

**Name:** Kriss Daniels  
**Date of Birth:** 24th November 1972  
**Telephone:** +44[0]2084074122  
**e-mail:** Kriss@XIXs.com  
**Address:** 82 Milton Road, Croydon, CR0 2BL UK

**General:**

I am a highly competent programmer, artist and self-learner. Much of what I know comes from thinking about problems, understanding them and then working out my own solutions. This is by far the most useful skill anyone can have on the bleeding edge of game development. Recently I've been employed as a programmer, but the activities involved are just the correct solution to the problem. No matter what department I've been employed in my full skill set has always been useful and a holistic viewpoint invaluable.

I am a strong team player which combined with my wide range of abilities means that I will naturally take responsibility for whatever area is currently lacking. I find that I quickly become the person that artists and programmers want to run their ideas past before they commit to a course of action.

I understand physics and math but more importantly I can animate. Instead of being bogged down with attention to pointless detail I'm simply interested in producing a good result. The ability to express good animation through math is one of my most valuable skills.

Games are my art and I'm interested in and well read in all areas associated with them. From the more obvious technical issues to the less well understood areas of design and more specifically game theory. One of the things I do believe is that although most programming is indeed purely technical and engine related there are some largely ignored areas that exist as important game content rather than generic game engine. Particularly this applies to physics areas that control the entire feel of a game. My skill set puts me in a perfect position to work well within these areas.

I am highly motivated to work in the games industry with a huge passion for games. I like to try and evaluate as many releases across all formats as is humanly possible. I own an old JAMMA cab, which I've modified to run modern consoles although it's mostly used to run old arcade classics. Sure it's just a huge old chipboard box but it feels good to have it glowing in the corner. As far as more modern online gaming goes I'm an active member of a counter strike clan and keep an eye on MMOGs in the hope of finding a good one.

This is what I have chosen to do with my life and I happen to think I'm quite good at it.

#### Hardware:

**Spectrum, C64, Amiga, PC, PS1, XBox, PS2 and GBA.**

#### Low level Programming:

I have strong low level skills ranging through the following languages **ARM/THUMB, Z40, RxC00, 680x0, 80x86, Z80 and 6502**. Most of my major asm work was on the **Amiga** where I reverse engineered a number of titles from raw binaries adding extra system functionality for their release on the **CD32** console.

#### High level Programming:

Inevitably I end up using **CPP** which I am highly experienced in. Due to my low level abilities I'm capable of monitoring compiler output, writing asm friendly code and replacing functions with asm where necessary. Although I am very interested in all language developments my low level background leads me more towards self imposed structure within **CPP** than a desire for an alternative language. That said I have experience with **C / CPP / LISP / JAVA / J2ME** and a number of lesser know dialects. In general I use what needs to be used and have an agnostic attitude to language wars.

#### Scripting:

I've always been fond of **REXX** since my Amiga days. Its easy prototyping functionality is something that has repeatedly proved invaluable. Today I'm keen on **Lua** which has taken over **REXXs** role for me. I've also used a number of application scripting languages such as **Vicon body language, BML, MAXScript and Action Script**. I have very strong programming skills that enable me to deal with whatever I need to use, regardless as to whether I've used it before.

#### Code Packages:

I've settled for **MSDEV** over **Code Wright** simply because it's easier than arguing and "its not that bad" I prefer **CVS / SVN** for version control since it mostly works and lean towards **CMAKE** as a metamake system. I like to design strong self contained build environments that convert art resources as well as compile the code. My ideal setup is that everyone should be able to checkout and rebuild the entire projects in a windows or unix environment. A well scripted build system and good art build tools are very important. I've also worked with **GCC** extensively on a number of platforms.

#### Art Packages:

**Photoshop, AfterFX and Promotion** are my preferred bitmap creation and modification packages. My **Photoshop** style is one of building masks via hand/render/draw&scan then applying FX through the masks to build up an image much like airbrush artwork. **AfterFX** is great for prototyping or building animations and can function as a perfectly good 2d level editor with a simple exporter. **Promotion** is just the closest thing to **D-Paint** or **Brilliance** when you really need to push pixels around.

