

Enabling WebGL

Catherine Leung

Seneca College of Applied Arts and Technology
70 The Pond Road
Toronto, Ontario, Canada
416-491-5050 ext. 3348

catherine.leung@senecac.on.ca

Andor Salga

Seneca College of Applied Arts and Technology
70 The Pond Road
Toronto, Ontario, Canada
416-491-5050 ext 3348

asalga@learn.senecac.on.ca

ABSTRACT

WebGL leverages the power of OpenGL to present accelerated 3D graphics on a webpage. The ability to put hardware-accelerated 3D content in the browser will provide a means for the creation of new web based applications that were previously the exclusive domain of the desktop environment. It will also allow the inclusion of features that standalone 3D applications do not have. While WebGL succeeds in bringing the power and low-level API of OpenGL to the browser, it also expects a lot of web developers, who are used to the DOM and JavaScript libraries like jQuery. This paper will look at how mid level APIs can help web developers create unique 3D content that is more than just duplicates of a standalone desktop application on a web page. We will present one such web application named Motionview, built with C3DL, that provides a new means for artist and motion capture studios to communicate with each other. We will also highlight some upcoming project ideas that make use of 3D browser technology in a way that would not have been possible in a desktop environment.

Categories and Subject Descriptors

H.5.1 [Multimedia Information Systems] Animations

General Terms

Design, Experimentation

Keywords

JavaScript Canvas, browser, 3D, WebGL

1. INTRODUCTION

When people think about 3D in the browser, one of the first thoughts that many people have is... cool, lets make "Doom on a web page". While it is possible to produce web versions of classic 3D games in the browser, 3D in the browser is much much more powerful than that. The web is not a one-way communication device. It allows not only the dissemination of information but

also allows collaboration and communication like no other medium. This uniqueness of the web has made it possible to create applications that are not possible in standalone environments. 3D content is just another form of this growing set of diverse applications.

WebGL is a powerful API that will soon be incorporated into upcoming releases of Firefox, Chrome and Safari. This API provides JavaScript bindings to Open GL ES functions making it possible to provide hardware accelerated 3D content on the web page. This API, while powerful is very low level and using it requires a good deal of understanding about 3D programming and 3D math. Web developers may find this to be unappealing or too much of a hassle due to its low level nature. One way to make it easier for web developers to make use of WebGL is to provide mid-level API's.

2. C3DL

C3DL is a JavaScript library that provides some basic 3D functionality for Web developers that wish to develop 3D content in the browser without having to go into the intricacies of WebGL and 3D math. C3DL has the following features:

- Collada model loading
- Mouse Picking
- Lighting System
- camera system
- Particle system
- Lines and dots
- Effects system that allows a swappable shader to be applied to alter its look. Currently we have the following effects implements:
 - cartoon (with or without outlines)
 - greyscale
 - solid colour
 - sepia
 - gooch

To see demonstrations of C3DL please check out our website at: <http://www.c3dl.org>

3. Motionview

One truly unique web application made with C3DL is Motionview. Motionview is a web application designed for better communication between artists and motion capture studios

Copyright is held by the author/owner(s).

WWW 2010, April 20–24, 2010, Raleigh, North Carolina, USA.
ACM 978-1-60558-799-8/10/04.

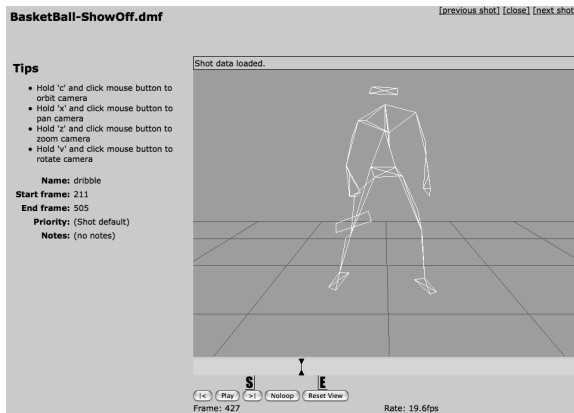


Figure 1: Screenshot of Motionview running in Firefox

Motion capture is the process of capturing the movements of actors and objects and importing those motions into animation software to make animations move realistically. It is used heavily by both the gaming and film industries. However, motion capture is expensive. Not only is the studio space and equipment expensive but the post capture processing of the data, called cleanup, is also highly labour intensive. After filming motions for a day, the cleanup technicians must manually fix all the data that were found to be in error. Furthermore, only a small portion of the motions ends up being used in the final product making the time and effort used to clean all the data a waste. Due to the fact that this cleanup process happens far away from the end users of the motions and there is no effective way to communicate with the technician which shots the artist wants and which ones can be left uncleaned. Motionview is designed to solve this communication problem. The web application allows the motion capture studio to upload a down sampled version of the mocap data (full versions are considered highly prized intellectual property and disseminating it over the web is considered unsafe by many companies). The data can then be viewed remotely in a 3D viewer, made using C3DL, by the animator. The animator can mark up the portions of the shots they wish to have cleaned and send that information back to the mocap studio.

This is a pretty standard web application for communication. 3D in the browser made it possible for the artist to view the shot from multiple camera angles and determine if it is the one they wish to use.

4. Processing.js

Processing is an open source programming language initially developed by Ben Fry and Casey Reas at MIT. The language was implemented with Java and using it over the web required a plug-in. In 2008, John Resig created a partial implementation of the 2D context for Processing using JavaScript and Canvas making it possible to use Processing in any web browser that supports the HTML5 <canvas> element without any plug-ins. WebGL has made it possible to port over the 3D portions of the Processing Language.

The Processing language was made to reduce the complexity of graphics programming more accessible in the era of C++ and Java. Processing.js brings this same idea to the web by reducing the complexity of drawing with WebGL and <canvas>

5. Conclusion

The addition of 3D to the browser environment will enable some unique opportunities for applications. For example, the ability to visually 3D scenes in the browser will allow for collaborative editing of 3D scenes. Higher level libraries such as C3DL and Processing.js will make it more accessible for web developers who do not wish to deal with the details of 3D math and low level API's

6. ACKNOWLEDGMENTS

Motionview was made possible with the support of the Ontario Media Development Corporation on behalf of the Ministry of Culture.



7. REFERENCES

- [1] Peter Callaghan, Catherine Leung, Andor Salga. C3DL blogs <http://www.c3dl.org>
- [2] Processing Website, <http://processing.org>
- [3] Processing.js Website, <http://processingjs.org>