John Carmack Archive - Slashdot Archive

http://www.team5150.com/~andrew/carmack

July 6, 2008

Contents

Chapter 1

Posts

1.1 Linux - Where Carmack Goes Next - Nov 1999

1.1.1 Re:More Open Source than we give him credit

There have been a few things that didn't have prior art that probably could have been gotten past a patent examiner – constant Z perspective texturing in DOOM, surface caching in Q1, and the overbright gamma table stuff to trade range for precision in Q3, for example. The patent issue came up at Id a few times until a made it perfectly clear that if the company pursued any patents, I would leave.

1.1.2 Re:Off the rails at last!

Hey, I didn't say "virtual reality"... I tend to agree with your assessment. VR is a term loaded with high-enthusiasm / low-results connotations. We have worked with a few VR companies in the past, and I have always found them to not have finishing ability. So much of the VR world (and much other academic style research) is high concept, but sketchy on the details. Most VR experiences are heavy on the "You are in a virtual world!!!!", but don't spend too much time on exactly what you are

supposed to be doing there. Can you poke and prod to find interesting things? What happens when someone pushes you? Can you dodge something effectively? Are the controls linear, or integrated over time? etc. I think that one of my strengths is a blend of idealism and pragmatism that has resulted in good results over the years. In any case, of the half dozen things I listed, I am clearly not going to be able to do all of them, so it may be a moot point...

1.2 SGI Steps out of the Visual Workstation Market - Nov 1999

1.2.1 Re:Changing Market

I am typing this on a loaded SGI 320.

When it debuted, it was a very good all around performar, and it had the highest fill rate of any intel based system.

Now, an Nvidia GeForce is just plain superior in almost every aspect. Higher fill rate, even in high res, 32 bit, trilinear modes. Faster, more capable geometry acceleration.

Any remaining areas of SGI superiority are probably due to driver optimization rather than hardware architecture. Nvidia hasn't had much call to optimize stippled lines, for instance.

The super-memory-system wasn't all it was touted to be. It worked well for sharing the load between the graphics and the cpu, but the cpu didn't actually see any better bandwidth than on a standard intel chipset. The cpu write bandwidth was actually about 10% LOWER than a consumer machine.

I have used a lot of intergraph and sgi machines, and the bottom line is that the consumer hardware has just outpaced the workstation hardware because they were on different growth curves. The workstations are better than they have ever been, but the consumer systems are just orders of magnitude better than they used to be.

I think SGI shot too low with the VW's graphics, somewhat out of fear of canibalizing their other workstations, and somewhat out of underestimating the consumer competition. Being quite a bit late didn't help, either.

1.3 Review:Toy Story 2 - Nov 1999

1.3.1 Re:Too bad

For back end rendering, they have a room full of MP sparc boxes. To my "SPARC? Why use the slowest of risc processors?" question, they replied that it isn't the speed of the individual processors that was important to them, but the speed PER CUBIC FOOT OF SPACE. Sun made quad pizza boxes, so it was computationally dense.

For modeling and development, they use a lot of SGI octanes. They also use linux + mesa for some internal tools.

1.4 Games - Another Software Spy - Nov 1999

1.4.1 The straight answer

This has been discussed before, and has been going on with the previous tests.

The message of the day server was intended as a half-assed auto update feature that could be cross platform.

We send a normal message most of the time, but if the version is out of date, we can send a message with telling you where to get the update.

I didn't want to deal with binary auto-updates on three platforms, and I worry a bit about security issues with that in any case.

You can disable it by setting "cl_motd 0" when the game starts up if you

really don't want to send anything or see our message.

We added the result of glGetString(GL_RENDER) to get some much needed information about the distribution of video cards and drivers.

We can see how many people aren't following directions and running glsetup. This is a big support issue.

We can see how many people are running minidrivers, which are going to make our lives a mess in the future.

We can see how many mac (steady 5%) and linux (5%at initial release, tailed off to 2%, probably due to dual booting) people are playing.

Getting this information has been usefull. We can compare the numbers of people playing with a given card with the amount of support emails we field, so we know which vendors (3DFX) we need to give more crap about their driver quality.

1.4.2 More comments

When the article first showed up, I thought "It IS documented in the release!". I went and looked, and unfortunately, that documentation from the previous release didn't make it into the latest release. Sigh. Our fuckup.

Apropriate quote: "Never attribute to malice what can be explained by incompetence".

I remain unconvinced that we have done something morally offensive.

Yes, we could have (should have, meant to) included a notice that it was going on in the EULA, but honestly, how many people carefully read and consider every line of all the EULA's they click through? How much of a difference would that have made to people?

I dislike lengthy legal verbiage, but it is reactions exactly like these that cause them to grow. Every time someone says "Sue 'em!" over something, a lawyer proposes another paragraph in a license document.

The most upstanding thing to do would be to have explicit UI that asks

on installation if you don't mind sending your data when you play multiplayer games. I would consider that justified if we were sending a detailed system spec. That is something we may want to do in the future. Data like that is helpfull in making good development decisions.

But this is just a driver string riding along with your game version. It just seems silly, like requiring you to acknowledge before leaving your house that someone might see you. I would rather have fixed a bug somewhere.

I can see that it is a slipperly slope to be on, and I can easily project it to a scenario that I would be offended by, but I just can't convince myself that knowing the reletive distribution of different OpenGL implementations is violating people's rights.

The system was set up to allow us to notify people with a one-line message when their versions are out of date. I imagine some people are offended even by that, but I consider that a positive service to the community.

Including the renderer string was an afterthought to get some good unbiased data to help make future decisions on. Every once in a while we tally up the numbers, then dump all the logs. That's it.

1.5 Games - Quake 1 GPL'ed - Dec 1999

1.5.1 Mac glquake should be pretty easy now

Producing a mac version of glquake or glquakeworld should be pretty easy with the existing code now that Apple has real OpenGL support.

Producing a version of the software renderer with decent performance would be VERY HARD. A huge amount of effort went into the assembly optimization for the PPC, and it still didn't quite measure up to the x86 code.

1.5.2 Re:Just speculation...

Heh. You don't know how much trouble it is to convince biz oriented people that this isn't just plain stupid.

While thinking in terms of money and profit are probably good ways of understanding the way most things work in the world, don't let yourself become so jaded or cynical to think that it is the ONLY way things work.

I do think The World Would Be A Better Place if all software companies released older code so users still interested could work with it or learn from it. (I'm not holding my breath, though)

1.5.3 Re:Level maps *are* GPL'd

Nope. We are the copyright holder of all works, and we can release any part of it under any license we choose.

Completely aside from that, I think it is still unclear exactly where the GPL wants the separation of code and content.

Few would argue that every document read by a GPL word processor would be covered by the GPL, and most would place maps entered by quake into that catagory, but things can quickly get murky.

Quake game mods are written in QC, but turned into data to be processed by the main code. I think the spirit of the GPL would want that code to be released, but it is only a small step from there to saying that every program loaded by a GPL operating system must be GPL, which is clearly not the case.

1.5.4 Re:Just thought this was important to say

How about this:

Make a closed source program that acts as an exe loader / verifier / proxy for the open source main game.

1.6 Games - Quake GPL Release Causing Cheating - Dec 1999

1.6.1 Some more depth

First, the Quake architecture of (reletively) dumb clients conencted to an authoritative server prevents the egregious cheating possible in some games ("I say you are dead now!", "I say I have infinite ammo!").

For the most part, a cheating client can't make their character do anything that couldn't happen as a result of normal game interaction.

The cheating clients/proxies focus on two main areas – giving the player more information than they should have, and performing actions more skillfully.

The "more information" part can take a number of forms. A reletively harmless one is adding timers for items and powerups. Good players will track a lot of that in their heads, but a simple program can "remind" players of it.

Media cheating provides more information. Changing all the other player skins to bright white and removing all the shadows from a level give players an advantage not within the spirit of the game. Some would say cranking your screen brightness and gamma way up is one step on that path.

More advanced clients can make available information that is not normally visible at all. The server sends over all of the entities in the potentially visible set, because the client can move around a fair amount between updates. This means that the client is often aware of the locations of players that are around corners. A proxy can display this information in a "scanner window". The server could be changed to only send over clients actually visible, but that would result in lots of players blinking in and out as you move around or turn rapidly.

The worst cheats are the aim bots. In addition to providing more information, they override the player's commands to aim and fire with very high accuracy. The player usually "drives" around the level, and the pro-

gram aims and shoots for them. This is usually extremely devestating and does ruin the game for most people.

There are many possible countermeasures.

There are server-side countermeasures that look for sequences of moves that are likely to be bot-generated and not human-generated, but that is an arms race that will end with skilled human players eventually getting identified as subtle bots.

Media cheats can be protected by various checksums, as we do in Q3 with the sv_pure option. This is only effective if the network protocol is not compromised, because otherwise a proxy can tell the client that it's hacked media are actually ok.

If the network protocol is not known, then the extra-information cheats generally can't happen unless you can hack the client source.

Q3 performs various bits of encryption on the network protocol, but that is only relying on security through obscurity, and a sufficiently patient person with a disassembler can eventually backtrack what is happening. If only they would find something more usefull to spend their time on...

With an open source client, the network communication protocol is right there in the open, so any encryption would be futile.

Any attempt at having the client verify itself isn't going to work out, because a cheating client can just always tell you what you want to hear. People have mentioned nettreck several times, but I don't see how a completelty open source program can keep someone from just reporting what it is supposed to for a while (perhapse using a "real" copy to generate whatever digests are asked for), then switching to new methods. Anyone care to elaborate?

I think a reasonable plan is to modify QW so that to play in "competition mode", it would have to be launched by a separate closed-source program that does all sorts of encryption and verification of the environment. If it just verifies the client, it would prevent the trivial modified client scanners and aim bots. It could verify the media data to prevent media information cheating. To prevent proxy information cheating and aim bots, it would have to encrypt the entire data stream, not just the

connection process. That might have negative consequences on latency unless the encrypter is somehow able to be in the same address space as the verified client or scheduling can be tweaked enough to force task switches right after sends.

In the end, it is just a matter of making it more difficult for the cheaters. If all it takes is editing and recompiling a file, lots of people will cheat. This is indeed a disadvantage of open source games. If they have to grovel over huge network dumps and disassemblies to hack a protocol, a smaller number of cheats will be available.

Even if the programs were completely guaranteed secure (I havem't been convinced that is possible even in theory), an aim bot could be implemented at the device driver level.

It would be a lot more work, but a program could be implemented that intercepts the video driver, the mouse driver, and the keyboard driver, and does bot calculations completely from that.

Kind of sucks, doesn't it?

1.7 Games - ESR on Quake 3 Troubles - Dec 1999

1.7.1 Re:how netrek really works

Thank you.

Lots of people were just waving their hands saying "Netreck solved this years ago" without being specific.

As you say, it isn't really a solution. If there were a couple million people playing netrek, there would be cheat proxies for it just like there are for Quake.

I think everyone can agree that it just isn't possible to solve the problem completely, you can only make it more difficult, which is exactly what the netrek solution is – release binaries without the exact source and make it difficult to decypher.

1.8 Developers - Tim Sweeney On Programming Languages - Jan 2000

1.8.1 Re:Who says,,,

An Nvidia GeForce/quadro kicks the crap out of a $100k+ SGI reality engine on just about any test you could devise.

The reality engine is quite a few years old, and its follow on, the infinite reality and IR2 still have some honest advantages. You can scale them with enough raster boards so that they have more fill rate, and they do have some other important features like configurable multisample antialiasing and 48 bit color.

Even now, lots of applications can be compiled on both platforms and just run much better on WinNT+nvidia than on the best sgi platform available. Having a 700mhz+ main processor helps, irrespective of the graphics card.

Some people still have a knee-jerk reaction that claims that ”real applications” still run better on the high end hardware, and there are indeed some examples, but the argument bears a whole lot of resemblence to the ones put forth by the last defenders of ECL vector supercomputers.

Most people’s applications don’t show advantages on high end SGI hardware.

The big issue is the pace of progress – SGI aimed for a new graphics family every three years with a speed bump in between. The PC vendors aim for a new generation every year with a speed bump at six months.

1.9 Games - John Carmack Enforcing the GPL on Quake Source - Feb 2000

1.9.1 Re:Secure Quake

There are valid, legal ways to provide a level of protection equal to closed source binaries (which is really only a level of obfuscation).

I realize that they (proxies / loaders / obfuscated modules) may be more of a hassle, but he doesn't get to choose to break the license to avoid a hassle. I traded several emails with Slade over the past month, and I still have a degree of sympathy for his position, but I can't just let him walk around the code license.

All the conspiracy theories about me wanting to destroy the Quake community are silly. I loved what happened with the DOOM source release, and I hoped that the Quake release would have similar effects.

1.10 Games - Dave 'Zoid' Kirsch Leaving id Software - Feb 2000

1.10.1 Re:What the hell is ID doing?

I had talked with Zoid about it a few times over the years, and I think he is making a pretty good decision.

It wasn't always clear from the outside, but Zoid was a remote contractor, not an employee. It was a low key relationship that worked out well for all of us. He stayed in canada and basically worked on whatever he liked, because I thought he had pretty good judgement. He had responsibility for the linux ports and the CTF code, but much of his time was his to allocate as he wanted.

With Loki now picking up the maintenance of the linux port (as well as my steadily increasing involvement with linux), and a new game design

starting at Id, his choice was basically to either go develop a brand new Q3 mod by himself, or go work for one of the many gaming companies that had been trying to hire him.

We weren't interested in bringin on another core programmer at id, especially another one with immigration hassles (we have had enough issues with that for a small company). We would have been happy to continue the current arrangement indefinately, but he wanted to get out of the holding pattern.

Another thing he mentioned that I am sympathetic to is the desire to get a bit out of the community limelight. Being a public figure of some note isn't always all it is cracked up to be.

We are parting on the best of terms (leaving right AFTER a project completes is the considerate way to go). He is going to finish up the Quake2 linux updates (better X/GLX support), even if he has to complete the work from his new job.

1.11 Bsd - Squid, FreeBSD Rock the House at Caching Bake-Off - Feb 2000

1.11.1 Re:Very true

Static web serving is not problem (once you debug the code).

Nothing is a problem once you debug the code.

1.12 Tera Will Buy Cray Research - Mar 2000

1.12.1 Supercomputers

I have been following both Cray and Tera for many years now. I have been saddened watching the last of the supercomputer companies wither and

die. Supercomputers were always so COOL, but for most things, they just aren't so "super" any more.

I have benchmarked several of my back end utilities on cray systems, and one of them on the early tera machine (the early compiler exploded on the others). None of the single processor runs were as fast as a pentium III, and this was quite some time ago.

Understand that this was often branchy and recursive code running with only 3D sized vectors, so it isn't the sweet spot for traditional supercomputers. If I was doing nothing but multiplying 1k by 1k matricies of doubles, even a five year old cray would kick the crap out of the latest athalon. Unfortunately, none of my code looks like that.

I even spent some time thinking how I could restructure calculations into a vectorizable form, which might make a cray J90 competative. I wanted to buy a Cray! Of course, this was silly. It took less effort to make the code SMP friendly, and the payoff was much larger.

We wound up with a 16 processor SGI origin-2000 system, which has been easy to develop for and predictable in performance. We just recently bought an 8 processor Xeon, which is actually faster than our old 16 processor SGI, but at exactly one tenth the price (the downside is that it is maxed out, while the SGI still has tons of growth potential).

I program all heavy workloads in a parallel fashion now as a matter of habit, but it is easy to overstate the benefits of parallel systems.

The common lunux advocate position of "beowulf makes supercomputers obsolete" isn't quite right. Even with code that is already written in a parallel manner, there is a large difference between developing for a shared memory system and a distributed cluster. Developing a compute (and especially data) intensive program for a cluster rather sucks.

If there was a single processor system that was really four times as fast as "consumer" machines, even if it cost fifty times as much, we would buy it. Unfortunately, there isn't. When the product release cycles are favorable, Alpha systems may be twice as good as x86 systems, but not much beyond that unless you are doing the 1k by 1k matrix type stuff.

It is often forgotten that the original Cray-1 was largely a success because

it was the fastest SCALAR processor of the time. The vectorization was just a bonus. Now, vectorization is the only thing that gives them a reason for existance.

The tera architecture is very interesting, but for scalar code, it is VERY, VERY slow. I'm not sure if it will be competative with large processor count SGI origin-2000 systems even after it matures. It gains ease of programming from the lack of caches, but it gives away a lot of problem domains where it is going to look stupidly slow.

If a supercomputer company could make a scalar processor that ran many times faster than existing processors and had similar SMP capabilities, it would probably be a success. Even if it cost a million dollars, filled a room, and burned hundreds of kwatts.

The problem isn't really that supercomputers are bad, its just that we are so spoiled by how AMAZINGLY GOOD our cheap consumer hardware is.

I do still worry about the stiffling of innovation that comes from having so few architectural directions for systems, but in the end, wall clock performance is what really matters.

1.13 Games - New Atari Jaguar Game Running $1,225 on eBay - Mar 2000

1.13.1 Ah, the Jaguar...

I actually dug up all my old jaguar development hardware to give to these guys a year or two ago.

Unfortunately, it turned out that I had lost the C compiler that I had retargeted to the jaguar RISC engines, so DOOM was no longer buildable.

There is something noble about developing on a dead platform – it is so completely for the joy of the development, without any commercial motivation.

The quick recap on the jaguar:

The memory, bus, blitter and video processor were 64 bits wide, but the processors (68k and two custom risc processors) were 32 bit.

The blitter could do basic texture mapping of horizontal and vertical spans, but because there wasn't any caching involved, every pixel caused two ram page misses and only used 1/4 of the 64 bit bus. Two 64 bit buffers would have easily trippled texture mapping performance. Unfortunate.

It could make better use of the 64 bit bus with Z buffered, shaded triangles, but that didn't make for compelling games.

It offered a usefull color space option that allowed you to do lighting effects based on a single channel, isntead of RGB.

The video compositing engine was the most innovative part of the console. All of the characters in Wolf3D were done with just the back end scalar instead of blitting. Still, the experience with the limitations and hard failure cases of that gave me good amunition to rail against microsoft's (thankfully aborted) talisman project.

