SIMON MAURICE - IPHONE - OPENGL ES

WELCOME IPHONE OPENGL FAQ ABOUT ME DRACONIA

Wednesday, 12 August 2009

OpenGL ES 19 - So You Wanna Be an iPhone Games Programmer?

Sorry for the corny title but back in 1990/91, Amiga Format magazine had a series where Dave Jones, shared the code of how he wrote the first level of the game Menace. Whilst I really didn't learn much about the coding side of the articles, I did learn a lot when it came down to *how* he thought as a programmer and *what* decisions he made when it came to actually implementing a game.

This tutorial sub-series is going to be a serious nod to Dave Jones and to that great magazine for what they did back in the days. I'm going to show you how to code a "Menace-like" game for the iPhone. Hence the title of this post, I'm pretty sure Amiga format used to splash "So you wanna be a games programmer" on their covers for one of their series with a pro game developer (they had several from memory). Like I said, this is a nod to them.

BRITAIN'S LEADING MAGAZINE FOR AMIGA OWNERS
AMIGA
FORMAT
ISSUE 7 | FEB 1990 | £2.95
GRAPHICS
Tricks of the trade from the master artist
PROGRAMMING
Menace and Blood Money author tells all
MUSIC
You too can become a keyboard maestro

From my last post, I said that I would do a A-to-Z of programming a 2D shoot 'em up. Well, that's exactly what this set of tutorials is going to be about. At the end of it, we should have a working game which can be submitted to the App Store for release. Yes, that even means I'll show you how to create levels, game graphics, and deal with iPhone issues such as what to do when the phone rings.

SECRETS
REVEALED
The experts help you to get the most from your Amiga
BEGINNERS
Indispensable guide for the new Amiga owner
GAMES
Definitive reviews of all the big games

Since this is going to be a "proper" project series, there really is going to be a few things you're going to need. They are:

1. An iPhone or iPod touch: Since most of the control will be via the accelerometer, you can't avoid having a device. I won't do "click on screen" controls. Well, maybe but I'm not promising anything.
2. A developer certificate from Apple: Yes, you've got to spend the US$99 to get the finished code onto a device for testing. This goes back to the first requirement.
3. Needless to say, you'll also need a Mac with Apple's Xcode tools with **iPhone SDK 3.0** installed.
4. Apple's X11 installed: we'll be using the GIMP for some graphics work so you'll need X11. Sorry, I don't do Photoshop. So yes, you'll need the GIMP, Blender, and a couple of other pieces of free software. Nothing that will cost you money to get.
5. Music: this is an unknown at the moment. I think all you'll need is GarageBand. To be honest, I haven't worked this out yet. Maybe I'll use a MOD player library as a nod to the Amiga.

You won't need everything at first. You'll only need one and two once we get to the accelerometer support. I know I've avoided that before because I know not all of you have a device to test on. To do this properly, you're going to need one. Just go spend the money when the time comes.

I'll give you some warning one or two tutorials in advance on those cost-item requirements.

BTW: Don't ask me how to get the code onto a jailbroken iPhone. I don't go there. Just pay the $99 and code things properly. It's part of being a professional (even if you are working out of your bedroom).

This code will be released under the GNU GPL v3. You know that means if you release something based on this code you need to release the source code as well. If you don't want to, then just contact me and we can probably make a deal but I don't think it should come to that. I really just think it's best if you release the source code as per the GNU GPL. I used to think that I had to keep my source code secret. Back in the days, I used to release Amiga demos as binary only but these days I think different. There really is very little in a game engine that another professional coder can't figure out by looking at your game so I don't think there's much point of treating most code as a national secret.

I know I said I'd wait until I got my own game done but, in reality, I've not got the time at the moment to do both a tutorial series and work on my own things. I do have a life. So I think for now, I'm going to do this as an open project and write it all in public so I can kill two birds with one stone. Either that or I won't bother as I am writing a desktop Mac app at the moment which for me is a bit more interesting because it's not a graphics project. It's actually something for my main hobby: beer.

But that's another story.

The Basic Game

The game that's going to be created is a side scrolling shoot em up. Like I said, it's going to based on the Amiga game Menace as a nod to Dave Jones' Amiga format tutorial series which is the first that I can remember seeing and taking it seriously. I haven't got all the details worked out yet but I've pilfered enough code from the other 2D game engine I'm writing to get started and we'll be able to finish this in a couple of weeks.

For those of you who don't know the game, there's a youtube video showing someone play the entire game through (AmigaLongPlay.com or something like that). If you search for "Menace Amiga Longplay" you should get it. That will show you the kind of game that we're going to develop.