#### 3D Packages:

This is a little more complicated, I have a soft spot for **Lightwave** since it was the first one I actually liked back on the Amiga. Although **Imagine** is the first one I used in anger I never liked it. Ultimately each package has its strong points and I'm inclined to veer towards the best one for the job, if the job is creating something in a certain package then that's the package to use. I have **plugin writing** and user experience with **Lightwave, 3DS, MAX and Maya**. Personally I tend towards character creation and animation since I find them more interesting than static environments, which is probably one of the reasons why I still favour **Lightwave** despite its obvious flaws.

#### Middleware:

I have experience with Intrinsic **Alchemy, RenderWare (3.6)** and **Havok**.

August 2003 - Now

<http://www.WetGenes.com>

Please do not consider this recent work as me simply keeping a hand in. Understand that in my experience the chance of any actual time to research on the job is below zero. This knowledge acquisition although not considered a viable use of time in the standard fire fighting development environment, is, never the less, completely vital to it.

April 2003 - August 2003

**HotGen Studios**

<http://www.hotgen.com>

**Medal of Honour Infiltrator**

<http://spong.com/x?vid=11035695>

Mostly low level hardware interface systems, from highly GBA optimised fixed point math class to video memory/display handlers as well as being in charge of the shooting sub game, build process and the link system. Generally lending my experience to everyone as I was the only programmer on the project that had previously worked on the GBA. I also wrote some custom export plugins for AfterFX allowing it to be used as a level editor.

May 2001 - July 2002

**HotGen Studios**

<http://www.hotgen.com>

**Egg Mania**

<http://spong.com/x?vid=11029707>

**Kelly Slater's Pro Surfer**

<http://spong.com/x?vid=11030113>

At Hotgen I worked on 3 GBA games. Produced interfaces to and gained experienced with the full range of GBA hardware from CPU base 3D rendering to save game and custom link code. Obviously it's not an overly complicated machine but I do have all of it firmly in my head. The game styles ranged between mode7 racing with CPU rendered cars, surfing with CPU rendered surfers / FMV playback and a highly colourful / active puzzle game.

October 2000 - April 2001

**Climax ( Brighton )**

<http://www.climax.co.uk>

**Superbikes 2002**

At Climax I started converting the PS2 version of Superbikes2002 to PC and XBOX. It was in a working state with the only really PC specific part, the network code, implemented and playable. This was then cancelled by the publisher due to lack of confidence in PC sales. Then I worked on the PS2 version, adding some eye candy in the form of a nice sun/lens flare effect with directional rays dependent on occlusion and a particle system. The particle system was predetermined rather than feedback. This means only emitter state information needed to be stored with all particle information calculated at draw time then discarded. This allows for more particles with less memory, real time unlimited modification of the number of particles drawn depending on visibility and the ability to off load the entire particle task onto the PS2s VU.

July 1999 - September 2000

**Etchinghill Studios**

[http://web.archive.org/web/\\*/http://www.etchinghill.co.uk/](http://web.archive.org/web/*/http://www.etchinghill.co.uk/)

**Apostle**

At Etchinghill I was originally employed to help produce an actual game out of the company's engine. However what I was actually asked to do was mostly engine rewriting and fixing. Rebuilt the collision system into one that didn't fall through floors, based mostly around spherical to polygon collisions. The results were perfectly smooth repositioning under all circumstances including special cases for walking up steps, etc. Rebuilt the sound system, which unlike the previous one, supported CD control, streaming audio, 3D positional sound sources and the ability to play the same sample more than once. Built a generic atmospheric sample player designed to give situation based random atmospheric sound as background noise. Built a particle FX editor as a generic MFC editor shell able to cope with any number of tweakable parameters with real time view. Based around a generic data definition structure table which allowed the same editing to be done both graphically and in command line form. Rebuilt the input system into one that supported full redefinition of input signals along all available DX input types. Again using the generic data definition structures as used in the PFX editor. Built the CCC AI system as a large state machine dealing with various situations from map navigation to fighting. The source is still in a clean commented structured state despite various additions since the original design. Built a nice font system for the front end, using alpha channel bitmaps for smooth edges and full kerning as well as layout and control sequences allowing colour/size to be changed midway through strings. The use of alpha bitmaps means the fonts can be fractionally scaled with little loss of definition allowing the same bitmap to be used for the same sized text no matter what the screen resolution. Scripts to clean up a smooth the data from our in house motion capture studio. Built a simple gadget system allowing scrollable text boxes and user input for simple parameter editing. Again this uses the same data definition system as the PFX editor but is an in game system rather than MFC.