The little risc engined were decent processors. I was surprised that they didn't use off the shelf designs, but they basically worked ok. They had some design hazards (write after write) that didn't get fixed, but the only thing truly wrong with them was that they had scratchpad memory instead of caches, and couldn't execute code from main memory. I had to chunk the DOOM renderer into nine sequentially loaded overlays to get it working (with hindsight, I would have done it differently in about three...).

The 68k was slow. This was the primary problem of the system. You options were either taking it easy, running everything on the 68k, and going slow, or sweating over lots of overlayed parallel asm chunks to make something go fast on the risc processors.

That is why playstation kicked so much ass for development – it was programmed like a single serial processor with a single fast accelerator.

If the jaguar had dumped the 68k and offered a dynamic cache on the risc processors and had a tiny bit of buffering on the blitter, it could have

put up a reasonable fight against sony.

Now the LYNX, on the other hand, was very much The Right Thing from a programming standpoint. A fast little processor (for its niche), a good color bitmapped display, and a general purpose blitter.

Price and form factor weighed too heavily against it.

1.13.2 Re:Ah, the Jaguar...

Actually, I believe you're referring to Carl Forhan of Songbird Productions

Heh, sorry... I just assumed all jaguar development was coming from a single crazy group. :-)

Even if the memory controller hadn't been broken, performance would still have sucked really bad without a cache.

The jaguar was definately significantly hampered by its technical flaws, which kept me from ever being too big of a jaguar booster. I was proud of my work on Wolf and DOOM (more so than just about any of the other console work Id has been involved in until just recently), but in the end, the better consoles won the war.

1.13.3 Re:Way off topic, but I'm curious since it's "you"

I was only into the Apple II/IIGS during the Amiga's strong times, so I never really got to give it a fair evaluation. My impression of the Amiga is mostly colored by later years of fanatics hounding me about supporting the "inherently superior amiga" when it was obviously well past its competative prime.

1.13.4 Re:Fanatics, zealotry, and dead platforms

I mean that I never actually worked with low level register programming specs for the amiga, so I can't comment authoritatively. The reason is that

when I was young and the Amiga looked interesting, I couldn't afford one. When I had the means, I no longer had the desire.

I certainly don't mean to imply that all Amiga users are fanatics, just that the advocates that made it to my mailbox were less well mannered than those for many other platforms. You are right, it did color my response.

So, to give you a somewhat better answer:

The Amiga's success was in demonstrating the large benefits of specialized graphics coprocessors for personal computers, and providing close to a workstation like environment while the PC was still struggling with segment registers in dos.

It wouldn't have been obvious at the time, but the Amiga was basically fated to go the way of a console generation, rather than evolve as the PC or mac did.

The reliance on low level hardware knowledge and programming provided the obvious visual superiority, but also locked it in to a very ungracefull evolution.

1.14 Features - NVidia and Linux Troubles - Mar 2000

1.14.1 Nvidia's drivers will have strong points

I have been working on the utah-glx project for about nine months now. I am proud of what we have accomplished, and I think it has been a good example of a working open source project. Matrox and ATI have been pleasantly surprised at how well things have worked out.

However, there are only a half dozen coders working part time on utah-glx, and we are split among three active chipset trees. Nvidia has more people than that working full time exclusively for their chips. We are pretty good. So are they. We can work from specs. They can go interrogate the designer of the hardware. It's a pretty simple equation - I expect

their driver to be better than our drivers.

Nvidia is working to maintain a common source base between their windows drivers and their linux drivers. Bugs tracked down by the order of magnitude more windows users will be fixed automatically in the linux version.

DRI does not have all of its problems solved, and there are valid reasons for them to not use it. They might change their mind later.

It should be remembered that some people want to do 3D graphics on linux and don't care about open source principles. Most of the people coming from a technical workstation background just want a vendor to deliver a tool to help them get their work done. I also suspect that most game players will choose a faster driver, even if it is closed source.

The choice isn't between making their driver open source or closed source. They CAN'T open source it because of legal encumbrances on the code. The choice is between doing a closed source driver with their existing code, and doing a completely new driver. Not too many people get excited at the prospect of rewriting perfectly good code.

If you care about getting open source drivers, support Matrox, 3dfx, or ATI. They have released specs to the community, and put out cash for PI to develop and support DRI drivers.

If you just want good 3D, I think nvidia will satisfy you. As for not being done yet, it hasn't been that long since Xfree 4.0 shipped.

1.15 The Dual 1GHz Pentium III Myth - Apr 2000

1.15.1 Re:Wow

A GeForce should be able to run Q3 at 200 fps at 400x300 (r_mode 1) or possibly even 512x384 resolution if the cpu was fast enough. A dual willamette at the end of this year will probably do it.

We currently see 100+ fps timedemos at 640x480 with either a 1ghz pro-

cessor or dual 800's, and that isn't completely fill rate limited. DDR GeForce cards are really, really fast.

Yes, it is almost completely pointless.

The only reasonable argument for super high framerates is to do multi frame composited motion blur, but it turns out that it isn't all that impressive.

I did a set of offline renderings of running Q3 at 1000 fps and blending down to 60 fps for display. Looked at individually, the screenshots were AWESOME, with characters blurring through their animations and gibs streaking off the screen, but when they were played at 60hz, nobody could tell the difference even side by side.

Motion blur is more important at 24hz movie speeds, but at higher monitor retrace rates it really doesn't matter much.

There are some poster-child cases for it, like a spinning wagon wheel, but for most aspects of a FPS, realistic motion blur isn't noticable.

Exagerated motion blur (light sabers, etc) is a separate issue, and doesn't require ultra-high framerates.

There are still plenty of things we can usefully burn faster cpu's on...

1.16 Apple - Apple Announces Darwin 1.0 - Apr 2000

1.16.1 Time to contribute

I was elated when Apple announced the original Open Source Darwin initiative. I never would have guessed they would go for it, and I think it is a Very Good Thing.

Getting everything together for a public release is a very non-trivial task. I know the hassles we go through, and darwin is 100x the size of our codebase.

After all that work, including pressing CD's, it was met with a fairly resounding silence.

The darwin mailing lists were dead. It sometimes seemed like there were a grand total of a dozen people with darwin installed.

It was looking like this might go down as a large example of how going to the trouble of Open Source doesn't get you anything but hassle.

It didn't help that darwin was basically unusable by itself, because all you got was a single very slow text console with messed up key bindings. Not exactly a happy development environment.

(most of the active development work is done in the usable environment of OS-X server)

The general response that interested people gave as to why they weren't doing any development with darwin was that "everything is going to change in the next release" (the driver architecture was massively reworked).

Well, the new release is here now. There is still the problematic issue that you can't run ANY current gui on darwin 1.0. OS-X server and the developer seeds of OS-X client are both out of sync with the darwin codebase. All the excuses won't really go away until the next OS-X client release.

A couple months ago, I took on the porting of X windows to Darwin, so it could actually be considered halfway usable by itself.

I released the patches to get X windows running under MacOS-X server, which was basically the same core as the earlier darwin release.

I was then given the same excuse as other people – why bother porting to the native darwin video and input drivers if everything is going to change soon?

As of now, I am actively feeling guilty about not finishing it. Everything is there for me now, I just need to find the time.

I had been spending my weekends on either GLX or darwin X server work after Q3 shipped, but my R&D "research" has shifted to "development" faster than I expected, and the past few weekends have been monopolized by new engine work. I'll get to it within the next month, but if

someone wants to pick up first, feel free...

It may turn out that many of ESR's arguments just don't pan out for Apple, as far as having outsiders improve the core codebase. Even so, releasing the source will benefit Apple by giving application developers the "ultimate docs" on the OS.

I think Apple deserves a lot of credit for the step.

1.17 The End Of The Road For Magnetic Hard Drives? - Apr 2000

1.17.1 Re:Performance issues.

You can't just choose to rotate a drive platter 100 times faster. It may be remotely possible to make a 50,000 rpm drive in the near term, but 100x faster (and larger diameter) is WAY beyond the limits of known materials. It could also double as a dandy KE tank killer or space launch system if it could actually be built...

The limitation isn't the ability to read the data off the platter, it is the ability of the platter to not break into shrapnel.

1.18 Games - Hasbro And Game-Design Lawsuits - Apr 2000

1.18.1 Re:Reality check

Apogee distributed the shareware trilogy of Wolf3D.

Id wrote the game.

1.19 Apple - Mac OS Mach/BSD Kernel Inseparable: No 'lite' vers - May 2000

1.19.1 Re:Remember...

I specifically asked Steve Jobs about this last time I talked with him (several months ago), and he said that terminal won't be hidden away.

Not that he isn't allowed to change his mind about things...

I was pushing for including at least the command line compile/link tools with every install, but NeXT had an established history of having the development tools all on a separate CD, so it doesn't look like that is going to happen.

1.19.2 Re:I don't get it

The real answer is just inertia from NeXT, but there are some true technical advantages to the mach base.

The mach interfaces for virtual memory and task communication have more scope than the standard unix ones. I was rather surprised when I found out that linux memory management is still basically based on sbrk (although you can fake up virtual memory objects with mapped files yourself).

There definately is some weirdness when you can have so many different types of threads: mach threads/tasks, unix tasks (threads also?), AppKit threads, and possibly some form of Carbon threads. They all come down to mach primitives, but they aren't interchangable.

1.20 Yro - Copyrant - Jun 2000

1.20.1 Re:About Quake3's serial numbers....

The "key generators" are all fakes. Some of them look like they work for a while because servers you have visited with a valid key keep a cache to let you in again.

As far as we know, there are no real key generators. If there were, we would have much more significant support issues.

We certainly will drop the CD-in-the-drive-for-single-player check in a future patch, that is our standard procedure after a game's primary sales are over.

1.21 Developers - Programming OpenGL Articles - Jun 2000

1.21.1 Re:It's Too Late For OpenGL

It is interesting watching the way the tides of public opinion flow around some technical issues.

Over the last year or two, it was amazing the amount of panic among hardware companies that Sony caused with the PlayStation 2. Engineers that really should have known better were walking around with a paniced look, thinking "my god, they are going to crush us, we need to rethink everything!". It was disturbing to see PR effect technical people that much.

PS2 is unquestionably the most powerfull console, but it is a straightforward evolutionary step in power, not the "unprecedented leap forward" that it was billed (and perceived) as. People generally realize that now.

Microsoft seems to have captured much of the same sense of technical inevitability with DX8.

DX8 is good. Microsoft has a long history of shipping an initially crappy product (DX3), then aggressively improving it until it is competative or superior to everything else. Many people underestimate the quality of microsoft's products by only forming opinions on early versions, and never revising them.

The crucial advances of DX8 are the vertex and pixel shaders. I think that the basic concepts are strong, and they will give real benefits.

I expect that that functionality will be exposed through OpenGL extensions by the time I need it.

For one thing, DX8 is modeled pretty closely on Nvidia's hardware, and Nvidia's hardware is already fully exposed through their register combiner extension, even somewhat moreso than under DX.

The issue will be finding consensus between the other hardware vendors.

The upside is that not all hardware designs are exactly in line with DX8, and some usefull and interesting features exist that DX8 doesn't expose. It is looking like several hardware vendors are making moves to expose ALL of their functionality through OpenGL extensions to be available when the product ships, rather than at the next DX cycle.

The other issue is still portability. I am 100% committed to delivering our next title on linux and MacOSX (NOT MaxOS-9), in an effectively simultanious timeframe. That would be more troublesome if I was gung-ho for DX8.

I'm happy that microsoft is doing a better job, but I don't feel that I will be in a disadvantaged position continuing to work with OpenGL.

1.21.2 Re:It's Too Late For OpenGL

Last I heard, Nvidia was going to be providing OpenGL for the X-Box. If they do, we will probably do some simultanious development for X-Box. If not, it would have to wait until after the game ships to be ported.

The X-Box specs put it as a larger leap over PSX2 than PSX2 is over Dreamcast, but anyone with sense can see that by the time it ships, the hardcore

gaming PC will already be a generation ahead of it in raw power.

The X-Box should be able to keep up for a while, because you can usually expect to get about twice the performance out of a fixed platform as you would when shooting for the broad PC space, just because you can code much more specifically.

I don't have much of a personal stake in it, but I am pulling for the X-Box. If you need to pick a feudal lord in the console market, I would take microsoft over sony/sega/nintendo any day.

1.22 Science - Inventor Building Rocket In Backyard - Jun 2000

1.22.1 Aerodynamically unstable!

At first I thought it was just bad reporting, with "Most of the weight will be behind, and gravity will keep the rocket pointed upward", but seeing the picture on his site backs that up.

Putting a big, fin-looking cockpit ahead of the fuel tank mass is going to make every breeze cause a heading change.

His site goes on with:

"What about guidance systems? The thrust will come out at the top of the rocket. An early American pioneer Robert Goddard did the same thing with his early test rockets. The rocket should "hang down" from the thrust like a pendulum"

That DOESN'T WORK.

It doesn't matter if a rocket is being pulled or pushed, all that matters is the relationship of the center of gravity to the center of pressure.

The reason why the intuitive "hangs like a pendulum" doesn't work out is that gravity acts on a deflected pendulum in a direction out of line with the pendulum string, while a rocket thrust will always be in line with the

body.

1.22.2 Re:What kind of unstable?

A pendulum has a pivot point, so when gravity tries to pull the center of gravity towards the earth, the linear acceleration is converted to a rotation torque around the pivot point, swinging the pendulum back down..

A rocket isn't held by anything, so the force of gravity will only pull it downwards, not cause any rotation. Gravity can't cause a rotation (ignoring very large scale gravity gradient issues), only aerodynamic forces.

Any wind will cause a rotation based on the CG/CP relationship, which will not be corrected by forward aerodynamic forces in this case because CP is forward of CG.

The truth is that I used to think along the same lines as this theory, but I built a couple models to test it, and they were complete failures.

After thinking about it for a while, I realized the difference between hanging from a pivot and having a force along the body.

1.22.3 Re:What kind of unstable?

All forces acting on a body can be summed to a vector through the CG and a rotation around it.

The thrust coming out at the nose or tail doesn't matter, it will still act through the CG.

—R ! ! ! ! CG !

With a rocket exhausting down from R, there will be both an upwards acceleration and a clockwise rotation. That obviously won't fly straight.

——R ! R ! ! ! CG !

With two rockets (or any symetric number), the rotational forces cancel out, leaving just a forward acceleration acting through CG. Again, no

matter where the forces are applied to a rigid body, they act through CG.

This rocket will fly straight in an airless vaccuum, or in a perfect world with non-moving air and EXACTLY balanced engine thrust.

When a body has a center of pressure that isn't exactly at the center of gravity (almost everything but symetric and uniform blocks of material), a sideways gust of wind will cause a slight rotation of the rocket around around CG.

——R ! R ! !W ! CG !

If a wind force acts to the left at W, it will cause an acceleration to the left through the CG and a counter-clockwise torque around CG. The existance of other forces on the same body have no effect whatsoever on this. The rockets don't thrust "down" (which WOULD cause a corrective force), they thrust "along the rocket".

A "stable" rocket will have the CP behind the CG, which causes the much larger forward aerodynamic forces to swing the rocket back towards it's direction of travel. That's why there is a minimum stable launch speed for unguided rockets – the forward aerodynamic forces have to be larger than the sideways winds.

An "unstable" rocket with CP ahead of CG will fly straight as long as there are no winds and it is pointed exactly in it's direction of travel. As soon as there is a slight rotation, the forward aerodynamic forces push it in the same direction as the existing disturbance, reinforcing it into a rapid spin.

Direction of travel determines the orientation of CG and CP. This rocket will be stable when falling down, just not when flying up.

Again, the pendulum is different because it is not a single rigid body. If you didn't hook a bendulum to anything, it would fall without any rotation in an airless space, and would fly with it's CG (the ball) ahead of it's CP if thrown.

1.23 Games - John Carmack on the X-box Advisory Board? - Jul 2000

1.23.1 Substantiated.

Geez, I don't think this really rates a news story...

I put off an interviewer with questions about the X-Box by saying that I was on the X-Box Advisory Board, and probably shouldn't discuss specifics, instead of just my usual "sorry, too busy" reply.

Here is the longer answer:

At last years CGDC, Tim Sweeny and I had a meeting with Bill Gates about the X-Box. It was not handled well.

For weeks ahead of time, I had been pressing for technical information so I could have something useful to comment on at the meeting. A couple days before the meeting, I finally got an email directing me to "look at this EETimes article, they are pretty close". Yeah. Ok.

So, we just wound up just talking about generalities.

A while later, I was contacted about being on the formal advisory board, with a promise that it wouldn't be like that "trophy meeting" at CGDC, but would be making critiques of real documents.

I am on a lot of advisory boards, and they vary quite a bit in level of participation.

3DFX's advisory board meets every quarter, and we go over detailed technical things. Unfortunately, the very first advisory board of over two years ago discussed a part that still hasn't shipped, so it is hard to say what the impact is.

Apple's gaming advisory board has met three times, and was moderately productive.

Nvidia listed me as a member of their technical advisory board in their IPO filing, but there has never been a group meeting. I meet with them

a couple times a year privately, but I haven't had a whole lot to complain about or suggest to them since they got past the RIVA 128 (until the recent push for 64 bit color)– they have been doing a great job.

All of the other companies just informally stop by everey once in a while to discuss things.

I had made some suggestions to microsoft about DirectSound and DirectInput in past years that were always at the wrong time to ever get acted upon, so I don't know what to expect from this board.

So far, microsoft seems to be sticking to the plan – I got a big fat binder of stuff in today to look over before our meeting next week.

I'm all for the X-Box as a console platform. The graphics hardware is a lot cooler than PS2, and there are a lot of other things going for it. I am still uneasy about all the market protection issues that go with consoles, but I tend to think that microsoft is a more open company than many of the traditional console companies.

I want microsoft to make good products. Heck, I want everyone to make good products. Even at the height of the D3D vs OpenGL antagonism, I had always given them source drops of what I was working on, and freedom to use it for demonstrating new features.