Essentially, we're going to use OpenGL ES 1.1 in an orthographic projection. All this means is that depth is ignored so we'll just render everything using the painter's algorithm where objects are drawn from the rear to the front. We'll use parallax scrolling to provide the illusion of depth.

Of course, there will be sprites, lasers, explosions, collision detection, point scoring, baddies; all the stuff you'd expect in a shoot em up game. Now how can I forget to mention end of level bosses as well!!

The Tutorial Series

I knew at the outset when I came up with the idea of doing a game and releasing the source code, I didn't really want to just chuck the project file out there. I'd rather go through it and discuss the code. This means more work for me but I think it will be a better learning experience for those who are interested. Also, it means that if I don't get around to releasing the game, the code is not lost (not like the other projects I've done over the years).

Essentially, there will be weekly posts. I'm going to post a new tutorial every Monday night, Sydney time so I'm keeping everything regular. That way I can work on this over the weekend, getting the posts ready for Monday night. I'm not sure how many tutorials will make this series up but I think it would be at least a dozen or so. Probably more.

I will do some screen casts. For the graphics development only but I'm not sure how I'm going to make that happen at the moment. I don't think discussing code works in a screen cast but showing how to make graphics in the GIMP does translate well.

Just on the graphics side, I'm not really that crash hot on graphics. However, I do know how to do it after looking over the shoulders of real graphics artists over the years. I just do the same things as they do, just not as well so I think the techniques are correct, but let down by my lack of artistic talent. Normally I do enough just for quick and dirty prototyping, or for non-critical elements. Whilst my creations may not turn out that good, with persistence or some artistic talent on your part, you should be able to get something quite a bit better than what I can create.

After some thought, I did decide to implement it in Objective-C, even though I do prefer to code in plain old C. As much as I prefer not to use OOP, this is an iPhone series and you need to know Objective-C at some level to work on the iPhone and, besides that, many people like to use OOP by choice. I'd rather present something in Objective-C rather than the evil that is C++ which is a language I straight out refuse to use.

So, for this series, Objective-C it is. For the next series, I think I might just do it mostly in C.

Having said that, I'm not going to make everything into a class. You do see that all the time so there's going to be no creating of a tile class and then sub-classing half the planet etc. I'll have to use some of course but I'm not going to go crazy about it. I don't really care what anyone says; Objective-C class creation and messaging is more expensive than a standard C function call so I want to keep all that to a minimum. We're shooting for 60fps here after all.

The Very Short Brief for the Game
I'm not going to write a design document. This will probably evolve over time but let's set the ground rules for what is absolutely promised to be delivered for this title:

1. A full game, with graphics and audio, all developed here in public. Code released under the GNU GPL.
2. 2D side scrolling game with parallax scrolling. Running at 60 frames per second on the iPhone 3G. The iPod Touch and iPhone 3GS should be fine if the iPhone 3G can hit 60fps.
3. Fully handle the possibilities on the iPhone which means phone calls and other events which may interrupt game play like battery warnings.
4. Correctness. We'll write proper code which cleans up after itself etc. It must be suitable for publishing in the App Store so that means only official API code only.
5. We'll produce a fun game. I hope.
6. Multiple levels loaded on the fly. Right now, I'm shooting for around 5 or 6 but we can do one fully then only the differences for the other levels I guess. We can work it out later. At least be able to handle multiple levels.
7. Saving games, whilst not normal for arcade games, this is recognition of the iPhone user's needs. Also pausing and other basic game functions.
8. A high score table. Gotta have that right?
9. No off the shelf game engines. I'll tell you why below.

So that's the brief as it stands right now. The order in which things are implemented will be exposed over time.

Why do I say no game engines? Look there's multiple reasons why not to and of course "why to" reasons as well. Personally, I don't believe in them. I'm not going to bag them out here or their authors. In fact, I praise the work of the guys doing open source engines for their generosity towards the programming community and giving of their time to do so.

I'm going this route because it's what I want to do. I don't just want to bolt graphics onto someone else's code and not truly understand what's happening in the background. I don't just want to make a clone of a clone of a clone (although you're of course not just limited to that with an engine).

I want to program and design. That's what I want to do.

Like I've said before, this is only one man's opinion. You're free to make up your own mind. I've chosen this route because it's *what I want to do*. If you chose the path I'm presenting here, then you're right. If you'd rather go the engine route, then you're also right.

The only wrong route is to start one path and then chop and change every time you strike a problem and think the other route might actually be easier.

Suck it up big boy. Make a decision and *stick* to it.

Killing is My Business...
...and business is good. Yeah! The old Megadeth song!

I've gotta give the project a name and, without thinking too hard about the title, I'm just going to call the project *Draconia*. Anyone who played Menace will know why.

So, now, let us begin...

previous next