



Joachim Bengtsson
Programmer

Born March 1985
Online <http://thirdcog.eu/>
Email joachimb@gmail.com
Phone +46 (0)70 260 49 20
Mail Folkparksvägen 12:06
 372 40 Ronneby
 Sweden

Education

2004 v 2008	<p>Blekinge Institute of Technology Game Programming</p> <p>Game programming at BIT is a software engineering education with emphasis on performance and 3D visualization. In addition to the programme courses, I have studied AI and mobile communication.</p> <p>I considered myself a programmer long before I began my studies at BIT. However, with all that I've learned at BIT, that notion almost seems silly. I've gone from toys in VisualBASIC to template hacking in C++ and excursions in a dozen different languages. Even as a kid, I was interested in networking and network programming. I've continued along that path, but also developed an interest in game engine cores, scripting integration, and lately also sound programming.</p> <p>My candidate thesis is called 'Real-time acoustics modeling in games'. I hope to get my Bachelor's degree by January 2008.</p>
2001 v 2004	<p>Söderportgymnasiet, Kristianstad International Baccalaureate Diploma Programme</p> <p>IB DP is an international secondary education/university preparatory programme in English. I chose to study economy, chemistry and physics, in addition to the compulsory languages and math. I excelled (7 pts) in physics and English.</p>

Work

2005 v	<p>AB Spelguiden</p> <p>I have done a number of smaller consultant jobs, mostly web sites and such. However, my biggest client has always been Spelguiden, a provider of horse betting odds and gaming/gambling advice. My first project for them was a relatively simple 'thin client' displaying PDFs off of a central server, but the project has grown over the years and now includes a relatively large network of touch screens throughout the country, the administrative tools for the all content, an online store, SMS tips and more. Spelguiden have always been very pleased with the system and its development, even though I'm the lone programmer and designer for it.</p>
1998 v	<p>Macdata AB</p> <p>Macdata AB is my brother Michael's Mac shop. I've worked at and for Macdata to and from over the years. I did my first work practice as a service technician there in '98, which was repeated as a summer job for a few years. Recently, I built their online store. I also help out with custom solutions for his customers at times.</p>

Skills

Programming Languages

Proficient	Average	Written a few apps	Have used
C++	Python	Java	bash scripting
C	Javascript	Lua	x86 assembler
Objective-C (Mac OS X)	SQL	C#.NET	Experimented with Lisp, Erlang, Haskell, Prolog, Io, Nu, ...
Ruby	AppleScript	(Visual/REAL) BASIC	
PHP			

Selection of Libraries and Technologies

Programming libraries: STL, PSPSDK (Homebrew PSP), OGRE, Cocoa (and related frameworks), OpenGL, FMOD, CoreAudio, J2ME, J2SE, ...

Web: Ruby On Rails, XHTML, CSS, jQuery, Mysql, ...

Computing: Macintosh System 3 to Mac OS X 10.5 (Mac user since I was a toddler), Unix systems (with all that this entails), Windows XP; XCode, Visual Studio 2005, NetBeans; Word, Excel, PowerPoint, Photoshop, Flash, ...

Summary

My never-ending curiosity has made me something of a Jack-of-all-trades. However, I would say my areas of expertise are:

- Embedding scripting languages into C-derivative applications; specifically [Lua](#), [Ruby](#) and [Python](#),
- [Touch screen applications](#),
- In [game development](#), specifically [core](#), [sound](#) and [network](#),
- [Mac development](#) and [Mac porting](#).

I'm also very much into User Interface design and usability in general.

Miscellaneous

For a complete list of my consultant jobs and other projects, see my portfolio online at <http://thirdcog.eu/portfolio/>. See also Appendix I for an excerpt of game-related projects in my portfolio.

I speak Swedish and English well, as well as some German (if my life depends on it). I have a driver's license.

I have been sound technician for three of Blekinge Kårpex' plays during college.

Appendix I: Portfolio Excerpts

LuaPlayer (<http://www.luaplayer.org>)

Role Co-programmer

Collaboration [Frank Buß](#): Lead developer

Description Collaboration with Frank Buß, bringing Lua together with a simple API for writing homebrew games on the Sony PlayStation Portable.



Frank started the project early '05. During summer '05, I spent my summer holiday adding sound support, rebuilding the API entirely to make it object oriented, and wrote a simple shell for launching games (called [Lowser](#)).

LuaPlayer became something of a phenomenon. I was invited to hold a session during EuroOSCON 2005 in Amsterdam. Since then LuaPlayer has unfortunately been pretty much dead in the water since both me and Frank have had our hands full. However, there is still a community of users writing games and utilities for it.

Robot (<http://www.robot-game.com>)

Role Core, Network and Game Logic lead developer; Mac platform maintainer; and Configuration Manager

Collaboration I was a part of [Team Heptico](#).

Description Robot was the first relatively realistically sized game project of the Game Programming programme at BTH. Our group was comprised of one project lead, three graphical artists, three game designers and five programmers.



During ten hectic weeks in 2006 (working 50% officially, 110% in practice), we spent literally day and night realizing how much more difficult it is to implement a game of any considerate complexity. Robot was built on top of the OGRE game engine.

I took on much too much responsibility, implementing a retain count-based class hierarchy of utility and base classes for the project, with tight Ruby scripting integration, serialization using RakNet, YAML, platform abstraction and an event manager; a network and game logic layer on top of that, relying on Ruby and RakNet; and configuration management (svn, trac, build env).

RMS (<http://www.friendlystapler.se>)

Role Project Manager; Sound sole developer; Core co-developer

Collaboration I was part of the [Friendly Stapler](#) team.

Description Rymdvarelser Mot Soldater was the last big project during Game Programming, spanning 30 weeks (at 50% officially, more like 100% in practice). This project was set apart from Robot in that we were eight programmers, no game designers, and one artist; and because we decided to implement almost all of the engine on our own, with almost no middleware except hardware interfaces.



[David Sveningsson](#) inherited my core and game logic code from Robot, which he heavily modified and I helped evolve. My main responsibility was the sound engine, which I integrated with my candidate thesis ("[Real-time Acoustics Modeling in Games](#)"). However, since I spent too much time learning the basics of audio programming and implementing a sound engine for just direct sound propagation, acoustic modeling was put on hold. I also managed the project, which included planning and scheduling, task delegation, and writing many, many reports.