I had hoped that they would use it as a real-world testbed for new features, rather than just dreaming them up and making the industry follow their plan without ever really testing things out.

In any case, talking with MS has no bearing on my development decisions. I'm still using OpenGL, and we are still planning simultanious releases for linux and MacOS-X. If things work out well with X-Box, that may be added to the list.

1.24 Games - New Doom Details - Aug 2000

1.24.1 Re:That wasn't the carmack we know..

what happened to that? apart from it: can you imagine ID coding something in c++

First of all, the fact that none of our deployed games used OOP does not mean that I don't think there are benefits to it. In the early days, there were portability, performance, and bloat problems with it, so we never chose to deploy with it. However, all of the tools for DOOM and Quake were developed in Obejctive-C on NEXTSTEP. Q3 also came fairly close to have a JVM instead of the QVM interpreter, but it didn't quite fit my needs.

I'm still not a huge C++ fan, but anyone that will flat out deny the benefits of OOP for some specific classes of problems like UI programming and some game logic is bordering on being a luddite. I still don't think it is beneficial everywhere, but, if anything, I think we are behind the curve on making the transition.

Carmack basically said, that single-player games suck from a money/replayability point of view

This was a balancing act in company morale and politics. My first choice for future projects would be to pursue the snowcrash-like extensible virtual worlds, but most of the company wants to work on a game with a story. We have three new hires coming on soon, so we are growing somewhat to support it.

In the same interview he said, that he doesn't like integrated editors, and will never do so.

That was a significant change in my stance over the last year. Did I actually say "never"?

You can just take the short answer of "I was wrong", but here is the longer argument (I went over this some in the talk):

Historically, we could make better games by using exotic and expensive

development hardware or software that differed significantly from our deployment platform.

While early games with integrated editors were hurting their 640k memory footprint and forcing developers to work with a 320 or 640 res screen for tool development, we were using NeXT workstations with megapixel displays for our tools.

Given the option of a good rendering technology that required a lot of preprocessing, we bought first a quad alpha, then a 16 way SGI to do the processing.

Even when we moved the editor to WinNT, it used intergraph workstation graphics, while our deployment platform was mostly still software rendered or possibly Voodoo based.

The key thing that changed (after that earlier speech) was that the optimal development platform is now the same thing as the optimal deployment platform: x86 + nvidia.

That is important. The tools/editor/game HAD to be separate before. Now, the question has to be looked at with a fresh view.

There was a lot of code that was present in nearly identical form in the utils, editor, and game. Misc functions, image loading, model loading, pk3 filesystem support, etc. It was one of my big goals to make all that common.

The obvious step would be to make libraries out of them, but I have something of a personal dislike for managing code that way.

Rather than just endlessly debating the issues, I just took a day and combined the utility code into the main project. It went well, and I was happy with the many thousands of lines of code that got removed, and the increased functionality that resulted. Later, Robert did the same thing with the editor.

I do fret about code bloat issues, but I feel quite good about all of the common and not-quite-orthogonal code that has been removed, and in the scope of the entire project, it really isn't that much space – it adds maybe a megabyte to the executable, but it will never be paged in if you

are just playing the game.

I do think there are real benefits to the user community from having the tools with the game – my earlier objection were always based not wanting to give up any possible advantages that we could have as developers.

The advantages are going to be especially strong for the linux and mac platforms: the entire tool chain will be available from day one on every platform the game runs on.

1.24.2 Re:That wasn't the carmack we know..

Alright. This is wierd. However, it's a natural progression. If we have ingame editing, most likely we won't be using BSP trees anymore, or perhaps we will turn culling off when we walk around. I think such a thing could be handled as a mod in the game. It just makes more sense than some win32 app that has completely different requirements than the game itself

"In game editor" is probably being confused here somewhat. I am NOT talking about running around in the game, moving brushes around as you play. The editor is still a completely different user interface, with multi-view outlined drawing in addition to a 3D view. It just happens to live in the same executable as the game, and shares lots of code with it.

1.25 Games - John Carmack on Consoles vs. Personal Computers - Aug 2000

1.25.1 Linux gaming market

Yes, the linux sales figures were low. Low enough that they are certainly not going to provide an incentive for other developers to do simultaneous linux releases, which was a good chunk of my goal. The sales would cover the costs of porting, but they wouldn't make a bean-counter blink.

I think Loki did a fantastic job - they went above and beyond what was required, pestering us (a good thing in this case) about the linux deliverables, taking pre-orders, doing the tin box run, shipping CDs first, then boxes when available, etc.

There are a number of possible reasons why you might not have bought the linux specific version:

You couldn't find the game in stores near you. This is going to remain a problem for quite some time.

The game is available earlier for windows. Even with a simultaneous release, this is going to continue. Big publishers making large lot runs get priority, and that is just life.

The game costs more for linux. This is probably also not going to change. The wholesale prices are probably the same, but big stores severely discount popular titles and advertise them to bring customers in. This won't happen with linux versions.

Configuring 3D on linux is a significant chore. I expect this will largely be gone by the time we ship another game. As the DRI drivers mature and XF4.0 becomes standard in distributions, people should start having out-of-box 3D support.

The game runs slower in linux than under windows. While we did have a couple benchmark victories on some cards, the general rule will still stand: a high performance card on windows will probably have more significant effort expended on optimization than it will get from an open source driver. Nvidia's drivers may be the exception, because all of their windows optimization work immediately applies to the linux version, but it is valid for most of the mesa based drivers.

Trying to change this would probably have negative long-term consequences. There are certainly coders in the open source community that are every bit as good of optimizers as the driver writers at the card companies, but I have always tried to restrain them from going gung-ho at winning benchmarks against windows. Mesa is going to be with us five years from now, and dodgy optimizations are going to make future work a lot more difficult.

Loki's position is that the free availability of linux executables for download to convert windows versions into linux versions was the primary factor. They have been recommending that we stop making full executables available, and only do binary patches.

I hate binary patches, and I think that going down that road would be making life more difficult for the people playing our games.

That becomes the crucial question: How much inconvenience is it worth to help nurture a new market? We tried a small bit of it with Q3 by not making the linux executables available for a while. Is it worth even more? The upside is that a visibly healthy independent market would bring more titles to it.

The fallback position is to just have hybrid CD's. I'm pretty sure we can force our publishers to have a linux executable in an "unsupported" directory. You would lose technical support, you wouldn't get an install program, and you wouldn't have anyone that is really dedicated to the issues of the product, but it would be there on day 1.

1.26 Tom's Hardware Linux NVidia Benchmarks - Aug 2000

1.26.1 Page flipping should not be supported.

All signs are pointing towards a future without page flipping, so adding the messy infrastructure for it now would be a mistake. Don't let benchmarking furor encourage a messy code architecture.

Points:

The benefit of page flipping is decreasing as more and more computation is done per pixel to the back buffer.

In the old days of 2D scrollers, you might barely cover the screen with one pass of writes, so page flipping could double your speed over blitting.

On a typical modern 3D game that becomes fill limited, under 25% of the performance is in the blit, and often under 10% in scenes with significant overdraw.

In upcoming games that composite 20+ layers of textures, the cost of a blit is down in the noise.

Blits add flexibility. Anti-aliasing is better done through a blit operation than with a deep front buffer. Other operations, like converting from a 64 bit work pixel to a 32 bit display pixel, or performing convolutions, are also better done with blits.

Back buffers are more optimally arranged in tiled patterns, while front buffers prefer linear scans.

Basically, our back buffers are starting to look less like raster

Page flipping doesn't apply to windowed rendering unless you butcher the X server to render all 2D to multiple buffers and clip all 3D operations. I consider that a bad thing. Making the full screen rendering more distant from windowed rendering is also a bad thing.

Every implementation of page flipping brings in a class of bugs, and obfuscates several code paths. It's not worth it.

1.27 Games - Carmack About Q3A on Dreamcast - Aug 2000

1.27.1 Re:The PS2 Is Screwed

Make no mistake – the PS2 is definately more powerful than the dreamcast. For some types of things, it is easier to get a dreamcast game to look better due to a better back end filter, autoamtically working mipmapping, and larger addressable texture space, but the second generation PS2 games should really start showing off the increased power. Dreamcast should be able to undercut the price, but I don't know how significant that will be. There are few things that I would really call "revolution-

ary", but that doesn't mean that Sony didn't build a good machine. It just happens to be built with a set of tradeoffs that I don't completely agree with.

1.27.2 Re:Vector quantization compression?

You have confused two different forms of compression.

S3TC is a modified form of block truncation coding (BTC), which involves selecting two colors and generating two other colors by interpolation. This is done with 4x4 blocks, giving very nearly 4 bits per pixel. This is nice because it doesn't require any additional tables.

Vector quantization is a general process where you try to take a large set of number strings and pick some subset that can be used to aproximate all of them reasonably. In the dreamcast's case, you specify 256 2x2 blocks, so each pixel is represented by 2 bits, but you also have 2k of codebook overhead. This works out pretty well for smaller textures, but large textures often come out badly because there just aren't enough codebook entries to reasonably aproximate it.

1.28 Games - Salon on the XBox - Aug 2000

1.28.1 Developers all want a royalty. NOT.

I don't.

The argument for royalties is that it allows the console price to be lower, allowing more units to be sold, and theoretically allowing you to sell enough more units to offset the royalty.

The downside is that if a large chunk of the console revenue must be derived from software royalties, it must be made impossible to bypass the console company in the production of a title.

This forces them to resort to various copy protection and registered de-

veloper schemes, which open the door to all the back room scheming between publishers and the hardware vendor about shipping sequencing, and content aproval.

I would rather have a console that was six months less powerfull, but 100% completely open, and that anyone could press games for.

(Indrema has not disclosed me on their hardware.)

1.29 Games - VoodooExtreme Interview With John Carmack - Sep 2000

1.29.1 Re:Portals ?

PVS was The Right Thing when level geometry counts were much lower. With tens of thousands of polygons in a scene, creating a cluster tree directly from that becomes completely unrealistic from a space and time perspective.

The options are to either do PVS with a simplified version of the world, or ignore the geometry and just work with portal topology.

Unreal used a scan-line visibility algorithm, which crippled it's ability to have high poly counts or high framerates with hardware accelerators.

Tim Sweeny knows full well that the architecture is very wrong for modern systems, but many of the original decisions were based on earlier software technologies. Unreal was supposed to be a "killer app" for the Pentium-200 MMX processor.

I have a lot of respect for Tim and Unreal, but the visibility algorithm in Unreal turned out to be a bad call. He is changing it for future work.

1.30 The Good Old Days of 3Dfx - Sep 2000

1.30.1 Re:nope

Actually, even the original Verite V1000 could do 32 bit color rendering.

At a whopping 6 mpix or so...

Rendition did a lot of things right, even on their very first card. They had all the blend modes and texture environments from the very beginning. The only thing they didn't have was per-pixel mip-mapping.

If they had delivered the V2xx series on time, they could have had a strong foothold before voodoo2. The V3xx seried would have been a solid TNT competitor, but again, it wasn't ready on time. They wound up ditching that entire generation.

1.30.2 Re:3Dfx vs nVidia vs The World

Reguardless, I won't buy either. Why? Because they both claim OpenGL support. Now, I don't know about you folks, but seeing as I work in AutoCAD frequently, that means hardware support. Neither of them have it. ATI doesn't. #9 didn't. Why? Most of them view software as the future.

All of the modern cards have full rasterization support for OpenGL, but I guess you are refering to geometry acceleration.

The situation has changed since you last looked at it.

The Nvidia GeForce cards have an extremely capable geometry accelerator, and they have the ability to fetch display lists either over AGP with a large bandwidth savings due to vertex reuse, or store the display lists completely in local memory to remove all vertex traffic from the bus.

The issue with professional OpenGL support has mostly been the focus of the driver writers, not the hardware. I think that Nvidia's partnering with ELSA to work on professional app certification with the Nvidia hardware was an extremely good move.

There are a few edges that the expensive professional boards still have over the nvidia consumer cards, but not many:

You can get more total memory, like a 32mb framebuffer and 64mb texture memory configuration. We will probably see workstation graphics systems with up to a gig of memory within a year. Consumer cards will offer 128mb next year, but the workstation cards can easily maintain an advantage there.

This has a cost, though: large, expandable memory subsystems can't be clocked as high as the single-option, short trace layouts that nvidia does. Even dual pipe workstation boards can't match the memory bandwidth of a GeForce2.

You generally get better high end DACs and shielding on workstation boards. The upper end of the consumer boards will do the high numbers, but it just isn't as clean of a signal.

Dual monitor has been supported much better on the workstation boards. This is starting to become a feature on consumer boards, which is welcome.

The consumer cards are still skimping on itterator precision bits. Under demanding conditions, like very large magnitude texcoord values stretched a small increment across a large number of pixels, you can see many consumer cards start getting fuzzy texel edges while the workstation cards still look rock solid.

Probably the most noticable case is in edge rasterization, where some workstation cards are so good that you don't usually notice T-Junction cracks in your data, while the consumer cards have them stand out all over the place.

Next years consumer cards should fix that.

When the consumer cards first started posting fill rate numbers higher than the professional boards, it was mostly a lie. They got impressive numbers at 640x480 in 16 bit color, without blending, depth buffering, and filtering, but if you turned on 32 bit, depth, blend, trilinear, and ran at high res, they could fall to 1/4 or less of the quoted value.

Today, there isn't a single combination of rendering attributes that will let a wildcat out-rasterize a GeForce2.

Wildcat was supposed to offer huge 8 and 16 way scalability that would offset that, but it doesn't look like it is coming any time soon.

The workstation vendors do stupid driver tricks to make CDRS go faster, while consumer vendors do stupid driver tricks to make Q3 go faster.

We bought three generations of intergraph/intense3D products, but the last generation (initial wildcat) was a mistake. We use nvidia boards for both professional work and gaming now. I still think the professional boards are a bit more stable, but they fell behind in other features, especially fill rate. Being an order of magnitude cheaper doesn't directly factor into our decisions, but it would for most people.

1.30.3 Re:nope

Q. After reading the voodooextreme interview, it sounds like you are pursuing an allmost completely different rendering pass/phases with Doom 3. Can you give us any more details? :-)

It adds up to lots and lots of passes. I am expecting the total overdraw to average around 30x when you have all features enabled.

Q. Could you give us your thoughts on T&L? Why does 3Dfx say it's not important?

Contrary to some comments here, 3dfx didn't just "decide not to put in T&L", the didn't have that option. Product development cycles can take years, and you can't just change your mind at the end.

They don't have it, so naturally they downplay the importance of it.

1.31 Science - Cheap Launch Ends in the Drink - Oct 2000

1.31.1 It's harder than it sounds

I have been meaning to write an article about my involvement with, and impressions of the space community over the past year. Slashdot's space coverage started getting me interested in the field last year, and I wound up putting $34k into funding two of the CATS prize entries (JPA and SORAC). If one of them had won, they would have returned the funding money, but it was basically done as philanthropy.

From the outside, or with cursory knowledge, it seems so damn simple, and all the problems sound like they could have been prevented with a little thought. It's harder than it sounds. Feel free to help.

A lot of efforts came together in the last couple months, and none of them have been successful. After the fact, it's easy to toss off what should have been done.

Ky Michaelson's space shot got off better than most. It launched well, and went through it's full burn before losing a fin at mach 4+. If it had held together, it might have made it into space. Hindsight says he should have welded the fins.

The Wickman AN space booster static test CATO'd. Hindsight says he should have hydrotested the casing.

The HARC launch reached the full altitude for balloon launch (an achievement by itself), and the electronics all seemed to work, but the launch failed due to interference with the launch rail. Hindsight says they should have done a test launch from the ground with the launcher hung from scaffolding.

The SORAC team ran into some political and personal issues with the Bureau of Land Management, resulting in them being banned from black rock for a while, scrubbing their planned CATS launch. Being honest, it would have been a rush job if they had launched in the scheduled slot,

but they still would have had a chance. Hindsight says they should have started cultivating the BLM bureaucrats at the start of the year.

JP Aerospace got fucked by the FAA. They have had licenses to try for 100km a couple times in the past, and they always turn everything in more than six months in advance and check up on the process constantly. Four days before they were heading to black rock, the FAA informed them that there were problems with their application. Hindsight doesn't say anything. There was still a pretty good chance JPA would have had problems with their new launch structure because they missed their last AWAY test run schedule, but they probably had the best shot.

Interorbital has a sea launch slot next week, but pushing for the last week of the prize sounds like things probably aren't quite ready to go.

The SORAC and JPA CATS rockets will still be launched when the proper permits are all in line again (the government agencies have turned around), so there is a decent chance that there will be an amateur rocket getting into space come spring, but unfortunately there won't be a payoff for them now that the prize is expiring.

1.32 Games - id on Linux: ”disappointing” and ”support nightmar - Dec 2000

1.32.1 The Official Position

We are going to continue to support linux in future products, but unfortunately it doesn't look like a strong business case can be made for it. The mac version outsold the linux version by quite a bit, and even that didn't hit 5% of the windows sales. Mac versions are still valid business cases, because the support is way easier than on either windows or linux platforms, and the sales numbers amount to something noticeable.

There is no way that a linux box will hit the shelf at the same time and have the same price as a windows box, assuming the publisher is making a maximum effort for the windows box. If this is truly a gating factor,

linux boxed games just won't succeed.

Loki wants to get away from making games "convertable" between platforms, to force linux players to buy the linux boxes. I have issues with this. Not making executable binaries available online sucks. I hate binary patches, and requiring either patches from different versions, or the installation of all previous patches. Just releasing a new executable is so much easier.

Our options from here are to move towards a hybrid CD and pay Loki for official support (which makes linux support look like an expense, rather than a benefit), make a hybrid CD but leave the linux version in an "unsupported" directory, or just make unsupported linux executables available online like we used to.

It is going to be quite some time before DOOM ships, so we can't say anything definitive at this point.

I will probably do the initial development work for DOOM on linux, but I'm not interested in tracking every change that goes on in the linux world. The initial work will probably be with the Nvidia driver, which already has all the features I need, then I will work with the Open Source mesa drivers to bring them up to par.

1.33 Games - Linux Gaming: Looking Back and Looking Forward - Jan 2001

1.33.1 Re:I've re-installed my Windows partition

We are still supporting linux.

