Craig Macomber

6520 16th ave NE, Seattle, WA 98115
Email: CraigM@CraigM.info Cell: 206-432-7475

More information on most of these topics is available on my site: CraigM.info

EDUCATION

University Of Washington
Expected graduation: Fall 2012
Working on B S in Computer Engineering, Minor in Physics, and Minor in Mathematics
GPA: 3.74

SKILLS

Skilled in: Python, Java, Go, C, Cg, Cython, RealBasic
Familiar with: C++, Verilog, Scheme, JavaScript, Bash, ML, Assembly (Y86), machine code (Z86), LabView, R, ActionScript
Areas of focus: Algorithm design, Graphics, Data processing, Parallelism, Concurrency, Networking, Project design

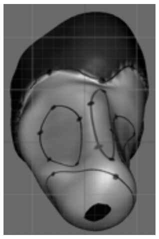
EXPERIENCE

Software Engineer Intern, Google **Summer 2012**

I worked on Google's doubleclick search product. This work focused on updating and refactoring parts of the backend servers, and writing a couple of new tools. (Java, networking)

Researcher, University of Washington **Summer 2011**

I developed a system to use a camera to track a WISP programmable RFID tag. (Java, LabView, networking)



Software Developer, Provel Inc. **2009-Current**

- Maintenance, upgrades and deployment of Provel Carve, an application for orienting and configuring 3D models for carving, and generating motion paths and sending them to the carver over serial. (RealBasic)
- Sole developer of a Prosthetics CAD tool (pictured). (Python, Cython)

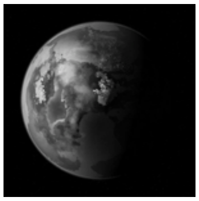
Programmer, Sunburst Sensors LLC **2009-2010**

I worked on Sunburst's SAMI2 marine data logger. Specifically I wrote the data processing and presentation code and user interface. As the logger is extendable, I implemented a generic and error tolerant data processing framework, as well as some of the actual use cases. It supports live streaming of data, as well as parsing large log files. (RealBasic)

Technical Writer, RLT Industries **Summer 2008**

I tested educational physics kits, and wrote assembly and usage manuals for them. This included all writing, photography and page layout, and educational coverage of physics concepts and experiments.

OTHER PROJECTS

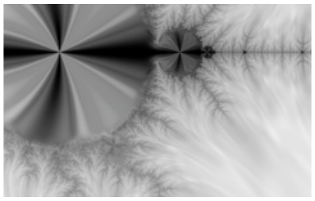


Planet Renderer

I implemented a fully procedural realtime interactive planet generator using CG shaders on the GPU. This included supporting deep zooms and working around the associated floating point precisions issues. (Cg, Python)

Shader Metalanguage

I designed and implemented a domain specific declarative shader metalanguage within python to allow easy coding of shader generators for Panda3D graphics projects. (Cg, Python)



Fractal Renderers

I've implemented many fractal generation algorithms, including GPU and parallel CPU based algorithms. I've written them in C, Cg, Glsl, RealBasic, Python, Java, Processing, and Go. I've implemented escape distance algorithms (including the Mandelbrot pictured), higher dimensional fractal flame experiments, interactive L-system editors and others. One of my renderers was web based and relied on user selection to evolve a population of fractals.

Panda3D Terrain System

I implemented a a system generating and displaying infinite procedural and semi-procedural terrain in the Panda3D game engine. Includes procedural tree and fern generators, and some GPU accelerated content generation. (Python)

2011 ACM Programming Competition

My team placed first at our site in the ACM intercollegiate programming contest.