OpenGL is on trend!

October 3, 2009

Christophe Riccio
www.g-truc.net
g.truc.creation[at]gmail.com

A year ago

12 months ago, after the release of OpenGL 3, Tom's Hardware released an article titled "OpenGL 3 & DirectX 11: The War Is Over". At that time, I more or less agreed with it content even if I thought that this title is way too extreme but at least I didn't though that OpenGL was anymore a competitor of Direct3D. I was certainly very disappointed so that I stop for 6 months almost all personal contribution for the OpenGL ecosystem and looked at others things. I think that most of the world doesn't rely on the human minds but on human hearts. Therefore, I can resume by position as "I had lost faith in OpenGL". What worse for someone whose job involves OpenGL, what worse for someone whose first of many passions is OpenGL? 14 months latter I notice something great: My heart is biting again, I believe in OpenGL again! With this article, I would like to share all the good reasons I found and there is a lot of them!

Game engines use OpenGL!

The widest wrong idea spread about OpenGL is that games don't use it anymore… This is just not true. What true is that on PC, the primary platform is Direct3D with few exception for old games like Quake Wars and World of Warcraft. A great buzz was "John Carmack gives up on OpenGL". That's not true has well. Rage will run on PC using Direct3D but the Id Tech 5 supports both Direct3D and OpenGL. This is the case of a lot of game engines, like the most used in game development Unreal Engine 3 and the one I think is the most impressive today: CryEngine 3. Why OpenGL is supported? On PC and Xbox360, yes, games use Direct3D but what about Mac, iPhone, PS3 and probably PSP2? They can't use Direct3D. It could be a proprietary APIs like Nintendo still does (I don't believe for long in the future) but it's simply better for them to use OpenGL, an known and not too badly designed API. Direct3D could not be use just because Microsoft would have to support all those platforms. A lot of game engines use OpenGL: There are markets to be OpenGL developers and I am going to detail them.

The Apple Effect

iPhone

Maybe the most important information for the last Apple Keynote was that Apple sold the iPod Touch as a gaming device: There is a market for games on iPhone and iPod Touch! Imagination Technologies PowerVR MBX chip (OpenGL ES 1.1) in iPhone 3G and PowerVR SGX chip (OpenGL ES 2.0) in iPhone 3GS are more that descent GPUs. It makes the iPhone a gaming platform!

On the development side, the PowerVR SDK is fairly great with a simulator, samples, tools and even a surprisingly detailed known issue list. Apparently, the drivers are not 100% compliant with OpenGL ES specifications but who cares if we accurately know what's wrong? I enjoy this transparency to developers. iPhone sells are still growing and actually the overall market of "smartphones" and embedded devices as well. There is a market here to be an OpenGL developer!

Mac

The idea of gaming on Mac would seem naïve for a lot of people; most gamers I guest. However, I think this is the remains of an old idea! I think there are few games on Mac for two mains reasons.

First, there are fewer Macs than PCs. Well, there are more and more Macs! They use to be really

more expensive than what you could get on a PC but it's less and less the true. The second issue is (or was) that Mac had graphics chips that suck!

For way too long, Apple didn't care enough about graphics and used Intel chips that I believe should not ever be called a graphics chip (my policy is Intel chips: I don't care because efficiency wise, feature wise and drivers wise: they really suck!). Fortunately, and even if Apple still doesn't care about graphics, Apple had a long term strategy for their products: OpenCL! (and possibly the GPU) to accelerate any kind of software using less power. This looking forward strategy builds a partnership with nVidia so that all Macs, sell since at least a year, have a fairly good graphics chip inside all laptops… Each single Mac sell today is a potential gaming platform (not for hardcore gamer but still!) and Mac market share is growing: There is a market here to be an OpenGL developer!

Throughput computing

OpenCL

OpenCL is supposed to be an initiative of Apple but a lot of developers would say it's a "copy paste" of nVidia CUDA. I said it as well. Today, I think this is true and false.

It is true because when you look at the API and the features, it is really similar. It is false because Apple understand first that throughput processing is the future for casual consumers. OpenCL was scheduled for Snow Leopard, for every consumer, CUDA never leaved the scientific or professional areas. To reach consumers or gamers, a standard was required and OpenCL was the answer.

OpenCL targets so many platforms! It could be use with GPUs but also with CPUs (x86, Cell, ARM's). OpenCL makes senses for so many areas beyond a use of GPUs for throughput processing. It gives some answers for multiple core platforms (I will write a separate article on that latter on).

I believe that OpenCL on PS3 would be such a great idea to unleash the Cell processor power or at least makes it easier. In few years (months?) time we will see OpenCL on mobile phones. It's exciting for developers and when I listen at PC Perspective podcasts I realize that it is exciting for some power consumers as well.