The only downside of the next product is that initially it will probably only work at full feature level with the Nvidia OpenGL driver, but after the first test gets out (still a very long time in the future), I will jump back in to the driver development to try and bring the other open source drivers up to par.

1.34 Features - Pride Before The Fall - Jan 2001

1.34.1 How much do you value these methods?

The article lays out all the things Bill should have done:

He should have compromised what he really thought.He should have "played the PR game".He should have coddled bureaucrats.He should have paid attention to "political sensibilities".

From the perspective of Fortune or Business Week, that all sounds right and proper.

But from a hacker perspective?

I'm not saying Gates is a hacker (although he is indeed really damn smart), but if you align yourself with those ideals, is it really correct to deride someone for being forthright and stubborn in the defense of their position?

1.35 A Brief History of NVIDIA and SEGA - Feb 2001

1.35.1 Quadratic surfaces

The article hints that the NV1's quadratic surfaces might have actually been a good thing, and it was held back by Microsoft's push to conformity with triangles.

Er, no.

For several years now, Nvidia has been kicking ass like no other graphics company, but lets not romanticize the early days. The NV1 sucked bad, and it would have been damaging to the development of 3D accelerators if it had gotten more widespread success. Microsoft did a good thing by standing firm against Nvidia's pressure to add quadratic surfaces to the

initial version of D3D.

There is an intuitive notion that curved surfaces are ”better” than triangles, because it takes lots of triangles to aproximate a curved surface.

In their most general form, they can be degenerated to perform the same functions as triangles, just at a huge waste in specification traffic.

Unfortunately, there have been a long string of products that miss the ”most general form” part, and implement some form of patch surface that requires textures to be aligned with the patch isoparms. This seems to stem from a background in 2D graphics, where the natural progression from sliding sprites around goes to scaling them, then rotating them, then projecting them, then curving them.

3DO did it. Saturn did it. NV1 did it. Some people are probably working on displacement mapping schemes right now that are making the same mistake.

Without the ability to separate the texturing from the geometry, you can't clip any geometry in a general way (not even mentioning the fact that clipping a curve along anything but an isoparm will raise it's order), and you either live with texel density varying wildly and degenerating to points, or you have texture seams between every change in density. No ability to rotate a texture on a surface, project a texture across multiple surfaces, etc. You can't replace the generality if a triangle with primitives like that.

Even aside from the theoretical issues, NV1 didn't have any form of hidden surface removal, and the curve subdivision didn't stitch, frustum clip or do perspective. It was a gimmick, not a tool.

All water under the bridge now, of course. NV20 rocks. :-)

1.35.2 Re:Quadratic surfaces

My point was that with texturing tied directly to the patch orientation, you can't do exactly the thing that you describe.

I'm not a big booster of hardware curves in any form, but I only rail against hardware schemes that use aligned textures.

1.35.3 Re:Quadratic surfaces

The hardware curved surfaces in upcoming hardware is basically ok, because you can have all the attributes independent of each other at each control point. My major point was that the 3DO/Saturn/NV1 botched it badly by explicitly tying the texture to the patch orientation, which prevents them from being able to do triangle like things at ALL.

1.35.4 Re:Quadratic surfaces

I was ranting specifically about square patches that have implicit texture alignment, not curves in general. I am on record as saying that curved surfaces aren't as wonderful as the first seem, though.

It was my experience that subdivision surfaces are much more convenient for modeling free form organic surfaces, but polynomial patches can be more convenient for architectural work.

1.36 GeForce 3 Demoed - Running DOOM 3 - Feb 2001

1.36.1 Re:Shades of Grey

Unfortunately, the giant projector screens are not color calibrated the same way at all.

The colors did get rather washed out on the big screen.

1.36.2 Re:Competitive clock speed???? Did I miss somethin

We did a ton of testing the last two weeks while we were putting the demo together.

The 733 G4 was not as fast as my 1 ghz PIII in any of the trouble areas.

Apple is doing a lot of good work, but the CPU's just aren't as fast as the x86 ones.

AltiVec can compensate in some cases, because it is way, way easier to program for than SSE, but it takes a very simple batched, computation intensive task for it to pay off in any noticable way. Amdahls law and all that.

We did a couple functions with AltiVec, but they didn't make much difference.

Video encoding and large image processing are two areas that it can pay off, because you may be spending 90%+ of your time in one page of code.

Even then, it takes a special balance to let a G4 come out ahead, because it has less memory bandwidthd than a high end x86 system.

1.36.3 Re:A question, John?

We moved to C++ for the current game (which does not have an official full name yet).

I will probably do a .plan update about it, because it has definately had its pros and cons.

Jim Dose had inadvertantly used a few MS specific idioms that we had to weed out over the past couple weeks of the bring up on OS-X.

1.37 Games - Yamauchi Puts the Game Industry In Its Place - Feb 2001

1.37.1 Re:Long overdue

Half Life made net profits of just over $200,000

Uh, no. Half Life made FAR more than that.

The top titles still bring in lots of money, but if you don't get a hit, you probably won't recoup your development money.

1.38 Games - Carmack on D3 on Linux, and 3D Cards - Feb 2001

1.38.1 Re:Question to JC about the video

We don't have any technology specifically directed towards character features. The animation was done pretty conventionally in Maya.

Our new animator comes from a film background, and we are finding that the skills are directly relevent in the new engine.

1.39 Science - Telemetry Made Simple: Rocket Phone Home - Mar 2001

1.39.1 Re:The significance of this

It used to be that NASA had to have a string of ground stations and ships around the world to get continuous data from space vehicles. This is still an issue for countries like China, and the logistics have impacted

their space program. NASA does now use a lot of custom relay satellites, but the communications hardware is still $100k+ one-offs for each application. Using $5k of commercial equipment that is probably better developed is indeed a good thing. The articles also lumped together with the LEO sat communication aspect the increasing use of GPS to augment and eventually replace dedicated radar for positioning.

1.40 Science - US Military May Resurrect X-33 - Apr 2001

1.40.1 Re:At least the Japanese are keeping DC-X alive...

My even-more-modest efforts are also ongoing:

www.armadilloaerospace.com

We hope to be making a properly controlled test flight within the next two weeks now that our gyros are working properly.

1.40.2 Re:The clipper was a joke!

Coming back as a glider implies wings and landing gear, which wind up massing more than vertical landing propellant.

Trying to do without wings on X-33 forced a sub-optimal tankage solution, which turned out to not work.

A VTVL isn't expected to "fly" to a landing. It plummets at terminal velocity (not as fast as you might expect, because it is mostly empty), only firing the engines a reletive few seconds before landing. The fuel required is then only a few seconds of several G thrust for the EMPTY mass of the vehicle.

A somewhat more valid argument against vertical landing is that doing it efficiently requires computer control, and it is unlikely that a human

pilot would be able to do it manually without a lot more slop fuel.

1.41 Rockets of Doom From Carmack And Friends - Apr 2001

1.41.1 Re:I was at the meeting

I don't have an orbital timeframe. There are too many things I need to learn before I can make a credible estimate.

The timeframe I do have is:

Year 1: work out all the kinks in the VTVL demonstrator.

Year 2: manned rocket ships and ballistic flight, but still rather low altitudes.

Year 3: space (100km) shots, both unmanned and manned

1.41.2 Re:VTOL problems

The throttle is manual, but attitude control is computer managed. The joystick input gives a target angle, and the computer deals with rates and pulses to try and get it there. Manual control of a differentially throttled vehicle is extremely difficult (the simulator allows you to try).

CG/CP is irrelevent for this vehicle, because it isn't designed to go fast enough that aerodynamics are a factor.

Four fixed position engines can give full 3 axis control if you are tricky about it. Opposite pairs of engines are canted a few degrees so that one pair of engines gives a slight positive roll, and the other pair of engines gives a slight negative roll. This does mean that there is a cross couple for every pitch or yaw adjustment, but with an order of magnitude difference between them, it is easy to correct out.

The difficulty of guidance and control is overrated.

1.41.3 Re:VTOL problems

I should ammend myself, and say that the difficulty AT LOW SPEEDS is not that bad. Removing aerodynamic factors simplifies things a lot.

Doing something like hypersonic kinetic kill missile guidance still sounds, uh, non-trivial.

1.41.4 Thanks, slashddot!

I should mention that really, all of this work can be traced back to slashdot.

Until a year and a half ago, I hadn't thought about space and (real) rockets since I was a kid.

I started reading slashdot for the open source coverage, but the occasional space story and the comments on them led me to the CATS prize and the other things going on in the space community.

I spent a year learning the engineering aspects and funding a few things that I considered interesting (JP Aerospace, SORAC, Space Frontier Foundation, and XCOR), and the last six months actually doing something myself.

I was sort of planning on submitting an article about the whole process at some point, but it looks like I got preempted (and our site is slashdotted)...

1.42 On the Subject of Ximian and Eazel - May 2001

1.42.1 Re:If you want financial information about the FSF

Heh. That sort of takes some of the wind out of the FSF financial conspiracy theory.

Yes, that was my blackjack winnings.

1.43 Hardware - x86 vs PPC Linux benchmarks - Jun 2001

1.43.1 I do this every year...

I wind up doing my own internal PPC vs X86 benchmarks almost every year.

I'll set up whatever current game I am working on to run with the graphics stubbed out so it is strictly a CPU load. We just did this recently while putting the DOOM demo together for MacWorld Tokyo.

I'll port long-run time off line utilities.

I'll sometimes write some synthetic benchmarks.

Now understand that I LIKE Apple hardware from a systems standpoint (every time I have to open up a stupid PC case, I think about the Apple G3/G4 cases) , and I generally support Apple, but every test I have ever done has had x86 hardware outperforming PPC hardware.

Not necessarily by huge margins, but pretty conclusively.

Yes, I have used the Mr. C compiler and tried all the optimization options.

Altivec is nice and easy to program for, but in most cases it is going to be

held up because the memory subsystems on PPC systems aren't as good as on the PC.

Some operations in Premier or Photoshop are definitely a lot faster on macs, and I would be very curious to see the respective implementations on PPC and X86. They damn sure won't just be the same C code compiled on both platforms, and it may just be a case of lots of hand optimized code competing against poorer implementations. I would like to see a Michael Abrash or Terje Mathison take the x86 SSE implementation head to head with the AltiVec implementation. That would make a great magazine article.

I'll be right there trumpeting it when I get a Mac that runs my tests faster than any x86 hardware, but it hasn't happened yet. This is about measurements, not tribal identity, but some people always wind up being deeply offended by it...

1.44 Science - Getting Into Space, One Way Or Another - Jun 2001

1.44.1 Re:faa

The FAA regs (FAR 101) used for model/high power rocketry specifically refer to "unmanned rockets", not "unpiloted rockets", so it isn't at all clear how something like this is regulated.

I have sent a query to the FAA about getting waivers for the manned rockets my team is working on, but it got booted up to Washington a couple weeks ago, and I haven't heard back yet.

1.45 Games - Five Years of Quake - Jun 2001

1.45.1 Re:What about Marathon?

I won't get into gameplay arguments about it, but from an engine standpoint:

Pathways into Darkness was Bungie's take on Wolfenstein.

The original Marathon was Bungie's take on DOOM.

1.45.2 History, etc

I don't put a lot of stock in pinning down "firsts", even though people in general, and the media in particular, love to harp on it.

Everything is built on past work.

A lot of people like to think of creativity and innovation as something that springs from the void, but the truth is that everything is traceable to its origins.

I consider myself fortunate that I am consciously aware of the process. I can dissect all of my good ideas into their original parts, and even when there is an interesting synthesis, the transformation can usually be posed as an analogy to some previous work.

Given that fact, you will rarely find me touting anything as a "first", because I could always say it is "sort of like this thing over here, but with the principle demonstrated by this over there added to allow it to give the feature we wanted back then" and so on.

There are the occasional "eureka!" moments, but they tend to be in twitchy little technical things, not the larger ideas like "3D environment" or "multiplayer gaming".

I'm not all that concerned with our place in history. The process has been interesting enough in its own right, and lots of people have enjoyed the work as we produced it.

1.46 Slashback: Shooters - Jun 2001

1.46.1 Manned rocket ships and the X-Prize

We will have a manned rocket vehicle flying by the end of the year, but it will be a modest little thing. The performance will only be about what you got out of the old Bell rocket pack, but it is fully fly-by-wire (and can be tested remotely) with active stabilization, and all the subsystems are directly scaleable to much larger vehicles.

I will probably enter as an X-Prize competitor at that point.

http://www.armadilloaerospace.com/"title="armadilloaerospace.com"rel="nofollow [armadilloaerospace.com]

1.47 Science - YAPSLP: Yet Another Private Space Launch Plan - Jun 2001

1.47.1 Re:Liquid-fueld rockets are no child's plaything

He is sort of correct.

It is easier to get a high thrust to weight ratio out of solids, even though they have a lower Isp and generally a lower total dV per stage.

To get more thrust out of a solid, you basically just need a bigger nozzle, while to get more out of a conventional liquid engine, you need a bigger nozzle, and bigger turbopumps, plumbing, and combustion chamber.

Not that I think solids on a manned vehicle are very sensible...

1.48 Science - Japan Tests Reusable Rocket - Jul 2001

1.48.1 Re:carmack

Yes, this is very similar to what I am working on.

www.armadilloaerospace.com

Their current vehicle is a good deal larger than ours, has an aeroshell, and significantly, uses liquid hydrogen / liquid oxygen, a much more potent and difficult propellant than the hydrogen peroxide we use.

On the other hand, my project has been a lot faster and cheaper. I have spent about $50k and we have been working on it for nine months, versus their $400k and four years.

Our first up scale vehicle is going to be ready in a few months. It won't go very high or fast, but we can carry a person on it...

Next year, at the very least, we will have a supersonic manned rocket ship.

1.48.2 Re:Just use a parachute!

Run the numbers.

With a 400 Isp engine, one percent of the vehicle landing mass can kill 40m/s of velocity. That is over 80 mph, which an efficient VTVL vehicle could expect to have as its terminal velocity.

A parachute masses more than 1% of its load.

There are lots of other factors that can push the decision either way, but it is certainly within the realm of feasible engineering decision.

1.49 Another Space Tourist In Training - Jul 2001

1.49.1 Re:TANJ! It makes me angry!

Then we get NASA and the US Government refusing to allow private launches so that people have to go off-shore to launch to try to claim the X Prize!!!

Actually, it's worse than that.

The US government (NASA doesn't have anything to do with it) claims authority over all actions of its citizens, even when they aren't inside national boundaries.

If you launched an X-Prize vehicle from another country or from international waters without getting FAA/AST clearance, you are still in trouble (and wouldn't be eligable for the prize).

1.50 Games - ATI&Nvidia Duke It Out In New Gaming War - Aug 2001

1.50.1 Performance benefits

The standard lighting model in DOOM, with all features enabled, but no custom shaders, takes five passes on a GF1/2 or Radeon, either two or three passes on a GF3, and should be possible in a clear + single pass on ATI's new part.

It is still unclear how the total performance picture will look.

Lots of pixels are still rendered with no textures at all (stencil shadows), or only a single texture (blended effects), so the pass advantage will only show up on some subset of all the drawing.

If ATI doesn't do as good of a job with the memory interface, or doesn't get the clock rate up as high as NVidia, they will still lose.

The pixel operations are a step more flexible than Nvidia's current options, but it is still clearly not where things are going to be going soon in terms of generality.

Developers are just going to need to sweat out the diversity or go for a least common denominator for the next couple years.

I fully expect the next generation engine after the current DOOM engine will be targeted at the properly general purpose graphics processors that I have been pushing towards over the last several years.

Hardware vendors are sort of reticent to give up being able to "out feature" the opposition, but the arguments for the final flexibility steps are too strong to ignore.

1.51 Science - Canadian Team Plans Balloon-Aided X-Prize Entry - Aug 2001

1.51.1 Re:Question for Carmack

Taking things one step at a time is critical for success.

We have a pretty clear plan of attack to take us to X-Prize level vehicles. There will be several intermediate vehicles to learn from along the way, but I am pretty confident that we can do it, and that I can pay for it. The regulatory approval is still uncertain. Things get much more questionable after that.

The next step would be using the X-Prize vehicle as a booster for a upper stage(s) that launch a microsat into orbit. That requires many times for dV, and the regulatory environment, telemetry, and logistics become a lot more challenging. This would get fairly expensive, because making a reusable upper stage(s) is a whole new level of problem, and you just can't test a lot of the systems without going all the way. Even on a shoestring, it could easily get to $100k for each attempt, after you factor everything in. Realistically, it will take a lot of attempts to learn everything you need to know. A lot of people will talk about how straightforward it is, but I have

a healthy respect for the challenges. Smart money probably wouldn't bet on any "garage shop" getting to orbit, but it certainly isn't impossible.

After that, you could either work towards reusable upper stages, or scale everything up to the point you could try to orbit a passenger or a semi-useful LEO satellite.

Sure, if all that works out, I would love to make a moon shot, but that qualifies as day-dreaming, not planning. The idea that Dennis Wingo has floated recently about M class asteroids rich in platinum group metals possibly being able to have survived impact on the moon without vaporizing under some conditions is Very Very Interesting.

The extent of my "business planning" for the rockets is along the lines of "If you actually make something really, really cool, you will wind up making money on it somehow". Hopelessly naive? Possibly. We'll see. I hate being involved in business, so we would probably just partner with some of the existing companies interested in suborbital rides or sounding rocket business.

In the short term, watch for us getting a man off the ground in the upscaled lander frame within a couple months.

On topic: I think pretty highly of the DaVinci project, and I would say they are definitely one of the leading contenders. Brian Feeney talks about some technical issues on open mailing lists, which is a good sign. My biggest concern for them would be that, from my experience with JP Aerospace, getting two successful rockoon launches off within the 14 days required by the X-Prize is going to involve a good sized helping of luck.

1.52 Games - 3D First-Person Games, So Far - Aug 2001

1.52.1 Errors.

It (DOOM) was designed by talented people with good skills and academic degrees in computer science.

None of us had degrees in computer science. Romero, Adrian, and I don't have any degrees at all, and Kevin's is in political science.

