JASPER BYRNE

-PORTFOLIO-

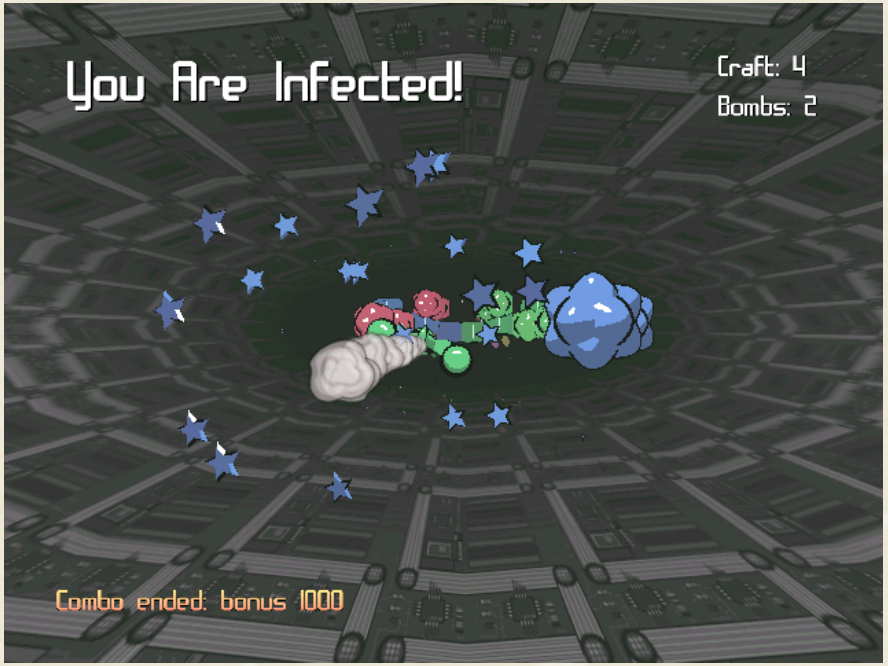
Section 1: Current demos

THE GERMINATOR

The first game I'd like to present has been aimed specifically at my job-search. I include links to the Windows binary, as well as the full source code.

The game is a 3D shoot-em-up in the vein of Sega's Space Harrier. I called it The Germinator. It was coded from scratch in under a month, and entered into independent games website TIGsource's (www.tigsource.com) Procedurally Generated competition in June 2008.

Forum thread, including instructions and downloadable game:
TIGsource thread for The Germinator
Source code:
Download



The Germinator (2008)

The Germinator is written entirely in C++, using the open-source 3D engine, Ogre. Sound is handled by FMOD. It was written in a combination of XCode and Visual Studio.

The Germinator features my first use of shaders, using GLSL / HLSL to provide the cel-shader (using the classic 1D lookup for diffuse / specular colours, and the dot product method of finding outlines in the pixel shader). In my research since making the game, I have learnt of the Sobel Filter technique for outlining, and have successfully implemented it in a 'compositor' pass (Ogre's terminology for a pixel-shader applied to a full-screen quad). I would have liked to plug this new shader into the game, if I'd had the time.

Although the game is not one hundred percent complete (I planned to include a boss, intro and hiscore screens), it is certainly playable and, as far as I know, bug-free. Hopefully it illustrates my understanding of design patterns in the underlying framework. I have tried, where possible to use as few global variables and as little inter-class dependency as possible.

Key skills learned:

