



Resume / Cover Letter / Sample Code

Updated May 2009  



Henry Smith

henry@enigmasoftware.ca

home: (780) 642-2822

cell: (780) 884-7044

Available September 2009

View my
profile on



15 people have recommended Henry

Currently in: *Edmonton, AB Canada*

Objective

- Senior programmer position at a world-class game development studio
- Interested in GUI programming/design, rapid prototyping, scripting languages, and engine architecture

Skills

- Eight years of game industry experience, plus many more as a hobbyist
- Expert C++/C programmer
- Development experience on PC, Mac, Console, Handheld, and Flash platforms
- Published indie/shareware developer

Languages

- Expert in C++/C, ActionScript 2
- Familiar with Ruby, Python, Lua, JavaScript, UnrealScript, XML
- Exposure to various teaching languages (Scheme, ML, Haskell, Eiffel, Prolog)

Tech

- Familiar with Scaleform GfX, Flash, Unity, STL, Boost, Perforce
- Exposure to Unreal Engine, NetImmerse/Gamebryo, iPhone, OpenGL

Experience

BioWare

Senior Programmer
Edmonton, AB Canada
2004—present

- Senior GUI programmer on Dragon Age: Origins
- Architected and maintained a GUI framework in C++ and Flash/ActionScript used for all game UI
- Mentored a junior programmer
- Spearheaded a “Study Lunch” group for sharing technical knowledge and expertise
- Member of the (internal) Technology Architecture Group
- Worked with many aspects of the game engine including: graphics, input, game-rules, scripting, tools

Irrational Games

(now **2K Boston**)
Programmer
Boston, MA
1999—2002

- Designed and built several major game systems for a PS2 3rd-person action title, using C++, scripting languages, and middleware (LithTech/Unreal). I worked closely with four other programmers and a small team of designers, artists, and testers.
- Built tools and enhancements in C/C++ for a proprietary PC 3D game-engine (The Dark Engine)

Enigma Software

Sole Proprietor
1996—present

- Built a shareware game for Mac, PalmOS, and Flash: www.enigmasoftware.ca/squish
- Created all tools, code, art, audio, design, and a resource pipeline to make a finished game on three platforms
- Working on two new games for Mac/PC/iPhone using the Unity game engine

Education

University of Waterloo

Waterloo, ON, Canada
B.Math in Honours Computer Science

Relevant courses:

- Introduction to Computer Graphics
- Introduction to Artificial Intelligence
- Developing User Interfaces
- Principles of Programming Languages
- Algorithm Design and Analysis

Interests

- All aspects of game creation/design
- Human computer interaction and user interfaces
- Board games, word games, poker
- Ultimate frisbee
- Hiking
- Piano

Henry Smith

henry@enigmasoftware.ca

9828 89 Ave
Edmonton, AB
T6E 2S4
Canada

To Whom It May Concern:

I'm a strong generalist programmer with over eight years of game industry and hobbyist experience. I'm looking for a full-time position, preferably starting no earlier than June 2009.

I grew up with computers and have always had a passion for creating games. Even before I started programming I made board games, card games and puzzles. I remember checking out library books and typing in hundreds of lines of BASIC code to make text adventures. I experimented with MUDs (and MOOs) as a "programmer". My first real games were written in HyperCard and I found that I loved combining stories, artwork, sound, logic, and simulation to make them come alive. I still think it's magic.

Before starting university I released a shareware game for the Mac (a puzzle game inspired by Lemmings), which was picked up by a publisher. I've since ported the game to PalmOS and Flash: www.enigmasoftware.ca/squish

Now I consider myself to be a very strong C++ programmer. I'm comfortable with object-oriented programming, generic programming, templates, the STL, Boost, and Design Patterns. I'm a strong believer in using the best tools for the job, and will readily use engines, middleware, and scripting languages as appropriate. In fact, I advocate using high-level/domain-specific languages wherever possible. When it's not possible, I'm no stranger to low-level memory and performance optimization.

I document my code, I have a keen eye for detail, I'm very organized, I love a challenge, I have my own opinions, I take pride in my work, I play well with others, and I don't need to be spoon-fed.