It even had a simple but multithreaded "operating system" of its own to handle asynchronous updates of graphics and playing sound while performing the game simulation.

No. We made the startup sequence busy and techie in a sort of imitation of the NeXT workstations we were using at the time, but there was no multithreading going on. The sound was done with interrupt driven processing, which doesn't qualify.

With the source code open for years, this should have been easy to check.

a resolution of only 320x240

320x200

I would take issue with some of the other vague statements made later on, but they aren't pointed enough to debate.

1.52.2 Re:Doom expandability

We were surprised at Wolf3D mods, but we knew it was going to happen with DOOM. I worked with some of the Wolf3D map editor guys before DOOM was even released, but they didn't wind up making the popular level editors.

The editor and utility source code was released quite early, but it was all for NeXT workstations in Objective-C, so it had to wait for someone to

rewrite it for more conventional systems.

1.53 Ask - What is Happening with OpenGL? - Aug 2001

1.53.1 The present and the future

I'm still developing everything with OpenGL, and I'm still targeting mac and linux as well as windows, but I want to rationally address some points in the API debate:

D3D is clunky, etc

Not really true anymore. MS made large strides with each release, and DX8 can't be called a lousy API anymore. One can argue various points, but they are minor points. Anti-Microsoft forces have a bad habit of focusing on early problems, and not tracking the improvements that have been made in current versions. My rant of five years ago doesn't apply to the world of today.

I do think that the world would be a slightly better place if Microsoft had pulled an embrace-and-extend on OpenGL instead of developing a brand new API that had to be rewritten a half dozen times, but its water under the bridge now.

Open for more sales, etc

It has been pretty clearly demonstrated that the mac market is barely viable and the linux market is not viable for game developers to pursue. Linux ports will be done out of good will, not profit motives. From an economic standpoint, a developer is not making a bad call if they ignore the existence of all platforms but windows.

DX8 Gives more features

Some people have an odd view that an API gives you features. Assuming you don't care about software emulation, hardware gives you features,

and an API exposes them. If you try to use vertex programs or bump env map on a TNT, DX8 doesn't magically make it work. DX8's hardware independence is also looking a bit silly now as they make point releases to support ATI's new hardware. They might as well say D3D-GF3 or D3D-R200 instead of DX8 and DX8.1.

All of Nvidia's new features have showed up as OpenGL extensions before they show up as new D3D features.

Divergent extensions haven't been a problem up until very recently. All of the vendors tended to support all the extensions they could, and if they had unique functionality, like register combiners, they made their own extension. The current status of vertex programs does piss me off, though. I really wish ATI would have just adopted Nvidia's extension, even if it meant not exposing every last bit of their hardware.

Abstraction in a high performance environment can be dangerous. If you insist that all hardware behave the same, you prevent vendors from making significant improvements. If the spec for behavior comes from people that aren't hardware oriented, it can be a huge burden. D3D still suffers somewhat due to this, with some semantics and odd features that make hardware guys wince.

The Good News

We are rapidly approaching a real golden age for graphics programming. Currently, cards and API's are a complex mess of hundreds of states and function calls, but the next two years will see the addition of the final primitive functionality needed to allow arbitrarily complex operations with graceful performance degradation.

At that point, a higher level graphics API will finally make good sense. There is debate over exactly what it is going to look like, but the model will be like C. Just like any CPU can compile any C program (with various levels of efficiency), any graphics card past this point will be able to run any shader. Some hardware vendors are a bit concerned about this, because bullet point features that you have that the other guy doesn't are a major marketing feature, but the direction is a technical inevitability. They will just have to compete on price and performance. Oh, darn.

It's a Turing machine point. Even if OpenGL 2.0 and DX10 don't adopt the same shader description language, they will be functionally equivalent, and could be automatically translated.

There is lots of other goop like texture specification and context management that will still be different between API, but the core day-to-day work of a graphics programmer will be basically above the disputes.

1.53.2 Re:I don't think it's really been established...

I think it's only been established that Id didn't do well with the Linux gaming market

All linux games sales EVER don't add up to one medium selling windows title. We are one of the creditors that aren't likely to see money that Loki owes us, so we have some idea just how grim it is.

That isn't saying that it can't change in the future, or that doing linux ports isn't a Good Thing, but it isn't an economic motivator at the present time.

1.54 Science - Private Rocketplane Test A Success - Oct 2001

1.54.1 Re:Rocket Racing!

The rockets we are currently firing use hydrogen peroxide, which produces nothing but water and oxygen in the exhaust. Not even the most rabid greenie could argue with that.

Hydrogen / oxygen rockets also produce water and excess hydrogen. Alcohol / ocygen rockets leave a few other things similar to auto exhaust, but not really worse.

Solid rockets leave some bad stuff, and some propellants are truly nasty, like nitrogen tetroxide and hydrazine, but those are also much more ex-

pensive, so wouldn't be used in a cost effective program.

1.55 Yro - Safeweb Turns Off Free Service - Nov 2001

1.55.1 Re:CIA Investors

I have a very interesting tale about this.

One of the suppliers to Armadillo Aerospace told me about an experiment that he tried. He was looking over the logs to his (very low traffic) site, and he wondered how an anonymized hit would show up in the logs. He went through Safeweb, and saw a properly obscured address in the logs.

One hour later, he also got a hit to the same page from cia.gov.

I'm sure this isn't standard practice for every access, but his site was probably on a hot list of some sort due to the aerospace content.

1.56 Games - Quake 2 Source Code Released Under The GPL - Dec 2001

1.56.1 Thanks

Thanks for the kind comments, it helps me brace a bit for some of the really vile hate mail that is already starting to come in from the people worried about cheating.

Bill Heineman is preparing the mac source code for Q2 for a release.

We will see about getting the 3.21 changes we missed into an updated release.

I am also happy to say that another old game's code will be released under the GPL soon. We can always hope that it becomes a trend...

1.57 SGI Sets Sights On Turnaround - Dec 2001

1.57.1 Re:Bali and Odyssey... *sigh*

*Cutting edge graphics, where did you go? Please tell me there's more to the 3D world than IR, WildCat II, and GeForce3. Has *nothing* (other than cost) really changed over the past five years? It's almost as though I haven't missed anything in the 28 months I've been away from 3D*

Dependent texture reads are the only really new thing in the last year or two (and only really got worked out right in the Radeon 8500), but next year is going to see floating point pixel formats, which was going to be one of Bali's truly important points. We should also see highly scalable boards built on consumer chips, which has been promised for years, but (with the exception of some 3dfx high end systems) not delivered properly.

1.58 Games - The Rise And Fall of Ion Storm - Jan 2002

1.58.1 Comments

From the article:

But with great success came great antipathy, not just for John, but also for many of his employees.

The employees did sort of get a raw deal by association, but to ascribe all of the antipathy towards Romero to jealousy is really missing the point.

Daikatana and Deus Ex were finally released in 2000. Predictably, Daikatana

was slammed while Deus Ex received many awards. Both made money for Eidos

Deus Ex made money. Daikatana lost an immense amount of money. We followed the PC-Data sales numbers for a little while, and it was really, really grim. It might have made a comback when it went to the bargain bin, but even if it had turned into the best selling game of the year, it wouldn't have covered the sunk costs at Ion.

My view:

Ion storm failued due to lack of focus, which came from the top. They had some great employees (we hired some of them!), but games don't get done without someone in a position of authority forcing everything together. Romero's primary mistake was believing that abstract creative design was a primary, or even significant, part of a successful game. The "strategic creativity" in a game is less than 1% of the effort, and if you put that on a pedestal, you will deephasise where all the real work needs to be done.

I think Romero has a chance at a comeback with his current foray into handheld games. I don't think he ever lost the enthusiasm for games, but if he can recapture the personal work ethic that he had early on, he can probably still do some pretty cool things.

1.58.2 Re:Definitions of terms

To elaborate a bit:

Probably everyone reading this has done some "game design" while talking with friends. In an evening, you can lay out the basic character of a game – what the player does, what the environments are like, what the obstacles are, what the tools in the game are like, what the plot is, what the style of the game is, and a few unique hooks for the game.

There is not a hell of a lot of difference between what the best designer in the world produces, and what a quite a few reasonably clued in players would produce at this point. This is the "abstract creativity" aspect. This part just isn't all that valuable. Not worthless, but it isn't the thing to wrap

a company around.

The real value in design is the give and take during implementation and testing. It isn't the couple dozen decisions made at the start, it is the thousands of little decisions made as the product is being brought to life, and constantly modified as things evolve around it. If you took two game designs, one good and one bad, and gave them to two development teams, one good and one bad, the good dev team could make a good, fun product out of a bad design, but the bad dev team could ruin the most clever design. The focus should be on the development process, not the (initial) design.

The games with 500 page design documents before any implementation are also kidding themselves, because you can't make all the detail decisions without actually experiencing a lot of the interactions.

Putting creativity on a pedestal can also be an excuse for laziness. There is a lot of cultural belief that creativity comes from inspiration, and can't be rushed. Not true. Inspiration is just your subconscious putting things together, and that can be made into an active process with a little introspection.

Focused, hard work is the real key to success. Keep your eyes on the goal, and just keep taking the next step towards completing it. If you aren't sure which way to do something, do it both ways and see which works better.

1.58.3 Re:Definitions of terms

Clever design + bad dev team = Deer Hunter, so there is an argument to be made for both sides

That is a really good example. I might quibble that that was market creativity, rather than game design creativity, but it is still a good point.

1.59 Games - Carmack: Lord of the Games - Feb 2002

1.59.1 A few corrections

A few corrections to the article:

"My own graphics technology"

is OpenGL.

"Mr. Carmack also plays computer games in the office with his coworkers"

I played Q3 quite a bit, but not much since then. The team focus of TeamArena and Wolfenstein just isn't my favorite type of game.

"Polygon counts"

The Doom engine is not an ultra-high poly count engine, because it is built around dynamic lighting and shadowing, but it is still a large step up from our previous games. Typical scenes will have around 150,000 polygons, versus 10,000 for Q3. There will certainly be other games with higher raw polygon counts, but that is really focusing on the trees, not the forest (image quality). The large numbers that have occasionally been tossed around are the polygon counts for the high detail characters that are used in the generation of normal maps for the real time rendering. Some characters are over 500,000 polygons in their original form.

"It looks like the type of game that is so thrilling to play that gamers will do so over and over again, even though it lacks a narrative plot."

Unlike everything we have done before, the new Doom actually DOES have a real plot, and I think it is going to be presented well. I don't really expect most people to believe us at this point, but wait and see...

"The new Doom likely will require a no less powerful chip than the soon-to-be-released Nvidia GeForce3"

It is designed for full impact on a GeForce-3, but it still runs on a GeForce-1 or Radeon.

They didn't reproduce the graph of our revenues from the print version, but that was also way off base. I guess they estimated them based on our title sales, but while Doom II remains our best selling title, we have much better royalty arrangements now than we did back then, so we make more money today.

1.60 Science - Maverick Rocketeers Pursue Space Access - May 2002

1.60.1 I'll comment later...

I am helping a hardware vendor optimize the E3 build of Doom right now, but I'll make a pass of replys and comments later on tonight...

(yes, the Id net connection is slashdotted at the moment)

1.60.2 Re:It won't be cheap

I am going to have to save the parent post, because it is such a perfect example of the mindset that has made progress in aerospace so damn slow. I couldn't have said it better if I was trying to intentionally construct the stereotype. This ties directly in to the quote I had in the article – "rocket science" has been mythologized out of all proportion to its true difficulty.

First, you are severely understating the achievements of the Wright brothers. They had to invent almost everything from scratch, including much of the theory, and there was no existence proof to show that it was possible at all. I'm really not an aviation buff, so I'm sure someone else can recount the challenges better, but it is worth noting that at the time the Wrights did their work, there was also a high profile, government funded effort underway headed by Samuel Langley. With the "best minds in the country" and government resources behind it, they still didn't make the breakthroughs.

I contend that building and flying an X-Prize class vehicle (100 km suborbital, three passengers, reusable) today is a much less daunting task than the original invention of the airplane.

We have existence proofs of what is being attempted. There is no question that it is possible, because it has been demonstrated in many different forms. The only question is how cheap it can be done.

There is a massive amount of information available. Today, anyone can go read up on things that NOBODY knew back when they were building the early rocket systems.

Obviously, computers and electronics are vastly better. Our current electronics box has all the necessary sensors and actuators for flying a spaceship, and it cost less than $15,000 to put together (yes, it runs linux).

It isn't as blatant, but other manufacturing areas have also made great progress. I had a batch of a dozen small motors made at a CNC job shop for only $1000. Even counting everything that goes into them, the total cost including valve is less than $300 each. These may well be better than the peroxide thrusters used on the Mercury capsules. It was amusing to hear the NASA pad manager tell stories about having to go bang on the Mercury thrusters with a wrench to get them to stop sputtering. Don't think that all NASA hardware performs as designed.

Pressure vessels are significantly improved. A common all-carbon-fiber tank for natural gas vehicles has a better compressed volume to mass ratio than anything that could be built in the sixties. Filament winding can make large structures that are both stronger and cheaper than the classic welded structures.

There are direct spinoffs from government rocketry development. To drill the tiny, high aspect cooling passages for the Agena upper stage engine, they had to invent brand new machining technologies. Today, you can get the same techniques done at standard industrial job shops. As far as expensive materials go, the Agena engine was made out of aluminum.

The general industrial infrastructure is also a heck of a lot better. I can order damn near anything I need for our work from McMaster-Carr at 4:00 in the morning, and it shows up two days later.

NASA spent $50 million to set up the tracking and telemetry networks for Mercury. You can get far, far better results today with a GPS and satellite modem. There are billions of dollars of space based assets already at our disposal.

I could go on for quite a while on why we would have an easier time today just replicating the efforts of the past, but that is only part of the issue. What we are aiming for in the near term is far smaller in scope than any of the projects that the public normally associates with space. Even with all the advantages of today, it would be absurd to think that we could put together a space shuttle or a Saturn V. I hesitate to make analogies, but we are effectively working on building little microprocessors instead of big mainframes. 100 km straight up and down (that gives a 5G reentry, which, while not for your grandmother, doesn't take a superhero) is just not all that hard.

Yes, there are lots of challenges to be met, and we will doubtless run into all sorts of things that we haven't even considered. We will solve them as we go. People do hard things all the time, in many different fields. The reason "rocket science" looks so much harder is just lack of familiarity.

Because the existing way of doing things in space costs tens to hundreds of millions of dollars a shot, there just isn't an opportunity for radical experimentation. The optimization problem is slowly trending towards a stable local minimum, with little chance of getting out to the global minimum. Imagine trying to develop software if you only got to compile and run your app four times a year. Imagine how much that would slow down progress, and what contortions you would go through if $100 million was riding on each run.

Build fast. Test often. Stay flexible. Mind the critical path.

1.60.3 Re:I'll comment later...

Are you now using an inertial guidance system or is there a better alternative? I assume that GPS does not provide enough accuracy for low speed guidance

We are currently using a Crossbow inertial unit with fiber optic gyros for the fast attitude stabilization.

We have flown GPS on a couple flights, but the update rate is too slow for active control. I do feel that in the longer term, carrier wave interferometry GPS sensors will offer the most cost effective attitude sensors, but right now they are $15,000+ system. If I was doing this on a much tighter budget, I would consider trying to build a fast updating CW GPS system from available cheap GPS cores, but that is a project of significant complexity all by itself.

I have integrated a new laser altimeter with the electronics box, but we haven't flown it yet. I am looking forward to this, because it will allow us to begin working on auto-hover and auto-land control software.

1.60.4 Re:RocketGuy!

He came to Space Access to meet with us, and it was interesting talking with him. He is certainly not an engineer, but he is actually building a lot of hardware, which is more than can be said for most folks in the space crowd.

The abject stupidities in his original design that got him a lot of flack (Fins at the top! 1.2 T/W ratio without guidance!) are now gone, and he has decided to have a testing plan before launching himself, so I think he has a decent shot at flying something and living to talk about it. I wish him luck.

An interesting question: is it easier to motivate a learned individual that never does anything, or educate an ignorant individual that actually produces things?

1.60.5 Re:Straight from the horse's mouth

Anyone know if any of these milestones were achieved? Or if not, what armadillo's latest estimates are for the same things?

The estimate from day one was:

Year one: VTVL demonstratorYear two: manned rocket shipsYear three: space shots

The VTVL demonstrator went faster than expected, and it looked like we were going to lift a person off the ground before the end of the first year. We had a couple crashes and redesigns that set us back a bit, and we were forced to make a major change in our catalyst packs to allow us to get enough back-to-back flights without changes before putting a person on it, so we haven't yet made that "milestone bunny hop".

However, while we were waiting for some things along that development path, we wound up developing some other technologies that weren't even in the original plan – our recent work on biprop engines wasn't really scheduled until year three or later, and the rocket rotor work is looking like it will allow some big improvements in our upcoming designs.

The current goal of record is to set some of the manned aviation 3000 meter time-to-climb records before the end of this year.

1.61 Games - Carmack on Doom 3 Video Cards - May 2002

1.61.1 Misrepresented.

This batch of comments from me have let people draw conclusions that leave me scratching me head wondering how they managed to get from what I said to what they heard.

Other people have outlined the issues in detail in comments already, but the crux is that, even with driver quality removed from the discussion (not counting conformance issues, running at fill limited resolutions), GF4 hardware is still faster than 8500 hardware on basically everything I tested. The 8500 SHOULD have been faster on paper, but isn't in real life.

The hardware we used at E3 was not an 8500, and while the drivers were

still a bit raw, the performance was very good indeed.

Take with a grain of salt any comment from me that has been paraphrased, but if it is an actual in-context quote from email, I try very hard to be precise in my statements. Read carefully.

1.61.2 High end hardware reasoning

We know for sure that we will be excluding some of the game buying public with fairly stiff hardware requirements, but we still think it is the right thing to do.

The requirement for GF1/Radeon 7500 as an absolute minimum is fundamental to the way the technology works, and was non-negotiable for the advances that I wanted to make. At the very beginning of development, I worked a bit on elaborate schemes to try and get some level of compatibility with Voodoo / TNT / Rage128 class hardware, but it would have looked like crap, and I decided it wasn't worth it.