HLSL / GLSL shader languages & theory

The Ogre 3D engine

Visual Studio 2008 / XCode use

FMOD sound library

Designing a simple 3D framework

Using procedural generation for level design

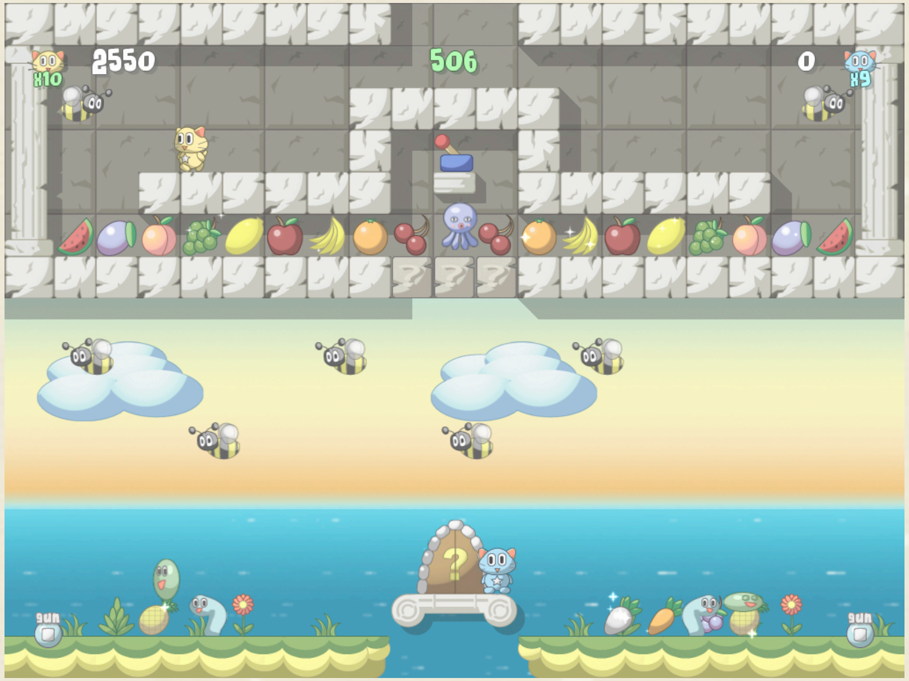
SUPER NEKO WORLD

The second playable demo I'd like to present is Super Neko World. This is a 2D single-screen platformer that resembles Taito's Bubble Bobble, but with different kinds of weapons. This I feel is my most polished work - although still missing some content the game itself is finished; it took me several months of solid work.

It features more than 200 kinds of bonus, many of which give you strange and interesting powers. It's great fun head-to-head.

Please use the following link to download from my server space:

Super Neko World for Windows / OSX



Super Neko World (2007)

Here are the controls which are re-definable (you can also use a joypad):

Player 1

Shoot - ';'
Jump - '.'
Left - left arrow
Right - right arrow
Up (when flying) - up arrow
Down (when flying) - down arrow

Player 2

Shoot - left control key
Jump - left shift key
Left - 'a'
Right - 'd'
Up (when flying) - 'w'
Down (when flying) - 's'

The idea is to shoot the baddies with your ice gun (or flame gun, lightning gun or rock gun) and then jump on their heads before they have time to revive.

Powerups can do many things, from making you fly, to hypnotising the baddies with sweet music. Wiping out all the baddies finishes the level.

You'll be awarded a chest for finishing a level while the timer remains green. Sometimes this can lead to bonus rooms. The size of the chest depends on your maximum combo (consecutive head-jumps without touching the ground.) In turn, collecting 'groups' of bonuses will put a special item in the chest, some of which give you more powerful powerups.

The system is quite deep, and I'm quite proud of it. It would take many, many replays to see all the pickups, and hopefully it's fun enough to do so!

Key skills learned:

Fast, easy-to-use map editor

Polish & playability

Balancing player movement 'feel'

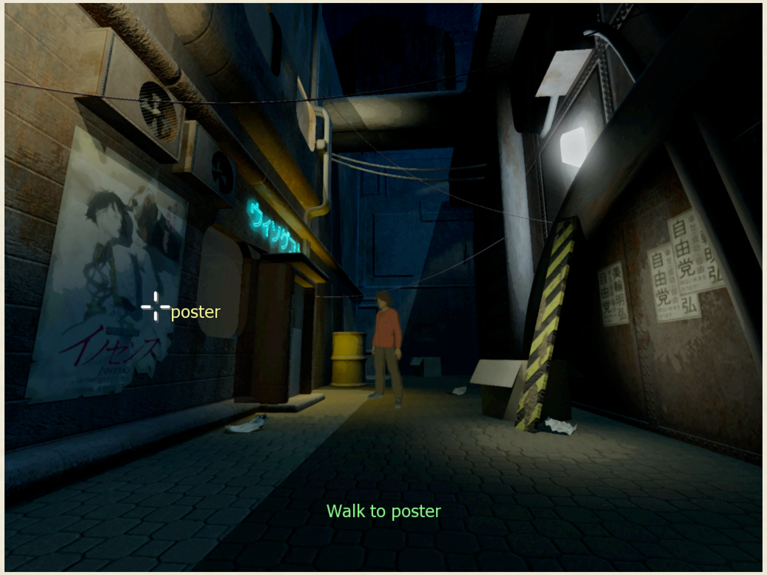
Writing a tidy framework that meant
little work in prototyping