The 2 Open*L brothers

What the link between OpenCL and OpenGL? OpenCL is for OpenGL what DirectCompute is for Direct3D and I would like to say: It's even more! OpenCL specification contains a whole part for interactions with OpenGL API.

Again the PS3 is such a great example: I think that most game programmers will agree with this statement: "The nVidia RSX chip is weak, weaker than ATI Xbox 360 chip". However, the PS3 Cell is so powerful but too complicated to program (or so unusual?), it would be so great to use some of its power to compensate the RSX lack of power. OpenCL has many answers on this topic especially thanks to the interaction API between OpenCL and OpenGL. Basically, this API allows sharing the same memory between OpenGL and OpenCL objects: To share buffers and images (Where CUDA is limited by sharing buffers).

Some good examples of use could be vertex skinning that could be done on the Cell and the

resulting vertex data would be processed by the RSX without copies. Other example, all the framebuffer post processing effects could be done with OpenCL on the Cell instead of an extract pass in a RSX fragment shader. OpenCL is on trend, OpenCL interacts with OpenGL which give more reasons to use OpenGL!

WebGL

OpenCL is really on trend but there is a project which will probably be use even more in a year time: WebGL. This project has been announced at Siggraph 2009 during the OpenGL BOF. It's an OpenGL based API for website! We are browsing the web from so many platforms beyond computers: Mobile phones, consoles, TVs, etc. A cross platform standard is required.

OpenGL is such standard! It has been announced 2 months ago, no specification released yet but already some developments and showcase have been done, Webkit used for Safari (Apple again...) was the first to include some code for WebGL on the source repository, Firefox the second and Google Chrome has already presented similar technologies and state on their interested for WebGL. If you dare to call it a web browser, Internet Explorer, just got hacked by Google to hide Chrome behind it! The project is called Google Chrome Frame.

How many websites, how many web users? Seriously, I would love to use some WebGL stuff for G-Truc Creation 7! There is a potential of 6 billion users for WebGL, this is not a train that a lot of companies would like to miss!

OpenGL: a high end library!

OpenGL: an up to date API (usually)

An idea is that OpenGL is so out to date, that graphics features reach OpenGL after Direct3D. This is wrong!

Actually, it happens just once with Direct3D 10.1 and it's partially happened a second time at the launch of ATI Radeon 58XX, 23 September 2009, the first Direct3D 11 card. The main thing with Direct3D 11 is definitely DirectCompute but OpenGL has its brother, OpenCL, running on MacOS X since Snow Leopard release, 28 August 2009.

Other main feature is tessellation but AMD released an OpenGL extension (GL_AMD_vertex_shader_tessellator) on the 6 March 2009. To be fair, Direct3D 11 tesselation is more flexible than what this extension seem to do (I haven't tried it yet).

Finally, the main lack of feature is the deferred context, which I believe is a huge feature. When it happens with Direct3D 10.1 at ATI Radeon 38XX released, on 15 November 2007, it was clear: none of the Direct3D 10.1 features were available on OpenGL.

nVidia support of new features

If we move backward again in the history we reach Direct3D 10 which was release on Windows Vista and exposed the features of the GeForce 8, the first Direct3D 10 chip the 8 November 2006! This GPU has been released before Windows Vista (30 January 2007) but at the GeForce 8 release, nVidia exposed all the features with OpenGL extensions available on Windows XP. So when it's

nVidia who is “first” we get the quick OpenGL updated at release but if it is ATI, it's not the case? Is there an issue with OpenGL on ATI cards?

ATI gets better

Let's be honest: Yes there was an issue for a long time, we still see the results of it especially with the ATI support of OpenGL on Linux and MacOS but it became really better on Windows! Months after months developers see the evolution of ATI drivers and it evolves fast! I don't know what change in the ATI policy about OpenGL (AMD bought ATI?) but it's a matter of fact. The drivers become more reliable and ATI has been really innovative the past year. A good example was the (GL_AMD_vertex_shader_tessellator) extension but some OpenGL community members dig into the drivers and notice a lot of functions entry points for OpenGL support of Direct3D 11 features.

nVidia is still reactive with OpenGL drivers

Other digging results were a lot of OpenGL entry points for Direct3D 11 level features in nVidia drivers. When nVidia will release the GT300 in December or January, I bet that they will release once again the OpenGL extensions to use GT300 features!

OpenGL: A standardize API

An extension need to become an ARB standard for adoption

To make an extension useful for a product released, it has to become a standard, which means that the ARB has to validate the features and the way they are exposed. It takes so long sometime in the past... Still, it’s faster that most standardisation, at least so much faster than C++ 0x!