My notable work experience includes working for **Irrational Games** in Boston (now 2K Boston), and **BioWare** in Edmonton, Alberta.

At Irrational I worked with a small team of programmers on a 3rd-person survival horror game for the PlayStation2 and was involved in almost all aspects of the game, personally designing several critical systems. I worked closely with artists, animators, sound engineers, and testers. Unfortunately the project was cancelled, but the experience I gained there was invaluable.

At BioWare I started out building a general purpose GUI framework for the Technology Architecture Group. I then transitioned to Dragon Age and became the Senior GUI Programmer on the project. I was responsible for the design and maintenance of the GUI system and implemented or oversaw almost all GUIs in the game. We used a combination of C++/Flash/ActionScript and the Scaleform GfX library to build the system. I also provided support and mentorship to junior programmers.

You can find some recommendations on my LinkedIn profile:

<http://www.linkedin.com/in/henrysmith>

Please let me know if I would be a good fit with your team, and thank you for your time.

Henry Smith

henry@enigmasoftware.ca

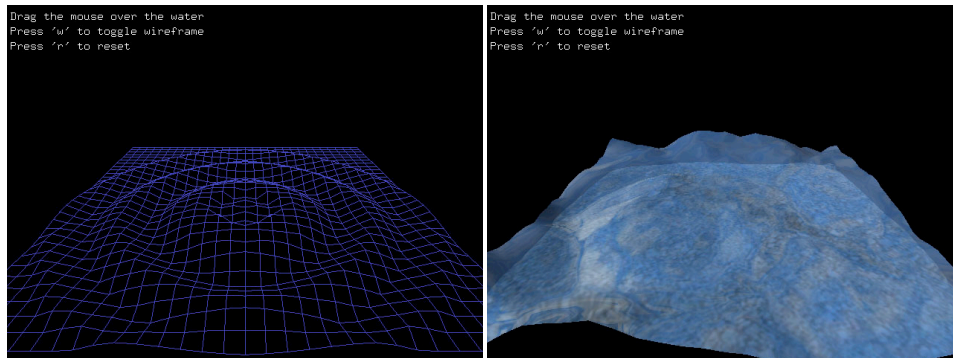
SpringWater

Files

- [SpringWater.zip](#)

Description

This is an OpenGL simulation of a body of water using a mesh of springs. It also simulates reflection and refraction. It's written in C++ using GLUT. Mac and Windows binaries are provided.



MultiNode (unrelated to SpringWater demo above)

Files (HTML-formatted code)

- [main.cpp](#)
- [MultiNode.cpp](#)
- [MultiNode.h](#)
- [MultiNodeBase.cpp](#)
- [MultiNodeBase.h](#)
- [MultiNodeImpl.cpp](#)
- [MultiNodeImpl.h](#)
- [MultiNodeRef.h](#)
- [NodeImpl.h](#)

Description

This is a set of classes taken and adapted from one of my projects (so some of it is a bit out of context). It consists of a data structure for organizing objects into hierarchical trees. Individual nodes can be in more than one tree simultaneously (have more than 1 parent), hence "Multi"-Node.

The interesting thing (I think) is the iterator class (defined in MultiNode.h, used in main.cpp). I found I often needed to scan sections of a tree and perform operations on nodes matching some criteria (ie. sort of a combination of STLs find_if and for_each algorithms). However, I've never liked the style of for_each because the operation is always separated from its use (the code that does the work is never in the same place as the loop control). eg:

```
for_each (start, end, functor_defined_somewhere_else)
```

I almost always wanted to do something simple enough and specific enough not to warrant a totally new function, but doing the iteration by hand every time is messy and error-prone, so I built a predicated iterator that enabled me to retain the structure of a traditional for-loop:

```
for (/* loop control */)
{
    /* operation */
}
```

Specifically, it allows things like:

```
for (tree_iterator i (root, SomePredicate); i; ++i)
{
    i->SomeOperation();
    i->AnotherOperation();
}
```

This way the iteration mechanics are factored out and kept together, and the body of the loop is devoted to the operation. I've found this to be a useful tool when dealing with my object hierarchies. It allows for very complicated iterations to be represented clearly with all the appropriate parts abstracted.

Henry Smith

henry@enigmasoftware.ca