W3bworld – Bringing L3DGE to HTML5

Chris Holman
6963420@student.swin.edu.au

L3DGEWorld

- Visualises the state of an environment through visual characteristics
- Each entity is represented by a single in-world object
- Requires the use of a sizeable dedicated client program
HTML5 – Taking L3DGE to the cloud

- Can a modern browser replicate L3DGEWorld?
  - 2D animation and 3D rendering (WebGL) from within the browser, using Javascript
  - Move from AJAX polling to full duplex communication
  - Significantly faster JS engines

What is this “WebGL”?

- A 3D renderer inside the browser
- Provides an OpenGL-like interface.
- Plenty of third-party libraries to ease development
What is this “Websockets”?
- Socket-like functionality
- Operates over “upgraded” HTTP connections

L3DGEWorld Architecture
W3bworld Architecture

- Server uses Node.JS – chosen for its Websocket support (Socket.IO)
- Client uses Javascript and Processig.JS
- Server ↔ Client communication is websocket
- Server accepts input and sends output using I3dgecomm 2.3
Demo – W3bworld

- Fullscreen API
- Camera movement
- Inspecting objects

Demo – input daemon

- Using Gpoll + W3bmon, we can emulate the functionality of LCMon.
- Gpoll reads data from the Green Machine supercomputer, takes key datapoints for each node and turns them in to a visual metric. It was written for the original LCMon.
Demo – input daemon

- Environmental sensors
  - Light
  - Temperature
  - Open doors

Demo – output daemon

- L3dgecomm allows for data sources to be interacted with from within the w3bworld client
- In this demonstration, the entity ID that we shoot dictates the angle the servo moves too.
- Utilising multiple ‘tools’, we can perform different actions on a node (not implemented yet)
Real applications?

- W3bmon, as demonstrated
- Environmental monitoring using Arduino (or similar)
  - Temperature around a DC
- Network monitoring
  - Home networks
  - Server farms, etc
- Device monitoring
  - ITS’ UPS monitor, LupsMon

Trivial to expand

- The application server is separated into two parts: a l3dgecomm server and the w3bworld server.
- Replacing l3dgecomm with a custom input is trivial

```javascript
var EventEmitter = require('events').EventEmitter;
module.exports = new EventEmitter();

function doThings() {
  var updateMessage = ['', entityID, metricID, messageType, Math.floor(Math.random() * 40).toString()];
  module.exports.emit("update", updateMessage);
}
setInterval(function() { doThings(); }, 5000);
```
Pros and cons?

- L3DGEWorld provides a much better looking experience (shaders, detailed levels, etc)
- W3bworld has no entity limits
- W3bworld client will work on any platform with a WebGL-enabled webbrowser – browser support is increasing

Browser support

- Firefox 4.0 and above, Firefox for Android
- Chrome 9.0 and above
- Safari (disabled by default)
- Opera 11 and 12, Opera Mobile for Android
- IE10
Where to?

- A new version of l3dgecomm to expand the capabilities (eg, set absolute position/rotation)
- Find a better/faster Javascript 3D rendering library, for better looking worlds and more represented entities
- Enhance as HTML5 becomes more powerful (eg, the upcoming pointer-lock API)
- Interface with device consoles from within the webpage (eg, SSH to server or router)