The comfortable minimum performance level on this class of hardware is determined by what the artists and level designers produce. It would be possible to carefully craft a DOOM engine game that ran at good speed on an original SDR GF1, but it would cramp the artistic freedom of the designers a lot as they worried more about performance than aesthetics and gameplay.

Our "full impact" platform from the beginning has been targeted at GF3/Xbox level hardware. Slower hardware can disable features, and faster hardware gets higher frame rates and rendering quality. Even at this target, designers need to be more cognizant of performance than they were with Q3, and we expect some licensee to take an even more aggressive performance stance for games shipping in following years.

Games using the new engine will be on shelves FIVEYEARS (or more) after the initial design decisions were made. We had a couple licensees make two generations of products with the Q3 engine, and we expect that to hold true for DOOM as well. The hardware-only decision for Q3 was controversial at the time, but I feel it clearly turned out to be correct. I

am confident the target for DOOM will also be seen as correct once there is a little perspective on it.

Unrelated linux note: yes, there will almost certainly be a linux binary for the game. It will probably only work on the nvidia drivers initially, but I will assist any project attempting to get the necessary driver support on on other cards.

1.62 Ask - Improv Animation as an Art Form? - Jun 2002

1.62.1 Realtime and offline rendering ARE converging

There are some colorful comments here about how studios will never-ever-ever replace tools like renderman on render farms with hardware accelerated rendering. These comments are wrong.

The current generation of cards do not have the necessary flexibility, but cards released before the end of the year will be able to do floating point calculations, which is the last gating factor. Peercy's (IMHO seminal) paper showed that given dependent texture reads and floating point pixels, you can implement renderman shaders on real time rendering hardware by decomposing it into lots of passes. It may take hundreds of rendering passes in some cases, meaning that it won't be real time, but it can be done, and will be vastly faster than doing it all in software. It doesn't get you absolutely every last picky detail, but most users will take a couple orders of magnitude improvement in price performance and cycle time over getting to specify, say, the exact filter kernel jitter points.

There will always be some market for the finest possible rendering, using ray tracing, global illumination, etc in a software renderer. This is analogous to the remaining market for vector supercomputers. For some applications, it is still the right thing if you can afford it. The bulk of the frames will migrate to the cheaper platforms.

Note that this doesn't mean that technical directors at the film studios

will have to learn a new language – there will be translators that will go from existing langauges. Instead of sending their RIB code to the renderfarm, you will send it to a program that decomposes it for hardware acceleration. They will return image files just like everyone is used to.

Multi chip and multi card solutions are also coming, meaning that you will be able to fit more frame rendering power in a single tower case than Pixar's entire rendering farm. Next year.

I had originally estimated that it would take a few years for the tools to mature to the point that they would actually be used in production work, but some companies have done some very smart things, and I expect that production frames will be rendered on PC graphics cards before the end of next year. It will be for TV first, but it will show up in film eventually.

1.63 Interviews - Ask 'Rocket Guy' Brian Walker - Jul 2002

1.63.1 Re:Exotic Fuels

AFAIK (we met at Space Access this year), Brian is not interested in advanced engine work. For his goals, simple monoprop peroxide is far and away the most direct route.

For reference, while the theoretical Isp is usually listed around 155, we typically only see 115 or so at sea level with less than 300 psi chamber pressure.

hybrid (ie, plastic/nitrous oxide) propellants?

Peroxide makes a pretty good hybrid oxidizer, with better Isp and density-Isp than nitrous based hybrids, plus it auto-ignites after decomposition. Vec Isp may be as high as 275 with 90% peroxide, but sea level will be down around 200-225, depending on chamber pressures. We fired a couple peroxide / polyethylene hybrid grains last year, but we haven't pursued it much.

There is a very tantalizing possibility of using aluminum hydride as a hybrid graid with peroxide, giving a theoretical vacuum Isp of over 400 (!!!), and it is non-toxic. We are probably going to look into this one of these days.

buckyballs

Not much use. Buckytube composites may make for very mass efficient tanks and structures in the not too distant future.

multi-atomic nitrogen

If it can ever be produced affordably, a 600 Isp monoprop would sure be nice. Easy to go boom, though.

fluorine

Ick. Very toxic, very corrosive. Flourine / lithium hybrids can get over 500 Isp, but it would be very dangerous.

I feel that the best way to take advantage of exotic developments is to build a fully functional vehicle with conventional materials, so if a wonder material / propellent does materialize, you are well poised to take advantage of it.

1.64 Games - The Technology Behind ID's Games - Aug 2002

1.64.1 Re:Softdisk

Here is an interesting bit of history:

Greed was built on the engine I wrote for Raven/Origin's Shadowcaster game, while the other Id guys were working on Spear of Destiny, the commercial Wolfenstein game.

The reason softdisk got the technology was that they were still making lots of noise about suing us for doing Keen while we were working at soft-

disk. Our original parting deal was that we were going to continue doing the Gamer's Edge games for a while, basically for free, as penance. We weren't savvy enough to get anything binding down on paper, so even when it was all wrapped up, there was room for twisting our arm a bit. (another trivia bit – George Broussard at Apogee arranged to have Apogee produce one of the Gamer's Edge titles for us, so we could focus more on Wolfenstein).

We finally arranged a technology transfer of the latest engine code for free and clear severing of our ties. After they showed that just having the technology was not a guarantee of success, they had the nerve to come back and ask for more, but by then we were able to just tell them to go away.

BTW, Duke Nukem does not have a Softdisk heritage, it was by Todd Replogle (sp?), who was strictly Apogee-grown.

1.64.2 Originality, creativity, etc

I was fairly pleased with how that article turned out - when I first heard about it, I dreaded seeing a trivialized simplification of the issues, but it turned out as representative as you can be in that space.

However, I really dislike discussions of the attribution of techniques to a particular programmer. Everything is derived from things before it, and I make no claims of originality. I would say that one of my talents is the ability to be aware of what sources are feeding into my work, and be able to backtrack to them. Also, there are always lots of other possible answers for any given problem that can be made to work. BSP vs sector list, Portals vs PVS vs scan line occlusion, tilted constant Z rasterization vs block subdivision vs background divides, etc. Looked at in the proper perspective, individual techniques just aren't all that important. Sometimes it sounds like "Dude, he INVENTED needle nose pliers!!!"

Heck, I somewhat deride the very concept of originality. Creativity is just synthesis without the introspection. Lots of people will catch on that and start a rant about how Id games aren't original, but they are missing the point - it is possible to set out and develop something that will be

received as ”original” without ever having an ”original” idea spring into your mind.

The best way to get answers is to just keep working the problem, recognizing when you are stalled, and directing the search pattern. Many of the popular notions of innovation and creativity are in some ways cop-outs that keep people from being as effective as they could be. The little document I wrote about developing a part of the shadow algorithm for Doom that Nvidia has on their website was a pretty good example of my process. Don’t just wait for The Right Thing to strike you - try everything you think might even be in the right direction, so you can collect clues about the nature of the problem.

1.65 Games - Carmack Expounds on Doom III - Aug 2002

1.65.1 Re:Rendering - two generations from done?

My comment specifically regards the ”shelf life” of a rendering engine. I think that an upcoming game engine, either the next one or the one after that, will have a notably longer usable life for content creation than we have seen so far. Instead of having to learn new paradigms for content creation every couple years, designers will be able to continue working with common tools that evolve in a compatible way. Renderman is the obvious example – lots of things have improved and evolved, but its fundamental definition is clearly the same that it was over a decade ago.

This is only loosly related to the realism of the graphics. I don’t think a detailed world simulation that is indistinquishable from reality will be here in the next decade, except for tightly controlled environments. You will be able to have real-time flythroughs that can qualify as indistinguishable, but given the ability to ”test reality” interactively, we have a lot farther to go with simulation than with rendering.

1.65.2 Re:"Non-high-end-comp-owning geeks rejoice!"?

The X-Box GPU is more of a GF4 than a GF3, but a modern PC is generally much higher end than an X-Box.

However, you can usually count on getting twice the performance out of an absolutely fixed platform if you put a little work into it. There are lots of tradeoffs that need to balance between the different cards on a general purpose platform – things that I don't do with vertex programs because it would make the older cards even slower, avoiding special casing that would be too difficult to test across all platforms (and driver revs), and double buffering of vertex data to abstract across VAR and vertex objects, for instance. We might cut the "core tick" of Doom from 60hz to 30hz on X-Box if we need the extra performance, because it has no chance of holding 60hz, but the PC version will eventually scale to that with the faster CPUs and graphics cards.

1.65.3 Re:different backends useless then?

The generic back end does not use vertex programs, or provide specular highlights, so the custom back ends provide both performance and quality improvements.

There are some borderline cases that may or may not get custom coding – Radeon R100, Matrox Parhelia, and SiS Xabre are all currently using the default path, but could benefit from additional custom coding. I will only consider that when they have absolutely rock solid quality on the default path, and if it looks like they have enough performance headroom to bother with the specular passes.

The NV20 back end has more work in it than any other, with two different cases for the lighting interaction, but on the X-Box I would probably create additional special cases to optimize some of the other possible permutations.

1.66 Science - Amateur Rocket Launch a Failure; NASA Debuts Shuttle-cam - Sep 2002

1.66.1 Re:amateur rocketetry is irresponsible

NASA has launched more missions than anybody else

NASA has launched more manned missions than anybody else, but the Russians have launched nearly TEN TIMES as many space mission.

This is when someone adds "Yeah they had to, because their electronics suck, so they need to replace their sats more often", but that doesn't change the point about launches.

1.66.2 Re:Fuel and funds?

but similar attempts in the past have cost about $20,000

Ky has said it cost over $250,000. Just building the rocket and motor probably cost $20k, but everything else adds up.

1.66.3 Re:Fuel and funds? (correction)

The number I saw quoted was $150k, not $250k.

1.67 Games - Doom 3 Alpha Leaked - Nov 2002

1.67.1 Damnit.

No, this was not leaked on purpose.

Yes, we are upset about it, and it will have some impact on how we deal with some companies in the future, but nothing drastic is going to change

in terms of what support is going to be available.

Making any judgements from a snapshot intended for a non-interactive demo is ill advised.

1.68 Developers - Carmack on NV30 vs R300 - Jan 2003

1.68.1 Re:Once again...

But he mentioned something about next gen cards having less bandwidth. Does that make sense to anyone?

The RATIO of bandwidth to calculation speed is going to decrease. It is nothing short of miraculous that ram bandwidth has made the progress is has, but adding gates is cheaper than adding more pins or increasing the clock on external lines.

Bandwidth will continue to increase, but calculation will likely get faster at an even better pace. If all calculations were still done in 8 bit, we would clearly be there with this generation, but bumping to 24/32 bit calculations while keeping the textures and framebuffer at 8 bit put the pressure on the calculations.

1.69 Science - Carmack Needs Rocket Fuel - Feb 2003

1.69.1 Not exactly general interest news, but...

I agree with some comments that this isn't exactly general interest news.

I am not interested in hearing from every chem major that is interested in starting a business (already heard from a couple, that's how I found out

about the slashdot story). However, if anyone here does happen to have a brother-in-law that is a VP at FMC or some such, a little nudge wouldn't hurt.

The full story:

Rocket grade peroxide is stabilizer free, and 85% - 100% concentrated, as opposed to drug store peroxide at about 3% concentration. You can get up to 70% peroxide reasonably easily, but the high concentration stuff is a specialty item.

When we started our development work a bit over two years ago, we were doing some concentration of the peroxide ourselves, which is fine for making small test batches, but you really don't want to be making drums of the stuff, or you wind up spending as much time messing with that as you do building rockets.

We had some initial discussions with FMC about that time, but they weren't terribly encouraging. Shortly thereafter, we made contact with X-L Space Systems, a small company that was producing 98% concentration peroxide and selling it reasonably to several small outfits, as well as NASA and the USAF. I wound up buying a dozen or so drums from X-L, and everything was going well.

The owner of X-L was having such a hard time getting the government to pay their bills on time (he never had complaints about his small commercial customers) that he finally decided it just wasn't worth the headache, and he closed the company down. I was in discussion with him to make a large enough order to justify keeping production open, but we wouldn't need all that much peroxide for nearly eight months, so the storage logistics were looking troublesome. In hindsight, I should have worked something out, even if it was expensive or difficult.

About six months ago, we started contacting FMC again. The details haven't been very pleasant, largely because we keep thinking we are almost there, and it keeps not being the case. If they would just tell me exactly what I have to buy to make them happy, I would gladly do it, but they keep finding new things. That is the "stringing us along" part. They are mumbling again about lawyers and liability at the moment, which we thought had been worked through previously.

We have also spoken to Degussa about production, but they won't sell in drums, only large storage tanks (they supposedly have some drums in the US, but they are "promised to" NASA, and they won't sell them to us). We could live with that, but we broke off contact with them a while ago because FMC was sounding reasonable, but insisting that they be our sole supplier.

This is one of the unfortunate tradeoffs in modern society – in the 70's, FMC would just ship drums of peroxide to the guys doing rocket powered dragsters without any hassles (one of them sent me a scan of some of his old shipping invoices). Today, fears of liability are larger than basic business drives like making money with your product. I'm not a "back in the good old days" sort, I fully recognize that the other advantages of modern society outweigh the nanny-state disadvantages, but one can always hope for across-the-board improvements.

Other than being almost out of peroxide, things are going very well for Armadillo. We rescheduled a lot of our development now that the X-Prize is fully funded, so we are parallel tracking full scale vehicle development with subscale flight testing.

1.69.2 Re:Carmack is fragbait.

There is an interesting annecdote related to this.

At the world space congress last year, I was talking to Buzz Aldrin's son, who is head of acquisitions at Boeing. He really didn't believe that cheap, reusable launchers were possible (he thinks "billions of dollars in development"), but he said that if we win the X-Prize, demonstrating cheaper launch for even suborbital lobs, Boeing would "just buy us".

From our short discussion, it was clear that we have quite different world views, so I hesitate to read much into his statements one way or the other, but it was a bit curious.

1.70 Science - The Rutan SpaceShipOne Revealed - Apr 2003

1.70.1 Re:Burt Rutan vs. John Carmack?

We have obviously been eagerly waiting for this unveiling. Nobody has denied that Rutan is the odds-on favorite for the X-Prize, but I take a positive thing away from this unveiling – I have always contended that being an ”airplane guy” is going to hurt Rutan in the X-Prize, and this is definitely a ”winged thing”. I would have been more concerned if it was just a purely ballistic capsule being air launched. I have little doubt that they will fairly rapidly have successful zoom climbs to somewhat above 100,000', but it is far from the simplest design to go to 350,000'. It is certainly true that complex designs can be made to work with enough talent, experience, testing, and money, which Rutan has all of, but there is plenty of room for things to screw up.

I don't expect that they will make any flights to 100km this year, but I can certainly be proven wrong...

I am quite happy with our current design, and we are committed to following through irrespective of what Rutan does. Even if he makes it, we have a different ecological niche in terms of vehicle capabilities – our entire launch infrastructure can be towed by a light truck, and launched from anywhere. If he does win the X-Prize before us, we will ditch the monopropellant propulsion system and move to something more cost effective (at the expense of more development time) for the long term. We may be forced to do that anyway, if our peroxide situation doesn't resolve itself.

1.71 New Trailer for The Hulk - Apr 2003

1.71.1 Re:Physics?!?

To leap 50' in the air, you must be going 56.6 ft/s when leaving the ground, disregarding air resistance. Apogee will be in 1.77 seconds.

Assuming a linear acceleration, and a four foot period of acceleration from crouching to leaving the ground with legs extended, the average speed must be 28.3 ft/s over the four feet, for 0.14 seconds of acceleration, or 404 ft/s^2. 12.6 G's of acceleration isn't at all unreasonable for arm / leg contraction at light loads. You can make a $>$ 50G acceleration with a pitching motion of your arm.

12.6 G's of acceleration for an 800 pound hulk is only 10080 pounds, divided by two 24" long by 8" wide feet give a mere 26.25 psi force on the pavement.

If I botched these calculations, everyone is surely going to take the opportunity to say how the Armadillo vehicles will crash and burn... :-)

1.72 Hardware - Futuremark Replies to Nvidia's Claims - May 2003

1.72.1 Driver strategies

Rewriting shaders behind an application's back in a way that changes the output under non-controlled circumstances is absolutely, positively wrong and indefensible.

Rewriting a shader so that it does exactly the same thing, but in a more efficient way, is generally acceptable compiler optimization, but there is a range of defensibility from completely generic instruction scheduling that helps almost everyone, to exact shader comparisons that only help one specific application. Full shader comparisons are morally grungy,

but not deeply evil.

The significant issue that clouds current ATI / Nvidia comparisons is fragment shader precision. Nvidia can work at 12 bit integer, 16 bit float, and 32 bit float. ATI works only at 24 bit float. There isn't actually a mode where they can be exactly compared. DX9 and ARB_fragment_program assume 32 bit float operation, and ATI just converts everything to 24 bit. For just about any given set of operations, the Nvidia card operating at 16 bit float will be faster than the ATI, while the Nvidia operating at 32 bit float will be slower. When DOOM runs the NV30 specific fragment shader, it is faster than the ATI, while if they both run the ARB2 shader, the ATI is faster.

When the output goes to a normal 32 bit framebuffer, as all current tests do, it is possible for Nvidia to analyze data flow from textures, constants, and attributes, and change many 32 bit operations to 16 or even 12 bit operations with absolutely no loss of quality or functionality. This is completely acceptable, and will benefit all applications, but will almost certainly induce hard to find bugs in the shader compiler. You can really go overboard with this – if you wanted every last possible precision savings, you would need to examine texture dimensions and track vertex buffer data ranges for each shader binding. That would be a really poor architectural decision, but benchmark pressure pushes vendors to such lengths if they avoid outright cheating. If really aggressive compiler optimizations are implemented, I hope they include a hint or pragma for "debug mode" that skips all the optimizations.