The ARB is working hard!

In 12 months, the ARB released 3 OpenGL specifications: OpenGL 3.0, OpenGL 3.1 and OpenGL 3.2. That’s what the ARB claimed at OpenGL 3.2 release but in fact, it took 3 years to get a very controversial OpenGL 3.0 done, 6 months for OpenGL 3.1 (fastest release ever!) and 6 months again for OpenGL 3.2 (unbelievable!). Moreover, both of those last versions were great releases providing good features, good API evolutions. Feature wise, OpenGL 3.2 core specification (without any extensions and without deprecated features) reach Direct3D 10 level and if we add standardize extensions, it reaches Direct3D 10.1 level. Well, there is still few gaps.

The ARB involved

The past two years we saw more and more actors from the gaming industry participating in the OpenGL specification process. Historically, this is new! It is known that John Carmack contributes in the past so specify major features. He more or less influenced nVidia to add some features like two side stencil operations so that it Doom 3 stencil shadow volume would require just a single pass of shadow pass per light instead of two.

Well, that's the only name from the gaming industry I noticed before expect maybe Brian Paul from Tungsten Graphics. Since the past few years, new names appears from various software company: TransGaming, Aspyr, Destineer, CodeWeavers, Holographifika... I mainly notice names from company that are used to port games on MacOS X (Remember, there is a market!) but also

some big company names: Electronic Arts, Pixar, Epic Games and the most active of those Blizzard.

Warcraft 3 and Diablo 3 available on Windows and MacOS X at release? That would make sense to me. Available on Windows through OpenGL? Well, I wish but … not yet!

A happy OpenGL community

Open roots

For years OpenGL had this idea to be “open”. We can say that the release of OpenGL 1.0 in 1992 was the first demonstration of Silicon Graphics to be “open”. It was an open standard for the graphics industry, it didn't kill Silicon Graphics, what kill Silicon Graphics is the lack of innovations. For years, Jon Leech for Silicon Graphics was reporting to the community the content of the face to face meeting of the ARB every three months … and it stop.

The silence decadence

I think that during the development of OpenGL 3 (It might actually have started during the development of OpenGL 2), the ARB closed itself from the community and fall into silence. Usually, it isn't a good sign, isn't it? It makes me feel that a lot of divergences were present in the ARB for the future of OpenGL at that time and they didn't managed to solve them or even making progress to solve them.

OpenGL Long Peak fake and OpenGL 3 failure

I was present at the OpenGL BOF of Siggraph 2007 where OpenGL 3 “Long Peak” had been presented in quite some details and which was supposed to be released a month after. During the talks, I felt that some of the guys on stage, from the ARB, were discovering things … I was existed by Long Peak because Long Peak was supposed to be better than what we get with OpenGL 3.2 core (with less features to be honest) but something was wrong. I remember telling my friend: “This is so not going to be release in a month!” So many gaps... Still, I believed that the ARB would get it right at some point, probably like a lot community members. Six months later, the ARB announced that due to some issues they can't fix, the ARB was taking others directions without any details. Six months latter again, the OpenGL 3.0 monster was released at Siggraph 2008 and a whole community got really angry! Few constructive debates were on at that time.

Revolution through evolution

Fortunately, the ARB evolution was still on and I would like to especially thank Bob Barris from Blizzard for it. I think he made the ARB and the community involved. After the OpenGL 3 failure, he gives the impulse for an evolution by speaking to the community, trying to make the debate more constructive and even asking for our opinions, our needs, how we would like to see OpenGL evolved and not change (state as a community dream). Interestingly, I felt that we saw a direct feedback from these discussions in OpenGL 3.1 specification.

OpenGL is Open

These community contributions progress again when Barthold Lichtenbelt “nVidia OpenGL ARB chair” and “Khronos OpenGL ARB Steering Group Chair” becomes more and more present to the

point that few days ago he asked on a OpenGL post to the community, to summarise our ideas for the last ARB face to face meeting. We made a proposal with all our ideas 35 entries; I fell quite well about it and the ARB evolution. Thanks to Barthold Lichtenbelt for this.

All this makes the community happy, it makes OpenGL a better API and it makes the community members becoming contributors for the ecosystem: Developing tools, libraries, code samples and helping each other on the forum.

Final though

Is OpenGL on trend? The Apple strategy and the company market share growing; the embedded devices market growing; the OpenCL enthusiasm; the 3D graphics reaching the web with WebGL; the game industry contributions; the support by nVidia, ATI and Imagination Technologies; the ARB evolution and a dynamic community. Yes and my heart is biting again for OpenGL!