September 1997 - May 1999

**Abstract**

[http://web.archive.org/web/\\*/http://abstract.co.uk](http://web.archive.org/web/*/http://abstract.co.uk)

**Joe Blow**

[http://web.archive.org/web/\\*/http://www.telstar.co.uk/tes/html/joe/home.htm](http://web.archive.org/web/*/http://www.telstar.co.uk/tes/html/joe/home.htm)

For Abstract I did whatever needed doing to make Joe Blow, a free roaming 3D platform game. Designing and implementing all the necessary game and engine elements. This list includes, WYSIWYG object animation system for playing Lightwave anims at any frame rate. Collision system requiring little in the way of special polygons to keep it happy. Supporting static, moving and/or rotating map sections. Graphics device interface supporting Glide and D3D. Tweaking and playing with the user control/physics/collision until it had a nice feel to it. Simple game scripting and trigger system allowing various ways of triggering events and linking them to game progression. Fully configurable direct input interface supporting all input flags/pots available at the time of writing. Positional 3D sound system As well as the following achievements. "Bodging" together an ECTS demo within the first 10 days from the total mess the previous developers left behind. Producing a playable glide based PC game and the tools to edit maps and produce in game animation, from scratch. All functional within the first few months to keep the publishers happy. Conversions of the original game code to PS1, which was of course designed to ease the conversion. Building plugin code for Lightwave to simplify the creation of anims for use in game.

July 1995 - July 1997

**Cranberry Source**

[http://web.archive.org/web/\\*/http://www.cranberry.co.uk](http://web.archive.org/web/*/http://www.cranberry.co.uk)

**Super Match Soccer**

**QAD**

For Cranberry Source I produced graphics and in house utilities working primarily as a liaison between the artists and programmers I completed the following tasks. Designing plugin code for Lightwave and 3DS to render mesh objects as voxels for use as Sprouts in the games QAD and Super Match Soccer. Reverse engineering the Acclaim motion capture file format due to no documentation existing at the time (Actually had people from Acclaim asking questions so they could document it). Designing code to batch adjust motion capture as well as tweaking hundreds of files by hand for the game Super Match Soccer. Designing low polygon models and textures for Super Match Soccer as well as writing code to export the models from Lightwave in a form the games programmers could use. Designing the company's web site, this has since been replaced with a SMS only website after the company shut down.

March 1994 - March 1995

**Almathera**

[http://web.archive.org/web/\\*/http://www.almathera.co.uk](http://web.archive.org/web/*/http://www.almathera.co.uk)

**Video Creator**

**Emerald Mines**

**Lost Vikings**

**Battle Chess**

I worked on three main projects at Almathera. Reverse engineering a number of early Amiga titles to work on the CD32 such as Battle Chess, Lost Vikings and Emerald Mines. I acquired a good knowledge of CDROM technology back when it was an expensive hobby and was closely involved in the mastering process. I also made a substantial contribution to the Photogenics paint package. Designing much of the code and supervising the work of three other programmers. Photogenics was bundled with some of the last a1200s taking on the role of DPaint in earlier years. For the Almathera's Video Creator package, which allowed a slide show to be synchronized to a music CD, I designed and coded a 2D vector animation editor. This permitted a wide range of graphical effects to be included in the slide show.

September 1992 - October 1993

**Air-Time Interactive Marketing**

Air-Time is a company producing leading edge multimedia software for NCR fourth generation ATMs. These machines are capable of state-of-the-art audio visuals with which the user interacts via a touch screen. I produced a variety of graphics for these machines, ranging from the heavily rendered attract mode animated sequence, which stretches the machines capability to the limit, to simple static screens prompting for amount entry. The static screens are built up from a number of discrete components such as title bar, border, button tops. These are essentially the same on all screens to give a consistent "look and feel". In addition to designing these elements, I designed and implemented a scripting language to paste these screens up from their component parts, such that a change could be made to, say, a button, and then all the screens rebuilt with the new button design from scripts without further intervention.