1.73 Science - Solar Powered Helios Plane Destroyed in Test Flight - Jun 2003

1.73.1 Re:A thought or two...

IAARS. (I Am A Rocket Scientist.) ... The Shuttle uses LOX and LH2, both of which are f'nasty to deal with and are economical only to generate the immense thrust necessary to achieve orbit. While in orbit, the Orbiter ma-

neuvers using (relatively) small hydrazine thrusters. N2H4 is also f'nasty, but somewhat less so than either LOX or LH2.

???

The OMS uses hydrazine / nitrogen tetroxide, which is way, WAY more nasty than LOX / LH2.

LOX / LH2 are cryogens, and contact with them will give you frostbite. Hydrazine is carcinogenic and toxic, but nitrogen tetroxide is roughly as poisonous as the best war gasses from WWI. Plus, it has very low surface tension, so when it spills, it spreads extremely rapidly, which causes it to vaporize even faster than the already high vapor pressure would indicate. The various oxides of nitrogen are famous for the "BFRC" (big red cloud) that results from spills, which you should run away from very fast.

1.74 On-line Documentary on Machinima - Jul 2003

1.74.1 In The Wating Line

Arguably the most professional and widely viewed machinima so far is the music video for Zero 7's "In the Waiting Line", produced by my wife's company, http://www.fountainheadent.com/"title="fountainheadent.com [fountainheadent.com]. This was a real, commercial production using machinima tools.

It was neat to see the Q3 engine playing on MTV, but it made me greatly regret the quantized normals in Q3 models, which resulted in a noticeable popping on the environment maps. This was largely my motivation for adding per-pixel environment map calculation to the new Doom engine (under the ARB2 path, at least).

1.75 Science - 11-Pound Model Plane Vs. The Atlantic, Again - Aug 2003

1.75.1 Re:No need for GPS

Based on the first two responses to this post, you'd think people had never heard of inertial navigation. With MEMS accelerometers it ought to be pretty light, too.

Pure 3 axis inertial navigation with a strapdown inertial measuring requires extreme precision. MEMS inertial units aren't even in the right ballpark. Mechanical stable platform inertial systems that actually rotated inside the vehicles didn't require awesomely accurate sensors, but they are big, heavy, and not as reliable.

It is a useful programming exercise to write a simulation of a strapdown inertial system and play with bias, noise, and nonlinearity errors (add cross axis coupling and acceleration effects for micromachined gyros for bonus points). Pick reasonable ranges and quantize to 12 bits, then integrate at 100 hz or so. You can start the simulation motionless, but in a minute it will be cruising along at 60 mph in some random direction, hundreds of feet from the start position. An hour later, it will be heading for Mars.

The low end inertial systems that have been moderately soccessful are done by removing gravity from the equation and just doing 2D navigation, and often using other sensors, like magnetometers instead of rate gyros for heading, or odometer readings instead of double integrating accelerometers. Double integration of interrelated noisy sensors with an implicit 1G acceleration is really more demanding than it would initially seem.

The only reason you wouldn't want to use GPS in an ocean crossing is if you are afraid a Bad Guy might be jamming the signals.

1.76 Science - European Shuttle Program Update - Aug 2003

1.76.1 Re:stop making space planes, dammit

Inconel, the best commonly used alloy has a single use temperature limit of about 1030K, or 757C. It melts at / 1400C

The refractory metals are better, but less commonly used. Columbium/niobium is reasonable to form. Molybdenum and alloys like TZM take a bit more heat, but have a potentially annoing ductile to brittle transition point for systems that will cold soak. The state of the art is irridium coated rhenium, which doesn't melt until 2466 C / 4471 F.

We fabricated a TZM chamber a while ago at fairly high expense, but still burned through it after an extended length run:

http://media.armadilloaerospace.com/2002_08_31/burnedTZM.mpg"title="armadilloaerospace.com [armadilloaerospace.com]

This experience has convinced me that active cooling methods, like transpiration cooling, are probably a good idea for high reusability reentry vehicles.

1.77 Science - X Prize and John Carmack - Aug 2003

1.77.1 Re:Two Words

You probably mean "Burt Rutan", the aircraft designer at Scaled. Dick Rutan is his brother, who piloted the voyager, and was the test pilot for XCOR's EZ-Rocket, but doesn't have anything to do with Space Ship One, the X-Prize vehicle.

I have always maintained that Burt is the odds-on favorite to win the X-

Prize, but it isn't over yet. His design requires a pilot on board for all tests, so there is a non-negligable chance that there could be a fatality, which would almost certainly end the effort in the X-Prize timeframe.

1.77.2 Re:Cost

Just building the vehicle costs less than $100k, most of the money is in building multiple iterations of everything as you figure out exactly how you actually need to spend the money:

$ 6k 850 gallon fiberglass tank

$ 2k High pressure carbon fiber pressurant tank and regulator

$ 1k Honeycomb composite panels

$ 5k Aluminum fabrication for cabin

$15k Redundant parachutes, drogues, drogue cannons, releases

$13k Fiber optic gyro based IMU

$ 8k Unrestricted (supersonic / high altitude) GPS

$ 2k PC104 systems

$ 5k video, audio, and data communications

$20k Engine machining, catalysts, laser cut plates

$ 5k Plumbing, valves, etc

$ 5k Fastblock external insulation

For powered landings instead of parachute landings, delete the parachutes and add:

$ 4k Laser altimeter

$ 4k Wire rope isolator landing gear

You could trivially spend an order of magnitude more by just using "space certified" versions of everything, but the important point is that standard

industrial versions of many things are perfectly adequate. In many cases, todays standard industrial practice is far ahead of the best that could be done at any price in the early sixties.

This is all with free labor for assembly and testing, but that is still only a couple hundred man hours for a full vehicle. We are expecting to destroy the first vehicle in some (unmanned) testing mishap along the way, and build another one mostly from scratch. That will take less than two months, depending on lead times for some items.

1.78 Yro - Slashback: Diebold, Peroxide, Comdex - Oct 2003

1.78.1 Peroxide rocket propellants

High concentration hydrogen peroxide all by itself makes a low performance, but very convenient, rocket propellant. All hydrogen peroxide is in solution with some amount of water, because even if you had 100% peroxide, some of it would start decomposing to water (and oxygen) as you stored it.

Drugstore peroxide is 3% concentration. If you pour it on a catalyst, like silver or platinum, you will see bubbles forming in the solution (released oxygen), and the liquid will get somewhat warmer due to the released energy. Above roughly 70% concentration, the heat released is enough to vaporize all the water content, so if you pass it through a good catalyst, you will get all gas coming out the other side, and gas can be accelerated through a rocket nozzle to produce thrust. At 70%, the gas is only just above the boiling point of water, but as the concentration goes up, the temperature goes up fast. 90% peroxide, the most common grade used for propulsion, produces gas at about 1400 F temperature. Going all the way to 98% peroxide, the highest concentration produced, gives a few hundred degrees more temperature, but at a significant price increase. Higher temperature lets you use less propellant for a given amount of thrust-time, because it maintains a given chamber pressure with a less

dense, but hotter, mixture (a simplification).

"Real" rocket propellants have temperature several thousand degrees higher, which does indeed increase performance, but the engines have to be cooled, and you need to manage both a fuel and an oxidizer in some form. One of our fundamental system trades is that it is better for an X-Prize class vehicle to use a propellant that simplifies vehicle engineering, even if you have to use more of it.

We use 90% peroxide from a small specialty supplier for all of our flight vehicles, but they closed shop a while ago, and we haven't been able to come to terms with the only domestic supplier of 90% peroxide, FMC chemical corp. Because of this, we have been working on alternate propellant schemes for a good part of this year, in parallel with building the full size X-Prize vehicle. If we had been able to just buy 90% peroxide like we buy all of our other industrial chemicals, we never would have bothered with the research.

Just about every week, someone asks why we don't concentrate it ourselves. True, dozens of people have made a few gallons of high concentration peroxide at various times, but there have only been two large scale concentrators operated in the US outside of the official manufacturers - Rotary Rocket had a concentrator, but it only went to 85% concentration, and it didn't do purification, and Beal Aerospace had a large scale concentrator operational after the blew up their first one. Sure, we could figure out how to do it, but then we would be in the chemical plant business instead of the rocket business, and that's not what we want to do. I am funding an operator in Houston to produce a few thousand pounds of 90% for us, but he is six months behind schedule on delivery, which proves my point about it not being as simple as people think.

The direction we have been pursuing is using a combination of 50% peroxide, which is readily available through distributors from multiple manufacturers, and a small amount of miscible fuel (methanol in our current work). 50% peroxide by itself doesn't work as a rocket propellant, because you can't boil all the water, which makes even decomposing most of the peroxide difficult. Adding a fuel and (the tricky part!) getting it to burn with the released oxygen gives you the energy necessary to vaporize the water and get everything up to a high temperature. Mixing fuels with

high concentration oxidizers usually makes a touchy and deadly explosive (we have intentionally detonated a mix of 90% peroxide and alcohol - Very Scary), but buffered with 50% water, and running off of stoichemetric mixture ratio, the risk is not very high. We have a study report from the Department of Mines in the late 50's investigating these trinary mixtures, and our range could not be made to detonate after five blasting cap trials, which suits us for safety. In contrast, the nitromethane used in dragsters can detonate with a hammer blow, so we don't think we are being absurdly risky.

This combination gives performance slightly better than 90% peroxide, probably about equal to 98% peroxide when the inferior density is accounted for. It is less corrosive than 90% peroxide, can't have a thermal runaway, and is several times cheaper.

We are still a little concerned about the formation of organic peroxide compounds if the propellant was left stored for long periods, so we mix right before use, and always burn off any leftovers.

We aren't "done" yet, but it was an important breakthrough for us.

1.79 Science - X-Prize Progress Update - Dec 2003

1.79.1 We aren't being held up by regulatory issues.

We have a good working relationship with AST, the division of the FAA that handles launch license, and we are one of only three companies (along with Scaled and XCOR) currently in the RLV launch license process. We have found all the people there helpful and eager to work with us. There is a lot of paperwork to be done, but we are working through it, and do not see a problem satisfying them. Things like calculating and minimizing expected third party casualty rates are obviously necessary and sensible.

The environmental aspects are less rational, with no analytical sense of scale.

Still, I'm only mildly concerned about the regulatory side of things. I think it will work out. None of our work is held up by any of this, so the worst case is that we have a vehicle built and tested repeatedly at the 200,000 lb-sec waivered impulse limit, with no launch license to allow us to fill the tank the rest of the way up. If that happens, THEN we get peeved about the situation, but continue flight testing with what we can.

Let me repeat: In no way have we been hampered by regulatory burden. Yet. We have been VERY hampered by commercial companies being too worried about liability exposure to work with us - peroxide companies, filament winders, and parachute companies have all caused us significant problems.

The supply issue with 90% peroxide basically cost us almost the entire year of flight testing. We spent the last six months developing a propellant combination that could conveniently replace the 90% peroxide based on widely available chemicals instead of the ultra-specialized propulsion grade. We are in the final optimizing and scale up phase of that. Instead of being irate about it, I try to look on the bright side - it is lots cheaper, easer to handle, and even a bit higher performance.

There are lots of problems still to be worked, but everything is coming along fine. We are behind schedule and somewhat over budget, but no worse off than any other project I have ever worked on...

1.80 Apple - Steve Jobs' Grand Vision - Feb 2004

1.80.1 Re:Why?

Someone that has done "some good things (NeXT, the first iMac, OS X)" in their career gets my respect.

Most of the negative tales about Jobs probably have some grounding in truth – it was almost amusing watching him berate the stage people before a show for glitches in the prop moving systems: "What the hell is this??? Did you guys pick up these parts at Home Depot???". However, he did always listen when I was talking about a technical issue, even when

I was saying something that didn't sit with his current understanding of graphics cards / APIs / gaming.

When I was considering setting up to demo Doom 3 at macworld, all of the Apple people were going on about how we needed to sanitize it because "Steve won't let there be any blood or killing". I finally went to him directly, and he replied "If you think you can make it great, then let's do it. I trust you, so you'll have to decide." Not quite the overbearing micromanager he is sometimes portrayed as.

I'm not a regular mac user, but I'm glad Steve Jobs is still around.

1.81 Science - John Carmack's Test Liftoff a Success - Jun 2004

1.81.1 The full scale vehicle is also flying, sort of

For those of you that are underwhelmed by the 310 pound vehicle, do note that the big vehicle (1500 lbs) that can actually carry people is also flying. Look back in the Armadillo updates around April 19 for testing video. We have since reworked the propulsion system to follow what has worked so well on the subscale vehicle, and should be testing it this weekend. If it works well, we will be repeating the boosted hop with the big vehicle next week.

The flight time is currently limited by federal law to 15 seconds of rocket burn time. We have a waiver coming to extend that to 120 seconds, but beyond that we will need a full launch license.

The significance of all this is that the vehicles are intended to fly up, come back down and land right where they took off from, all without ablating, expending, or seperating anything. It should be possible to have turn around times under one hour even for quite large vehicles.

BTW, Doom beta testing is going very well.

1.81.2 Various responses

A variety of responses:

We don't expect to win the X-Prize, both because Burt probably has it in the bag, and we are behind schedule. We still plan on continuing our development, because our designs are nearly an order of magnitude cheaper to fabricate and operate than Space Ship One, and orders of magnitude matter. If SS1 crashes on Monday, we will throw more time and resources at an attempt, because there really is no other contender, but it will be a long shot.

We could have flown an unguided rocket to very high altitudes a long time ago, but we have instead concentrated on control systems, which is where the important work needs to be done. A team that was busy flying rockets to hundreds of thousands of feet altitude, then decided to add a guidance and control system to their rockets would be in for many rude surprises at high energy levels.

This isn't immediately obvious, but an X-Prize class vehicle pretty much requires an active control system (a trained pilot with appropriate controls is also an active control system). A short burn time rocket, like the recent CSXT 100 km shot, can live with just aerodynamic stabilization (note that it also landed 20 miles away), but the G forces are far too high for people. As the burn time lengthens with lower acceleration forces, the vehicle will gravity turn away from vertical, making it almost impossible to keep a 60 second burn time even accelerating upwards.

People that harp on about propellant specific impulse in the context of suborbital rockets are like programmers that obsessively optimize a function that isn't a hot spot. The goal of a rocket ship is not to deliver specific impulse, it is to move a payload reliably and cost effectively. Isp can always be traded away for mass fraction, and quite often you can improve operability or reliability by doing so. With our new vehicle designs using a single engine and jet vanes instead of four differentially throttled engines we are more likely to consider trading some engine and system complexity for performance, but issues like the requirement for deep throttling still make it a complex decision.

I do Armadillo work on Tuesdays, weekends, and late at night. At Id lately I have been working on next-generation rendering technology while the rest of the company manages the Doom beta process.

I don't issue press releases. I just publicly write about what I am working on, and other people find it noteworthy enough to talk about. All of our development work, including the dead ends and mistakes, is fairly well documented on the Armadillo Aerospace website.

1.81.3 Re:This is what a rocket ship SHOULD look like....

I figure you need about four times as much fuel at liftoff for a vertical rocket-borne landing as you would for a parachute- or wing-borne landing.

No, not even remotely close.

You only need enough propellant to kill the terminal velocity of the vehicle to land it safely. A vehicle that is stable reentering base first has a Cd right around 1.0, and any high performance rocket vehicle is going to be coming in pretty light after it has burned most of the propellant. The V2 impacted the ground still supersonic because it was aerodynamically stable nose first, so it maintained its 0.15 or so Cd on descent. A reasonably stubby base first reentry will have a terminal velocity of only 200 mph or so.

Killing that speed with a comfortable safety margin takes about 400 pounds of propellant in our vehicle, compared to 8000 pounds of propellant burned on ascent. A higher performance rocket engine could do it with propotionately smaller amounts. A parachute / drogue / ejection system for this weight vehicle is indeed lighter, coming in at about 100 pounds, but that brings a number of disadvantages with it, like coming down tens of miles away and still needing final impact attenuation.

We wanted to use parachtues as a quick hack for the X-Prize, but the test range where we were planning to fly was going to require a half million dollars of "engineering support" and wanted us to carry a thrust termination system (bomb) on the vehicle to satisfy themselves that it won't

drift out of the range.

Long term, there is no question in our minds that powered landing is the way to go. We just were given a pretty strong incentive to go there earlier than we were planning.

1.81.4 Re:This is what a rocket ship SHOULD look like....

Actually, you underestimated this one.

Since the "payload" of an X-Prize vehicle is three x 200 lb people, needing 400 pounds of landing propellant turns our 850 gallon tank vehicle from a three person vehicle into a one person vehicle.

In the most negative light, you could say that powered landing (with a low performance propellant like we use) takes away two thirds of our payload capacity, but that is a poor metric to base decisions off of, because operational issues have historically been orders of magnitude more important to cost effectiveness than propellant consumption.

We can get the 400 pounds back by either going to a carbon fiber tank instead of a fiberglass tank (cost: $40k up front design fee, then $25k per tank, compared to $9k for the fiberglass tank), or by upsizing everything to a 1600 gallon fiberglass tank (cost: $17k for the tank plus more for bigger engine plumbing, catalysts, and nozzles).

Upsizing the tank is lower risk, because it only uses suppliers that we already buy from, while the carbon fiber tank job would be custom from ATK, and I have already had two other vendors back out on me for big tank work. We already have a 1600 gallon fiberglass tank on hand.

Our mixed monoprop has a measured sea level Isp of 145, with normal increases with altitude. Our big vehicles have a mass ratio of about five, takes off with somewhat under one positive G of acceleration, and has a somewhat regressive thrust profile from partial blowdown pressurization. That combination is sufficient for suborbital flights. A 200+ Isp biprop can do it with a mass ratio of three, but the vehicle gets a lot more complicated to build and operate.

1.82 Games - Doom 3 Gets Reviews, Piracy Questions, Exultation - Aug 2004

1.82.1 Re:Quake3 engine open-source? When?

By the end of the year. There are still a lot of higher priority things, but it is coming soon.

Hopefully punkbuster will keep the source release from having any negative impact on the player community.