Section 2: Ongoing projects

AMNESIA

Amnesia is a point'n'click adventure game and editor, capable of producing games in the vein of Lucasarts' The Secret of Monkey Island. I learned many key skills during this project, from pathfinding to writing a GUI. In many ways is the spiritual sequel to my published Amiga game, 'Keith's Quest', as it takes what I learnt from that project and my university 3rd-year project which specialised in game scripting languages.

Youtube video of Amnesia in action:
Watch



Amnesia (2006)

Key skills learned:

Homemade scripting language (HAL)

A* Pathfinding

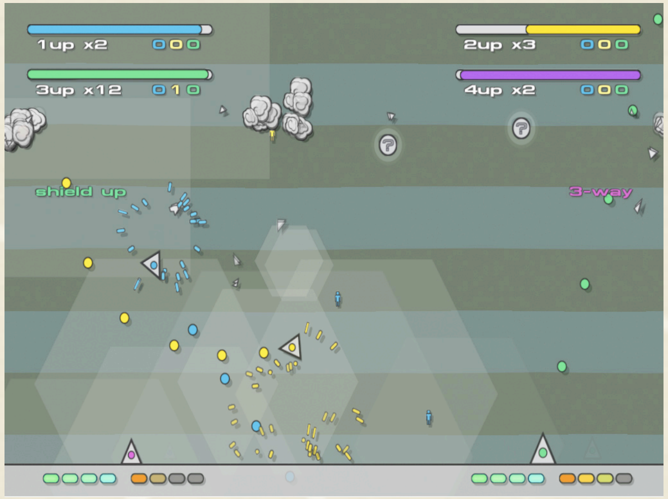
Writing a GUI

Automatic code highlighting

TRIP

Trip is a head-to-head gravity-based shoot-'em-up. It's been in development since 1992, and has appeared on many formats, starting life on the Amiga. It has developed a lot over the years, but hopefully kept the core addictive mechanic that makes me keep working on it! The current version is aimed at a platform like Wiiware or XBLA, featuring nice, abstract 1080p graphics. I wanted to make a game that emulated the competitive nature of Street Fighter II, and the energy bar, super move & rounds system is based on this Capcom classic. It features fully-fledged AI, a neat particle system and nice sound effects.

Youtube video of Trip in action:
Watch



Trip (2008)

Key skills learned:

AI within a physics-based world

Particle systems

Gameplay balance & 'fun factor'

Coding in OpenGL directly

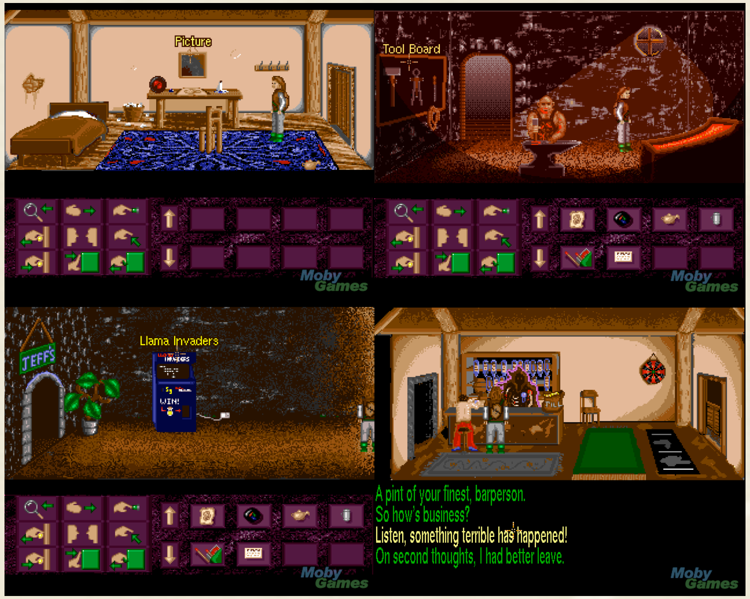
Section 3: Published work

KEITH'S QUEST

Here is the first of two of the games I produced for the Commodore Amiga. This is another point'n'click adventure game in the tradition of Lucasarts' adventures.

