

# GEORGE CORNEY

The Cottage, Mutley Park House ♦ Plymouth ♦ PL3 4SE  
07542 045079 ♦ haxiomic@gmail.com

## MOTIVATION

---

I have a love of tackling interesting problems, especially when they occur at an intersection between technology, physics and design. My work has been enjoyed by over 1.6 million people and featured on sites like [The Next Web](#), [Gizmodo](#) and [FastCoDesign](#).

I specialize in web development and computer graphics, however, I'm never shy to grapple with a new domain to meet the demands of a project.

[I'm a keen proponent of open source](#) and you can find my contributes under the handle 'haxiomic'.

## EDUCATION

---

### University of Sheffield

B.Sc in Physics with Astrophysics ♦ *First Class*

## EXPERIENCE

---

### Alchemy VR

*Java, C++, Android*

June 2015

*Contract Work*

- [Alchemy VR](#) is a recently formed branch of Atlantic Productions. Their VR premier "Life on Earth" (narrated by David Attenborough) was to be shown on 80 Gear VRs in the Natural History Museum from the 12th of June (2015).
- Alchemy asked me to solve a number of critical problems two weeks before their premier deadline. I was tasked with developing a custom VR video player and bypassing the built in oculus home store without resorting to rooting the devices.
- The custom VR video player was developed using C++, Java Native Interface and Oculus's Mobile SDK and the bypass was developed with Java and the Android API.

### fffunction

*JavaScript, backbone.js, HTML, SCSS*

April 2015

*Contract Work*

- [fffunction](#) is a digital design agency in the Southwest with clients that include [Roland UK](#) and the Bristol museums group ([BMGA](#)).
- I was brought in to contribute to a browser-based book reader and preview app (commissioned by Oxford University Press). The app was developed with backbone.js, node.js and Grunt.
- My role involved developing a page layout engine and viewer thumbnail alongside bug fixes.

### University Hospitals of Leicester

*JavaScript, backbone.js, HTML, LESS, e-Learning*

Sept 2014 - Feb 2015

*Contract Work*

- Imran Aslam of University Hospitals of Leicester commissioned the development of a browser-based e-Learning app.
- The app (developed with backbone.js) was designed to aid teaching medical students the correct procedures and behaviors when working in a team on a ward.

## Hive

Nov 2013 - Jul 2014

*Objective-C, OS X Reverse Engineering, UI & UX, JavaScript*

*Startup*

- Hive was a team collaboration app I worked on with a small group during university. It was an experiment in developing the ideal collaboration tool. The philosophy was that team cohesion could be improved by reducing boundaries between computers; the goal was to be able to push content (including running programs) from one device to a teammate's immediately and intuitively (in a similar manner to moving windows between multiple displays).
- The project won the Cisco Open Collaboration Challenge and was accelerated for 3 months at [dotforge](#).
- I left the project after the release of OS X Yosemite which contained features (Continuity and Handoff) which competed with our core technology 'State-sharing'.
- My role in the project was development of 'State-sharing' (which involved reverse engineering OS X's state saving feature), UI & UX design, and the development of the native OS X app.

## OPEN SOURCE PROJECTS

---

### SuperSL

Mar 2015 - Present

*Haxe, GLSL, JavaScript, C, Context Free Grammars, LALR*

*Work in Progress*

- [SuperSL](#) is a shader programming language built for the web. SuperSL is a superset of GLSL with additional language features designed to resolve some of GLSL shortcomings and improve the process writing shaders for WebGL.
- In order to strike a balance between GLSL's strictness and Javascript's flexibility it introduces features such as compile-time type inference, function inlining and shader modularity.
- SuperSL can be used to generate GLSL either at project compile time (like LESS or SASS) or at runtime. To make this possible with a single codebase SuperSL is written in haxe and compiled to (among others) Javascript, C++ and Java.

### WebGL Fluid

September 2014

*GLSL, Haxe, Lime, JavaScript, C++, WebGL*

- [This project](#) is a GPU fluid and particle simulation written in Haxe and GLSL, targeting HTML5 for browsers and C++ for desktop and iOS. The simulation solves the Navier-Stokes equation for incompressible flow over a grid with the Jacobi method and uses the velocity field to advect over 1 million particles.
- The motivation for this project was to explore using WebGL for high performance physics simulations and to investigate the performance factors involved.
- It's been played with approximately **2 million times** by **1.6 million users**
- It has reached the front page of [Reddit](#) and featured in articles on
  - [FastCoDesign](#)
  - [The Next Web](#)
  - [Gizmodo](#)
  - [engadget](#)