1.82.2 I'm proud of it.

I am extremely proud of Doom 3. I think it is the best game we have ever made, and it exceeded all of my expectations. That is a rather trite phrase, but it is literally true – I had a good set of expectations for how the game would turn out based on the technologies that it was built on, and it wound up being just plain better than that.

We think a lot of people will like it.

I don't follow gaming message boards, because, at its best, entertainment is going to be a subjective thing that can't win for everyone, while at worst, a particular game just becomes a random symbol for petty tribal behavior. This slashdot story is about as close as I want to go...

Amidst all the various Doom ports and expansions, we are starting up on our next game. It will have a new rendering engine, which will be keeping me busy for a while, but the only other thing we are saying for now is that it won't be a sequel to any of our previous work. We have a really solid team that did a lot of maturing through Doom's development, so I have high hopes that it won't be another four year odyssey.

1.83 Science - 1 Amateur Rocket Crashes, Another Explodes - Aug 2004

1.83.1 Re:doom3

I said the same thing – the puffs from the flying tank look just like a bad particle system that dropped points far too sparsely. Strange.

1.84 Games - In-Game Advertising Breaks Out - Aug 2004

1.84.1 Almost had one in Quake

We had a pretty good money offer to put a sponsored add in the Quake 1 entry level. We decided not to just on the basis of it being tacky, which was for the best, considering the company (some random early internet company) dissapeared into obscurity.

I don't have any fundamental problem with product placement in games, but it isn't something we pursue. I would just as soon have real brands in realistic settings instead of made up ones.

1.85 Games - Editorial: On the SpikeTV Video Game Awards - Dec 2004

1.85.1 Award shows

I did the Academy of Interactive Arts and Sciences awards show a few years ago – I was inducted into the hall of fame one year, then the next year I inducted Will Wright.

I hated it, but it is a big industry, and there is a broad range of people involved. Honestly, I'm almost certainly in the minority. One developer that I was talking to backstage was very bullish about how important it was to legitimize the industry with events like this, but I just don't have any empathy for what I perceive as "Hollywood envy".

Some award show issues are just a result of stupidity – I felt so bad watching Hironobu Sakaguchi of Squaresoft, a non-native english speaker, being forced to read a long speech written by some PR type about me. I threw out what they gave me to say about Will, and wrote something more to the point myself.

I do feel that there is a rather fundamental mismatch with big awards shows for game development, because game development isn't a performing art. You expect actors and musicians to show well, because that is what they do. Why aren't awards for authors the same glamorous events that the movie / TV / music ones are? Game developers are much closer to authors than actors.

1.86 Games - John Carmack's Cell Phone Adventures - Mar 2005

1.86.1 Re:That's gonna give the Java fanbois an aneurysm

I did try running that benchmark, but it won't load on the i730 (score one more for run-anywhere...).

One of our test platforms is a fairly high end Sony Ericsson, which is 10x as fast as our Motorola base platform. For a 128x128 screen, the Motorola renders about 4 fps and the Sony renders about 40 fps. Compare with Wolfenstein-3D performance (the DoomRPG engine has some extra graphics features, but it is still in that general class) at that resolution on older systems. A 386-16 would go significantly faster.

Note that the "As fast as a..." comparisons from the benchmark are against purely interpreted java on the P3, which is about 1/10th the speed of a

native implementation, and benchmarks that focus on expression and control operations will overestimate relative performance for applications that are array access heavy. Still, if a java app on that phone performed like a P3-100mhz, it would be pretty impressive.

It is true that a good JIT (which the phones don't have) can make java code go nearly as fast as C/C++ code that is written in the same style. The "in the same style" part is often overlooked – in lower level languages you often have options for implementation with pointers and CPU tailoring that would make the code look very different, but go significantly faster.

I still generally like java, and maximizing performance is only important in a rather limited subset of software engineering.

1.87 Science - Kansas Challenges Definition of Science - May 2005

1.87.1 Re:The Blind Watchmaker – great book on this subj

I just read that book recently, and while I enjoyed most of it, I found the chapter on the theories about the emergence of DNA extremely "hand wavey". The clay mineral culture idea was only presented as one possibility, but it didn't sound very convincing. If anyone has pointers to more compelling theories, I would be interested in reading them.

I always hated biology / life science in school because most of it was name memorization, but at the molecular biology level, it all starts looking digital...

1.88 Games - Are Video Game Patents Next? - Jun 2005

1.88.1 Parasites.

I'm proud that there is "a relative dearth of patent applications for the video game industry, especially considering how technology-dependent the video game industry is, and given its size in terms of annual sales."

Before issuing a condemnation, I try hard to think about it from their point of view – the laws of the land set the rules of the game, and lawyers are deeply confused at why some of us aren't using all the tools that the game gives us.

Patents are usually discussed in the context of someone "stealing" an idea from the long suffering lone inventor that devoted his life to creating this one brilliant idea, blah blah blah.

But in the majority of cases in software, patents effect independent invention. Get a dozen sharp programmers together, give them all a hard problem to work on, and a bunch of them will come up with solutions that would probably be patentable, and be similar enough that the first programmer to file the patent could sue the others for patent infringement.

Why should society reward that? What benefit does it bring? It doesn't help bring more, better, or cheaper products to market. Those all come from competition, not arbitrary monopolies. The programmer that filed the patent didn't work any harder because a patent might be available, solving the problem was his job and he had to do it anyway. Getting a patent is uncorrelated to any positive attributes, and just serves to allow either money or wasted effort to be extorted from generally unsuspecting and innocent people or companies.

Yes, it is a legal tool that may help you against your competitors, but I'll have no part of it. Its basically mugging someone.

I could waste hours going on about this. I really need to just write a po-

sition paper some day that I can cut and paste when this topic comes up.

1.89 Apple - Is Apple & Community Evangelizing Into Uncoolness? - Jun 2005

1.89.1 game performance

How much do you want to bet a bunch of those developers drop support for PPC Macs far sooner than the aforementioned "3-5 year" period and claim that the games demand the "performance" of the faster Intel machines. We already saw that when Doom 3 was released for the Mac. It supported only the very fastest Macs while leaving many other current and/or new Macs out in the lurch.

Does he think we just sit around and say "Lets just not support the rest of these macs because we want to screw the user base!"

We work with Apple, ATI, and Nvidia to make everything run as well as possible. Doom 3 had AltiVec code in it, and there were driver changes to make things work better. The bottom line is that the compiler / cpu / system / graphics card combinations available for macs has just never been as fast as the equivalent x86/windows systems. The performance gap is not a myth or the result of malicious developers trying to make your platform of choice look bad.

Yes, it is always possible to make an application faster, but expecting developers to work harder on the mac platform than on windows is not reasonable. The xbox version of Doom required extensive effort in both programming and content to get good performance, but it was justified because of the market. In hindsight, we probably should have waited and ported the xbox version of the game to the mac, which would have played on a broader range of hardware. Of course, then we would have taken criticism for only giving the mac community the "crippled, cut down version".

1.90 Science - Jeff Bezos's Space Company Reveals Some Secrets - Jun 2005

1.90.1 Re:No problems here

All the quenching problems were with our mixed-monoprop scheme that used low concentration (50%) peroxide mixed with a small amount of methanol.

If you can get high concentration peroxide (85%+), there are no catalyst quenching problems. We started out with 90% peroxide, and we would still be using it (and would have saved a year of work...) if we had a willing supplier. The original supplier we used went out of business, and the remaining domestic supplier didn't want to do business with us, even for $>$ $100,000 orders.

We did a number of peroxide / kerosene biprop tests back in August / September 2002 before we ran out of high concentration peroxide.

We are pretty happy with liquid oxygen now, but if Bezos is sure that supply won't be an issue, peroxide/kerosene is certainly not a bad choice. The sole drawback I would note is that it will put a lower limit on his operating expenses, and a LOX based system could potentially undercut him, although that would only be an issue when ticket prices are getting down towards $10k.

1.91 Books - Fab - Jun 2005

1.91.1 Rapid prototyping, etc

I have a good sized CNC mill in my garage that I use practically every week to make various rocket parts. It is certainly cool, but the realities of tool reach, work holding, and chip removal make it more of a "super power tool", rather than a free-form-fab.

The various technologies that essentially rasterize arbitrary parts are what excite the imagination, but I don't expect any radical changes in society any time soon from them. Stereolithography is pretty mature, and getting arbitrary parts rasterized in plastic is fairly common today. However, in 99% of the cases, these are still used as models / proof of concept / R&D, not actual manufacturing, because they are drastically more expensive than, say, injection molding, and more mechanically limited. There are a lot of technologies touted for rasterizing 3D metal parts, but I spent some time recently trying to find a place to fab modest sized rocket engines, and none of the companies I spoke with were able to handle it for various reasons.

I do expect this to become very exciting, but it is several years away. The excitement won't be about fabricating things that you currently buy (conventional mass production will retain significant cost benefits), but allowing low cost R&D. When you can send an arbitrary 3D CAD model over the net to a company with a metal rapid prototyping machine (they will remain expensive for quite some time) and get your part overnighted to you in a couple days with no setup fees, you will be able to iterate design cycles twice a week at quite low expense. You can do this today with plastic, and in some limited cases of small metal parts, but when you can start doing it in significant engineering materials that can be used in functional prototype machines, lots of new opportunities will arise.

1.92 Games - Quake 3: Arena Source GPL'ed - Aug 2005

1.92.1 Re:Unreal Engine 4

Yes, there was a budget title (Paintball somthing or another) that was developed based on the Q1 source that purchased a commercial license.

We didn't charge much, but I still think they should have just saved the money and released their source.

1.92.2 Re:Hmm

Personally, I think the Q3 code is pretty clean on the whole. It was a commercial product done under time pressure, so it isn't a polished gem, but I consider it good.

Anyone working on the Q3 codebase today should just delete all the asm code and use the C implementations. Making a commercial game with fairly high end requirements go 10% faster is sometimes worth writing some asm code, but years later when the frame rate pressure is essentially gone, the asm code should just be dumped in the name of maintainability. All the comments in the world wouldn't change this decision a bit.

But there's really little reason to use asm anymore, since the autovectorization in gcc is very nice.

I was pretty much with you until that point. I fully agree that there is little reason to use asm anymore (I haven't written any in years – Jan Paul did all the SIMD work for Doom 3). Knowledge of asm is good to allow you to manipulate compiler output by changing your C++ code, but there isn't much call for writing it by hand.

However, autovectorization in gcc is a particularly bad argument against writing asm code. Scalar compiler code is much, much closer to hand codeed asm in performance than compiler generated SIMD code is. Optimized SIMD coding almost always requires significant transformations that compilers can't really do on their own.

The argument about inline asm hurting compiler optimizations is only true if you are trying to use short snippets of asm, which is generally a bad idea. Asm code that doesn't loop a lot isn't likely to contribute significantly to your performance, with the exception of things like ftol replacements.

1.92.3 Re:And that's why id Software rocks.

Thank you.

1.93 Games - Doom Takes A Shot At Gamers - Oct 2005

1.93.1 I liked it.

I had fairly low expectations, and there were even some plans in palce to guide me away from any press after the premier if I didn't like the movie, so I wouldn't say something "unproductive", but I was pleasantly surprised.

No, it isn't an oscar movie, but it definitely isn't Super Mario Brothers / Street Fighter / Double Dragon.

I do wish they had kept the true satanic / hellish theme, but I think they did a credible job with their alternate direction.

1.94 Science - Are Liquid Explosives on a Plane Feasible? - Aug 2006

1.94.1 Re:John Carmack disagree's with the article

As a follow up, some people aren't realizing that it isn't necessary to actually have a chemical reaction and form an organic peroxide molecule to make an explosive. A solution of oxidizer and fuel can easily be a shock sensitive explosive. This requires higher concentration peroxide than is available off the shelf, but concentrating a modest amount is not very challenging.

The feasibility of this really isn't open for debate. There is no doubt that you can reliably mix two liquids and produce a high explosive that can be detonated with a sharp impact.

A quest for perfect safety from all conceivable threats is, of course, ridiculous, but I'm sure there will be many more added security measures thrown in as a result of this, to little real benefit and much general annoyance.

Personally, I would have been completely comfortable flying immediately after 9/11 with absolutely no additional security measures. Statistics and probability leave me with no fear of terrorism.

1.95 Ask - A Master's In CS or a Master's In Game Programming? - Nov 2006

1.95.1 Probably the CS degree.

Game programs have been somewhat useful for finding employees, but we don't actually think that the students are learning particularly valuable skills in the programs.

A CS or EE degree will almost certainly serve you better throughout your life than a game/media degree, but if getting into the industry immediately is your overriding concern, a game program will help with contacts and opportunities.

Exceptional merit will eventually be noticed (perhaps not as quickly as you would like, though), and a degree of any sort is not required if you can conclusively demonstrate that you will contribute great value to a company. However, many entry level positions are filled based on people's opinions about potential, and honest assessments from faculty that work with lots of students does carry some weight.

The best advice is "be amazing", but "diligent and experienced" counts for quite a bit.

1.96 Games - Silicon Knights Says Unreal Engine is Broken - Jul 2007

1.96.1 Re:This Is Rumor Control - Money Grab In Progress

I'm not sure where that quote came from – IdTech 5 as a whole is not a clean sheet of paper design, there are some good sized chunks that have clear heritage to Doom 3 / Q4 / ETQW. The rendering engine is certainly from-scratch, but that is a rather small chunk of a complete engine today.

I was always somewhat hesitant about broad licensing because I feared something exactly like this, where a developer thinks they see something in an engine, but it doesn't turn out the way they expected, and they sue. It is possible that explicit promises were made and broken, but it is also possible that the licensee just failed for the same reasons that most game development project fail, and is looking for a scapegoat. Game development is hard, engine license or no engine license.

During Doom 3's development, our licensees had access to our source control server, so there was never a question of them not having access to what we are using. They would have been foolish to try to use daily builds, but the option was available to them.

1.97 Games - Is id Abandoning Linux? - Sep 2007

1.97.1 A direct response

There is certainly no plans for a commercially supported linux version of Rage, but there will very likely be a linux executable made available. It isn't running at the moment, but we have had it compiled in the past. Running on additional platforms usually provides some code quality advantages, and it really only takes one interested programmer to make it happen.

The PC version is still OpenGL, but it is possible that could change before

release. The actual API code is not very large, and the vertex / fragment code can be easily translated between cg/hlsl/glsl as necessary. I am going to at least consider OpenGL 3.0 as a target, if Nvidia, ATI, and Intel all have decent support. There really won't be any performance difference between GL 2.0 / GL 3.0 / D3D, so the api decision will be based on secondary factors, of which inertia is one.

1.98 Mobile - An App Store For iPhone Software - Mar 2008

1.98.1 Re:Mr. Carmack are you still around?

We (Id) have put in our application like everyone else, so I don't have any inside information at this point. I think Steve is still pissed at me over some negative comments I made about iPod development tools a while ago. Just based on the blurbs, it looks very good – a simulator plus debugging on the native device is the best of both worlds, and a 70% royalty deal for apps over iTunes is quite good.

The iTunes distribution channel is really a more important aspect than a lot of people understand. The ability to distribute larger applications than the over-the-air limits and effectively market your title with more than a dozen character deck name, combined with the reasonable income split make this look like a very interesting market. This type of developer / customer interaction is probably the wave of the future for mobile devices, it will be interesting to see how quickly the other players can react. Based on our experiences with the carriers, I am betting not very quickly.

1.99 Games - Carmack Speaks On Ray Tracing, Future id Engines - Mar 2008

1.99.1 Re:Stunning!

Give me a little credit here. I am not suggesting that everyone blindly intersects rays with a huge list of triangles. That would be absurd, and I assumed everyone understood that. What you might have missed is that I'm not proposing a sparse voxel octree as some form of bounding hierarchy to reduce intersection tests against triangles, I am proposing that it REPLACE hierarchies of triangles or other primitives for some data sets, and this brings about significant improvements (data size) that you wouldn't have with even infinitely fast conventional ray tracing. I'm also not trying to say that this is some novel brainstorm of mine, but I have some practical experience with the direction, and I think it has promise.

One of my major points is that this is all still theoretical. I don't know what is going to be the right architecture for next gen systems. Neither do you, or Intel, Nvidia, Microsoft, Sony, or Nintendo. If I had to place a bet, it is that rasterization will still be dominant, but it is a Good Thing to have lots of people doing research into various alternatives. All the players have their own agendas, but we will all know the big win when we see it.

1.99.2 Re:Senor Carmack, one question

In our current generation codebase we have moved to completely separate representations for rendering and physics, and I expect that to continue in the future. The requirements are different enough to merit different internal storage.

1.99.3 Re:acceleration structures, etc...

Tracing into an SVO structure can be done with almost a Bresenham algorithm, and when you reach whatever depth of descent you want (a critical factor, you aren't going all the way to final detail on every trace), you pop out whatever data is stored there (probably some vector quantized BRDF sort of thingy) and run a "fragment program" on it.

The data sets for a game world represented like this would be massive, but it is intrinsically well suited for streaming, even over the internet, which may well be the dominant media distribution channel in the timeframe we are looking at.

1.100 Tech - Rocket Racing League Ready To Launch - Apr 2008

1.100.1 Re:Try keeping THAT "Carbon Neutral"

Not that I give a damn about being carbon neutral, but our rocket engines do burn ethanol.

1.101 Hardware - The Science of Iron Man - May 2008

1.101.1 Re:Iron Man's Suit Defies Physics – Mostly

Hydrogen peroxide powered rocket packs fly for around 30 seconds, because they have a specific impulse of around 125, meaning that one pound of propellant can make 125 pound-seconds of thrust, meaning that it takes about two pounds of propellant for every second you are in the air. Mass ratios are low for anything strapped to a human, so the exponential nature of the rocket equation can be safely ignored.

A pretty hot (both literally and figuratively) bipropellant rocket could manage about twice the specific impulse, and you could carry somewhat heavier tanks, but two minutes of flight on a rocket pack is probably about the upper limit with conventional propellants.

However, an actual jet pack that used atmospheric oxygen could have an Isp ten times higher, allowing theoretical flights of fifteen minutes or so. Here, it really is a matter of technical development, since jet engines have thrust to weight ratios too low to make it practical. There is movement on this technical front, but it will still take a while.