Enabling Direct WebGL in QtQuick 2

Pasi Keränen
Overview

About WebGL
  • Short History
  • What Can Be Done with WebGL?
Introducing QtCanvas3D
Using QtCanvas3D
  • Code Examples
  • three.js Port
Future Development
Q&A
About WebGL

What it is - History - Capabilities
About WebGL

**WebGL**

- A low level, state based, 3D vector graphics rendering API for HTML JavaScript
- Often described as “OpenGL ES 2 for the web”

**Khronos Group**

- Non-profit technology consortium that manages WebGL API (plus OpenGL/ES etc.)
- WebGL 1.0 is based on OpenGL ES 2.0
- Initial WebGL Standard Release in 2011
- Stable Release 1.0.2 in 2013
About WebGL

Widely Supported in Modern Browsers

Desktop:
- Google Chrome 9
- Mozilla Firefox 4.0
- Safari 6.0
- Opera 11
- Internet Explorer 11

Mobile:
- Safari on iOS 8
- Android Browser, Google Chrome 25
- Internet Explorer on Windows Phone 8.1
- Firefox for Mobile, Firefox OS, Tizen, Ubuntu Touch…
About WebGL

Scene Graphs
There are many scene graph libraries built on top of WebGL. They let you get started quickly without having to work with the low level API.

• three.js – http://threejs.org
• SceneJS – http://scenejs.org
• O3D – https://code.google.com/p/o3d/
• OSG.JS – http://osgjs.org/

Other Libraries & Resources
• See http://www.khronos.org/webgl/wiki/User_Contributions
• See e.g. https://developer.mozilla.org/en-US/docs/Web/WebGL
About WebGL

Projects

Some well known projects built with WebGL and HTML:

- Google Maps  – http://maps.google.com
- AutoCAD 360 – https://www.autocad360.com/

Chrome Experiments

Demonstrate beautiful things done on top of WebGL API:

- http://www.chromeexperiments.com/webgl/
QtCanvas3D

What it is – Why – Where to get
What is QtCanvas3D?

A Qt module that implements a 3D canvas component that can be added to a QtQuick scene.

You can get a WebGL like context API from the canvas component.

QtCanvas3D implements all the functions from WebGL 1.0.2 API spec, but is NOT tested for WebGL conformance due to the current nature of WebGL conformance tests.
QtCanvas3D

A Bit About Why?

WebGL is nice and productive environment for doing 3D
QtQuick is nice and productive environment for doing 2.5D UI
→ Combining these two makes for a very productive environment!

A lot of innovation going on around WebGL, allowing porting of that content to QtQuick makes a lot of sense. Allows developers to make 3D QtQuick applications that re-use existing 3D assets.

There are a lot of resources on the web to get started with WebGL.

About Qt3D 2.0…
QtCanvas3D

Requirements for QtCanvas3D Preview?
Qt 5.3 or later
OpenGL 2.0 or OpenGL ES 2.0 capable GPU

How to Get It?
• Go to https://codereview.qt-project.org/#/admin/projects/qt/qtcanvas3d
• Build (remember to add the “make install” step!)
• Open an example, build, run
• Start hacking..
QtCanvas3D

What is Different from WebGL?

WebGL talks about using HTML Image element, HTML Canvas element.
QtCanvas3D uses custom Texture3D element, Canvas3D element.
→ Just different names with same semantics

WebGL names objects as WebGLRenderbuffer, WebGLProgram...
QtCanvas3D names objects as Renderbuffer3D, Program3D...
→ None of the libraries we’ve tried have had problems due to this.

WebGL depends on TypedArrays (Float32Array etc.)
QtCanvas3D PREVIEW has custom TypedArray implementation.
→ This is an issue with PREVIEW, but will be fixed later.
QtCanvas3D

Implementation is based on the behaviour of typical WebGL apps.

Note: If you don't call canvas.getContext() in initGL(), then you must signal needRender to start the rendering loop!
Using QtCanvas3D

Overview on Usage
Using QtCanvas3D

Import and Declare the QtCanvas3D

```javascript
import QtCanvas3D 1.0
import "yourcode.js" as GLCode

Canvas3D {
   id: canvas3d
   imageLoader: textureImageLoader
   onInitGL: {
      GLCode.initGL(canvas3d, textureImageLoader);
   }

   onRenderGL: {
      GLCode.renderGL(canvas3d);
   }
}
```
Logging Flags

Two flags are available to configure how much logging is done by the implementation:

```cpp
Canvas3D { 
    id: canvas3d
    logAllCalls: true // Defaults to false
    logAllErrors: true // Defaults to true
}
```
Using QtCanvas3D

Declare the TextureLoader3D

TextureImageLoader {
    id: textureImageLoader
    function loadTexture(file) {
        textureImageLoader.loadImage("qrc:/QtQuick/"+file);
    }
    onImageLoaded: {
        GLCode.textureLoaded(image);
    }
    onImageLoadingFailed: {
        console.log("Texture load FAILED, "+image.errorString);
    }
}
Using QtCanvas3D

In the JavaScript code...

