is WebXR?

Specification for WebVR/WebAR and WebMR

https://www.w3.org/TR/webxr/

Webpage – presentation and rendering engine

WebGL is a rasterizer

WebXR fundamentals

https://developer.mozilla.org/en-US/docs/Web/API/WebXR_Device_API/Fundamentals

https://addons.mozilla.org/en-US/firefox/addon/webxr-api-emulator/

WebGL Rendering

WebGLRenderingContext - Web APIs | MDN (mozilla.org)

https://github.com/mdn/dom-examples/tree/main/webgl-examples/tutorial

Start the server

Windows/MacOS> python3 –m http.server 9000

Avoids CTRL-O open file:/// which is not secure – see CORS discussion

WebGL

Rasterization engine

Converts vector images into pixel images

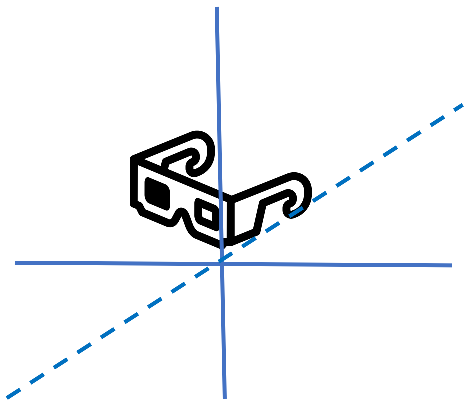
Leverages the GPU

Vector shader – computes vertex positions

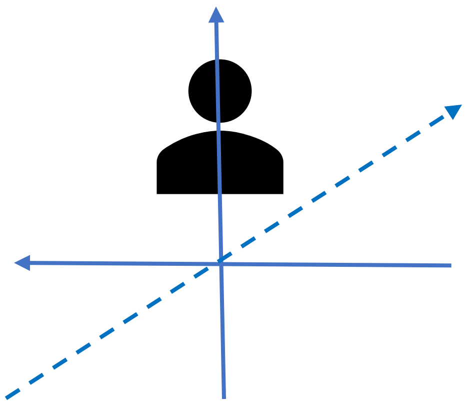
Fragment shader – computes colors of vertices

Uses OpenGL

3DoF or 6DoF



Rotation in 3D



Translation in 3D

<canvas></canvas> Tags: Rakesh Baruah

Drawing 2D and 3D graphics – render state and behavior

```
<body>
   <h3>WebGL</h3>
   <canvas id='myCanvas'></canvas>
</body>
<script type="text/javascript">
   const g1 = myCanvas.getContext('webgl');
   if (!g1) {
      alert('WebGL not available');
      console.log('WebGL not available');
   } else {
      alert('WebGL good');
      console.log('WebGL good');
   }
   </script>
```

canvas-gettransform-settransform

```
<canvas></canvas>
 <canvas></canvas>

 <script>
   const canvases = document.querySelectorAll('canvas');
   const ctx1 = canvases[0].getContext('2d');
   const ctx2 = canvases[1].getContext('2d');

   //ctx1.setTransform(1, .2, .8, 1, 0, 0); // h-scale, v-skew, h-skew, v-scale, h-trans, v-trans
   ctx1.fillRect(25, 25, 50, 50);  // fillRect(x, y, width, height)

   let storedTransform = ctx1.getTransform();
   console.log(storedTransform);

   ctx2.setTransform(storedTransform);
   ctx2.beginPath();
   ctx2.arc(50, 50, 50, 0, 2 * Math.PI);   // arc(x, y, radius, startAngle, endAngle)
   ctx2.fill();
 </script>
```

Matrix transformations

Magnitude

x.M	0	0
0	y.M	0
0	0	z.M

Matrix transformations

Skew

x	y	z

1	h.sk	0
v.sk	1	0
0	0	1

Matrices

https://glmatrix.net/