



Enabling Direct WebGL in QtQuick 2

Pasi Keränen

Who am I?



- Programming graphics since 1983
- Learned OpenGL 1.x in 1990's
- Joined Nokia's Graphics Knowledge Center in 2005
 - Tech lead from Nokia side to make Symbian graphics architecture more HW accelerator friendly.
- Participated Khronos Standardization
 - OpenKODE
 - EGL 1.4
 - OpenWF 1.0 (Chair of the Group)
- Joined Digia 2013
 - Part of the team that did Qt DataVisualization
 - Implemented QtCanvas3D
- In general around the office I'm the "graphics dude"



Overview



About WebGL

- Short History
- What Can Be Done with WebGL?

Introducing QtCanvas3D

Using QtCanvas3D

QtCanvas3D and three.js

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

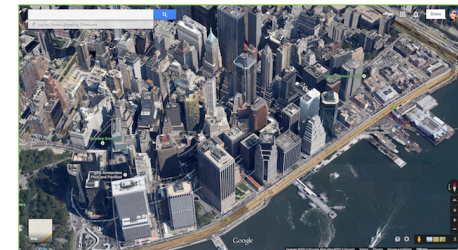
Some well known projects built with WebGL and HTML:

- Zygote Body – <http://zygotebody.com/>
- 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

QtCanvas3D



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

There is a lot of innovation around WebGL, allowing porting of that content to QtQuick on top of QtCanvas3D makes a lot of sense.

QtCanvas3D 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, these translate to QtCanvas3D as well.

Don't have to add the whole WebEngine/WebView to QML scene just to get WebGL based content running.

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?

- Source code only Tech Preview is included with Qt 5.4
- Or get it from <https://codereview.qt-project.org/#/admin/projects/qt/qtcanvas3d>
- Build (remember to do or add to Qt Creator 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..
→ Think of it as just different names with same semantics

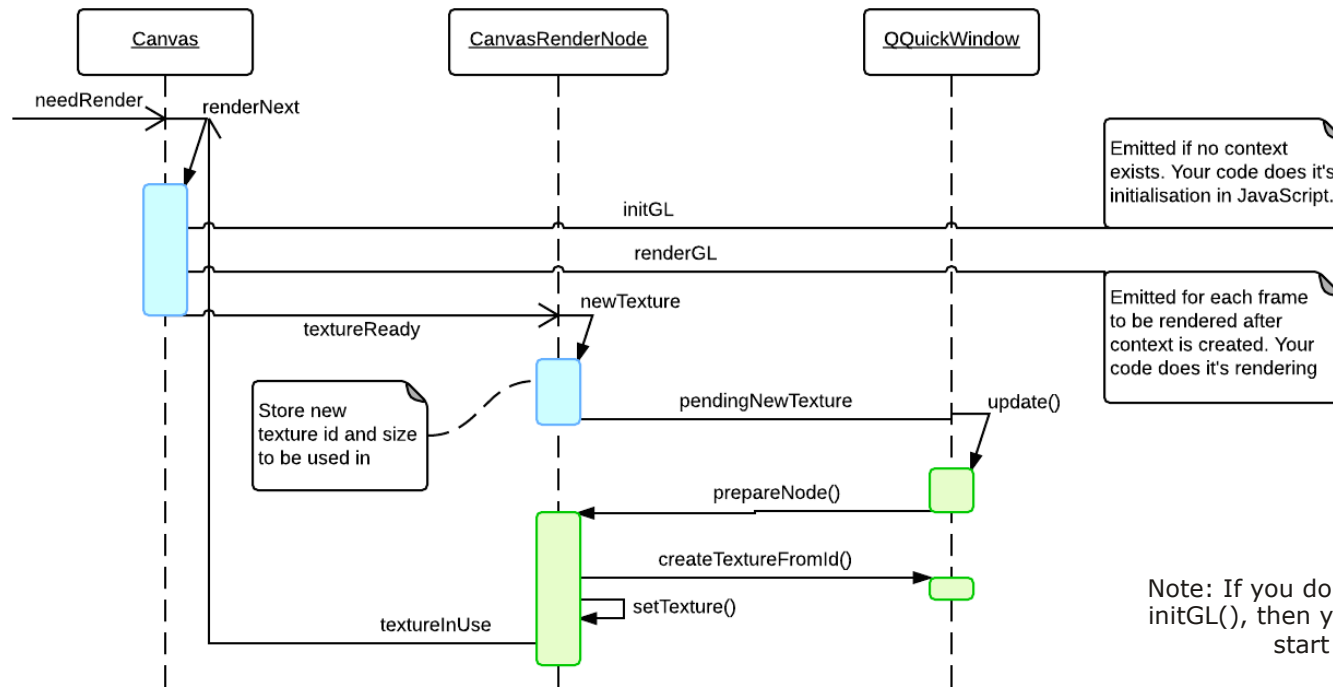
WebGL names objects as WebGLRenderbuffer, WebGLProgram...
QtCanvas3D names objects as Renderbuffer3D, Program3D...
→ None of the 3rd party libraries we've tried have had problems due to this.

WebGL depends on Java runtime environment TypedArrays (Float32Array etc.)
QtCanvas3D PREVIEW has custom TypedArray implementation.
→ Slight differences in the API, but this is an issue with PREVIEW and will be fixed in Qt 5.5.

QtCanvas3D



Implementation is based on the behavior of typical WebGL apps.





Using QtCanvas3D

Overview on Usage

Using QtCanvas3D



Import and Declare the QtCanvas3D

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

Canvas3D {
    id: canvas3d

    onInitGL: {
        GLCode.initGL(canvas3d);
    }

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


Using QtCanvas3D



Logging Flags

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

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

Using QtCanvas3D



In the JavaScript code...

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

```
function initGL(canvas) {
    gl = canvas.getContext("canvas3d", {depth:true, antialias:true});
    var texImage = TextureImageFactory.newTexImage();
    texImage.imageLoaded.connect(function () {
        cubeTexture = gl.createTexture();
        ...
    });
    texImage.imageLoadingFailed.connect(function() { ... });
    texImage.src = "qrc:/textures/yourtexture.png";
}
```

Using QtCanvas3D



In the JavaScript Code...

Handle resizing of canvas, HiDPI displays and draw:

```
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.uniformMatrix4fv(pMatrixUniform, false, pMatrix);
    }
    gl.clear(gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);

    // YOUR OPENGL DRAWING CODE GOES HERE..
}
```

Using QtCanvas3D



Code example:

Textured Cube

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



QtCanvas3D and three.js

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 like image elements 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!

Using QtCanvas3D



Code example:

Car Visualizer

- Quite complex example.
- Content ported from <http://carvisualizer.plus360degrees.com/threejs/> to QtCanvas3D.
- Uses QtQuick animations extensively.



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:

- Technology Preview release with Qt 5.4 (already in the Release Candidate package)
- Needs more content and more testing.

QtCanvas3D 1.0 will be part of Qt 5.5

→ Will compile and run only on Qt 5.5

Qt 5.5 introduces TypedArrays in the QtQuick JavaScript Engine

→ 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..)
 - You could have 2D QtQuick content running on a face of a cube.
- Would remove need to render via FBO (better performance, AA on mobile devices)

Profiling tools?

Extensions?

Three.js part of Qt and some tooling to get content flow in place?

Recap



About WebGL

Introducing QtCanvas3D

Using QtCanvas3D

QtCanvas3D and three.js

Future Development

Q&A



Thank you! Questions?

@pakerane on #qt3d FreeNode IRC channel