During initialization get the context, create GL resources, start textures loading:

```javascript
function initGL(canvas, textureLoader) {
    gl = canvas.getContext("canvas3d", {depth:true, antialias:true});
    textureLoader.loadTexture("sometexture.png");
}

Handle texture loaded events:

```javascript
function textureLoaded(textureImage) {
    if (textureImage.imageState == TextureImage.LOADING_FINISHED) {
        cubeTexture = gl.createTexture();
        ...
    }
}
```
Using QtCanvas3D

In the JavaScript Code...

Handle resizing of canvas, HiDPI displays and draw:

```javascript
function renderGL(canvas) {
  var pixelRatio = canvas.devicePixelRatio;
  var newWidth = canvas.width * pixelRatio;
  var newHeight = canvas.height * pixelRatio;
  if (newWidth !== width || newHeight !== height) {
    width = newWidth;
    height = newHeight;
    mat4.perspective(pMat, degToRad(45), width/height, 0.1, 500.0);
    gl.uniformMatrix4fva(pMatrixUniform, false, pMatrix);
  }
  gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);
  // YOUR OPENGL DRAWING CODE GOES HERE..
}
```
Using QtCanvas3D

Code examples:

Textured Cube

- A simple example from the QtCanvas3D example set.
- Very similar to WebGL tutorials.
- Shows combining QtQuick animations with QtCanvas3D rendering.

Car Visualizer

- Quite complex example.
- Content ported from http://carvisualizer.plus360degrees.com/threejs/ to QtCanvas3D.
- Uses QtQuick animations extensively.
Three.js on Top of QtCanvas3D

What it is - Where to Get - Tips for Porting
Three.js on top of QtCanvas3D

What?

• Three.js is one of the most active WebGL based scene graph libraries.
• Implemented 100% with JavaScript on top of WebGL
• Lot of WebGL content on the web uses three.js.
• Stable port on top of QtCanvas3D has been made from “67dev” version

How to Get It?

• Go to: https://github.com/tronlec/three.js/tree/stable
Three.js on top of QtCanvas3D

A Couple of Tips to Porting JavaScript Content

• Try to keep the delta small
  • Most of these libraries evolve rapidly
  • Implement wrappers for HTML stuff etc.

• All QtCanvas3D objects are QObjects exposed to QtQuick JavaScript Engine
  • You can’t dynamically add attributes or functions from JavaScript.
  • But you CAN add things with array access method.

        attribute.buffer.belongsTo = a;                    // Doesn’t work on Buffer3D objects
        attribute.buffer[BUFFER_BELONGS_TO_ATTRIBUTE] = a; // Works just fine!

• Qt.include() imports all declared functions from url into the current namespace

        function foo() { ...} //is a declaration
        foo = function() { ... } //NOT a function declaration!

• If running on mobile/embedded devices, take care of the memory usage!
Future of QtCanvas3D

Current Plans - Feature Requests
Future Development

Things We’re Aiming At:
Currently Working on top of Qt 5.3 and 5.4:
• Code cleanup
• Need to do more testing!
• Fix most severe bugs then do Technology Preview release on Qt 5.4

QtCanvas3D 1.0 will be part of Qt 5.5
→ Will compile and run only on Qt 5.5

TypedArrays in QtQuick JavaScript Engine will be part of Qt 5.5
→ QObject based TypedArrays will be removed from QtCanvas3D 1.0
Future Development

Proposed Things & Ideas (Not Committed To Implement Yet):

Move OpenGL calls to Render Thread

• Implement command queuing for all commands.
  • Implement syncing/stalling if a call that requires it (e.g. glReadPixels or glGetError) is made.

• Allows us to then add support for using any QSGTextureProvider as texture source (e.g. Image, ShaderEffectSource..)

• Would remove need to render via FBO (better performance, AA on mobile devices)

Profiling tools?

Extensions?
Recap

About WebGL
Introducing QtCanvas3D
Using QtCanvas3D
Future Development
Q&A
Thank you! Questions?

@pakerane on #qt3d FreeNode IRC channel
www.qt.io

See you there!