

**Computer Games Technology**

A JavaScript Runtime for Hardware Accelerated Applications

**Computing Honours Project (COMP10034) Interim Report**

William Taylor

**B00235610**

17/10/2016

**Supervisor: Paul Keir**

# Introduction

###### The Topic

In recent processing history there has been a seismic shift in technology. Processors have stopped getting faster at an exponential rate. Increasing the clock speed of processors has now been abandoned in favour of multicore processors and concurrent programming to better take advantage of modern hardware. What’s more as 3D and high resolution media has become popular we are now seeing the expansion of GPU’s being built directly on the CPU die in most modern chips. In this project we explore how best to provide access to this hardware from a high level language to gain the exponential performance increases to build fast and efficient software.

###### The Problem

The problem stems from the fact that this hardware is not exposed to most programmers. Instead for concurrent programs programmers tend to rely on the built in thread object now present in most programming languages. Plus, in higher level languages like JavaScript and Python this concept doesn’t exist having to rely on C++ libraries that do it for them.

In this case of JavaScript this is a big problem. The browser and the web now forms the base for any computer. The web keeps pushing innovation and has transitioned from pages of text to full blown cross platform applications.

###### The Solution

The solution is to demonstrate and provide that you can take control of low level hardware from within the high level environment that is JavaScript. A platform will be built that allows hardware accelerated applications to be written in JavaScript as a proof of concept to show that this low level access is possible to a great success. What’s more it will show how exposing low level API’s in a high level language environment.

# Technical Review

###### GPU’s

Why GPUs are important, dedicated + integrated

Lit review + blogs

###### Typed Arrays, Web Workers and SIMD

###### JavaScript + Web Assembly

Speed, portability, flexibility, simplicity

###### V8 JavaScript Compiler

V8 vs other compilers

###### OpenCL + OpenGL

Availability portability typical application use, GLSL, Shader language

# Current Progress

###### Development Technique

###### Gantt Chart Schedule

###### Prototype JavaScript Runtime

###### Demonstrations

# Plan for Completion

###### Finished Demonstration

###### Publically Available Source code

###### Bug fixing

###### Schedule

# Concluding Remarks

###### Excellent Progress

###### Looking forward to demonstrations + presentations

###### Possible future problems

# References

# Appendix