The game featured over 70 locations, 30 or so characters and thousands of lines of dialogue. It spanned a whopping 4 Amiga disks and took my artist and I two years to create. It was released commercially in 1995, receiving 85% in Amiga Format and 87% in CU Amiga.

This was the first game I'd produced to a professional standard. I wrote my own GUI and editor, including an animation scripting language. It is certainly one of my proudest achievements.



A pint of your finest, barperson.
So how's business?
Listen, something terrible has happened!
On second thoughts, I had better leave.

Keith's Quest (1995)

Although when I wrote Keith's Quest I was entirely self-taught as a programmer, I learnt a great deal. Perhaps the most important thing of all was learning to work through crunch-time, and not giving up when the last 2% of the work ends up taking 50% of the time!

I also learnt a great deal about taking the time to produce the right tools for the job, as it always saves time in the end.

It was also during this project that I decided to go to Manchester University to study Computer Science to further my knowledge. As mentioned briefly before, I specialised in game-scripting languages for my final year project (a reasonably new concept at the time.) This knowledge, combined with what I learned on Keith's Quest, lead me to produce Amnesia (see Section 2).

Keith's Quest on Mobygames:

Link

Keith's Quest on Hall of Light:

Link

Key skills learned:

Building a GUI

Producing to a commercial level

Script-writing

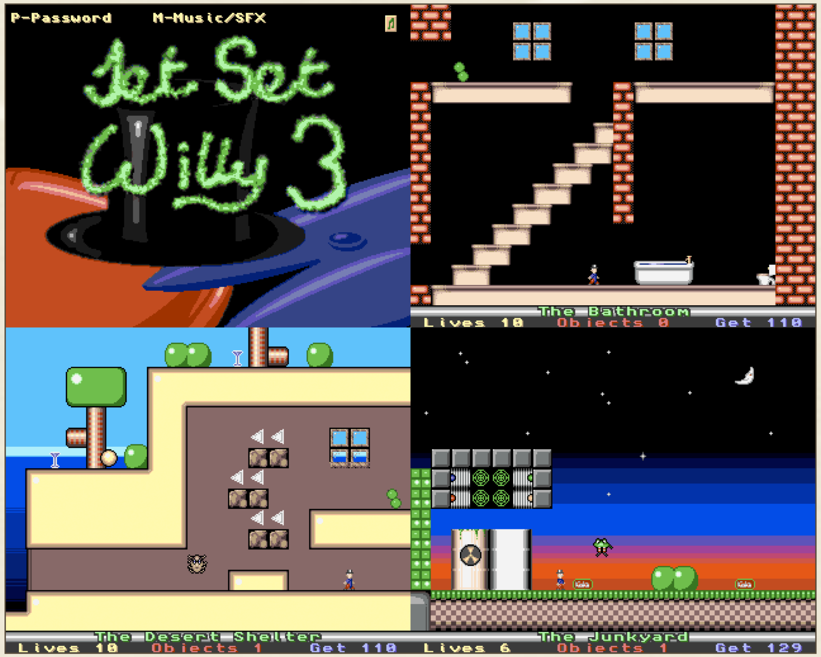
Puzzle design

Working collaboratively

JET SET WILLY 3

Jet Set Willy 3 was a freeware platform adventure, based on the Sinclair Spectrum games of the same name. I produced it alone. It had 80 screens and featured as the coverdisk of the now-defunct Amiga Format.

This was the first game I'd made to a finished standard, although it only took three weeks, having written most of the tools beforehand.



Jet Set Willy 3 (1994)

Key skills learned:

Tiled maps / editor building

Dirty rectangles (no longer relevant)

Collision detection

Actually completing something!

-