

Interview with Hans de Ruiter

Software News

Posted by: Sbaitso

Posted on : 2019/2/24 19:28:16

Prior to AmiWest 2018, I was lucky enough to be granted an interview with Hans de Ruiter. Hans graciously provided an overview of video driver development for AmigaOS as well as insight into some of the complexities and challenges he faces.

1. Can you please provide an overview of your history as an Amiga user? When were you first introduced to it? What drew you to it? What keeps you a fan of it?

I originally started on a Commodore 64. Friends started using a machine called the Commodore Amiga, which was clearly a good step up from the C64. Eventually, my parents got an A600. Later, I bought a second hand A1200, which I later upgraded with an 030 accelerator and did my own tower conversion (the conversion kit + shipping from the UK was too expensive for me). The A1200 served me well till the end of undergraduate engineering. Then I decided to do a Masters/Ph.D. at the University of Toronto. I couldn't take that full tower with me to Canada, but the AmigaOne XE came on the market soon. So I got one of those and waited impatiently for AmigaOS 4.0, and I've continued with AmigaOS since.

I originally liked the Amiga for its games, but quickly also enjoyed the creative side of it. As a result I dabbled in everything from Octamed to Real 3D and, of course, programming. Another drawcard for the Amiga was its graphics and capabilities, which was ahead of its time early on.

The main thing that keeps me here, is I enjoy the challenge of pushing AmigaOS forward. It retains its original charm, and we (the AmigaOS dev-team) have pushed it far beyond its original design.

2. How do you become involved in the development of video drivers? Did you work with video drivers for other platforms prior to developing them for Amiga? Have you developed any for platforms since you began your Amiga development?

That happened close to the end of my Ph.D. AMD opened up the documentation for the Radeon X1000 (R500) series, and there was lots of chatter on the forums about the possibility of new drivers. There was now proper documentation for the hardware, so the old excuse of "open-source code is *not* documentation" no longer applied.

At the time I was dreaming about writing my own video editing software, and I definitely wanted drivers for newer cards. I noticed that there was lots of talk, but nobody was taking any action. So, I downloaded the documentation and had a look at it. Next, I had a look at the new open-source RadeonHD driver for Linux. "Hmmm, looks doable," I thought, "Are there any cards that could plug into my A1-XE for a price I can afford?" Yes, on both counts.

So, I found and ordered a PCI Radeon X1300 on eBay, and plugged it in to my A1-XE nervously, hoping like crazy that it didn't kill the machine. To my delight, not only was the machine fine, but I was greeted with a UBoot screen on the new card. So far so good...

The next step was to get it to display something. At this stage I hadn't written a graphics driver before. In fact, the only "drivers" I'd written were for the small peripherals on PIC/Atmega microcontrollers. So I decided to write a program that can open a screen and display an image first. If I could do that, then I could ask for the documentation needed to write a driver.

Alas, the card didn't seem to respond to any register write I did. After a while I discovered the PCI-to-PCIe bridge chip was misconfigured. UBoot disabled the bridge, meaning that it never passed on any of the accesses to the graphics card.

With that out of the way, I started adapting the Linux source code. Radeon cards have an AtomBIOS in the ROM with platform-independent code on it. I wanted to use that code, which meant I needed an AtomBIOS interpreter. The documentation was missing those details, so the only place to get it was their newly released source code.

Eventually, I got the card to open a screen and display an image, which I announced on my personal website:

[https://hdlab.org.nz/projects/amiga-o ... og/radeonhd-first-screen/](https://hdlab.org.nz/projects/amiga-o...og/radeonhd-first-screen/)

The story continues below...

3. What lead you to Amiga driver development? How did the opportunity arise?

I contacted Hyperion to get the driver development kit once I had a working test program. Having a working proof-of-concept convinced them it was worth giving me access, although that still took a while. Communication with Hyperion was quite slow, and I needed to sign an NDA first. Eventually that was done, and I could get to work on an actual driver.

The development progress is recorded on my website:

[https://hdlab.org.nz/projects/amiga-o ... radeonhd-development-log/](https://hdlab.org.nz/projects/amiga-o...radeonhd-development-log/)

4. Can you please provide a brief history, prior to your involvement, of video driver development on Amiga?

I'm not too sure of the details because I wasn't involved. There were multiple attempts to add graphics cards to AmigaOS by third parties, all of which involved patching the graphics library. The biggest two are Cybergraphics (used by MorphOS), and Picasso96, which became the foundation for AmigaOS 4.x's graphics driver system.

In 1998 Haage & Partner released Warp3D for 3D graphics. That worked with both Cybergraphics and Picasso96. However, its performance was limited by two factors:

1. It's an external add on to two third-party patches to the graphics library. As such, it has to play tricks to temporarily wrestle control of the graphics card from Cybergraphics/Picasso96
2. Warp3D was designed around the capabilities of the first 3D graphics cards, which were very

basic. They were basically 2D rasterizers with a depth buffer. There was no hardware Transformation, Clipping, & Lighting (TCL), and so Warp3D very quickly became dated

I managed to work around the first limitation by building a render manager into the Radeon HD driver (RadeonHD_RM.resource). However, the second limitation couldn't be overcome without a new API (Application Programmers Interface).

5. Can you please provide an overview of the current state of video driver development for Amiga?

As far as I know, I'm the only one still actively working on graphics drivers on AmigaOS. At present we have both 2D and 3D drivers for modern graphics cards. I recently completed drivers for Polaris series GPUs (RadeonRX 5xx series).

Warp3D Nova brings modern shader based graphics to AmigaOS, and we have an OpenGL ES 2.x (GLES2) wrapper. The GLES2 wrapper was my suggestion. OpenGL ES ditches old legacy stuff that shouldn't be used any more while keeping the most useful features. Ditching the legacy stuff also makes writing drivers easier. Plus, Regal and GL4ES are two libraries that could provide full OpenGL on top. It was the fastest way to get modern OpenGL.

Shader based graphics opens up a whole new world of possibilities. Apart from enabling hardware TCL and higher performance, it allows developers to write custom code that runs on the GPU itself. Only a few games take advantage of it right now (Spencer and Amicraft), but more are coming.

6. Where do you foresee driver development for the platform headed in the future? If logistics (cost, market size, etc.) were not an issue, where would you like to see it head?

Part of the task is writing drivers for newer hardware so we're still able to buy new cards (e.g., the recently completed Polaris drivers). The other side is pushing the driver capabilities forward. Warp3D Nova has improved a lot over the last year, but there are still missing bits here and there (e.g., missing shader instructions, geometry and compute shaders, etc.).

Beyond that, I'd like to see more of the graphics card's capabilities unlocked. The biggest one is to use the Graphics Address Remapping Table (GART) to allow the graphics card to access data in main memory via DMA. Actually, these days GART is more of a complete IOMMU. Either way, allowing the graphics card to read/write main memory directly would boost the performance quite a bit.

The other big item would be using the onboard video decoder for video playback. Plus the encoder too for video editing and recording.

Oh, and I'd like to see the driver API replaced with a modern one that has support for stuff like: multiple monitors, VRAM that the CPU can't access, hot-plugging, etc.

7. What is the single biggest hurdle in the development of video drivers that you don't feel the average user is probably aware of?

GPUs are really complicated. The Vega 10 GPU has 12.5 billion transistors, and packs a lot of technology. Even earlier Radeon HD cards were insanely complicated, with many subsystems, each getting increasingly complex.

Every generation has added a new level of complexity. I've also found the cards to be very finicky. Make one tiny mistake, and it locks up. I get the impression that the ATI/AMD vs nVidia race for bragging rights on the fastest card means that these cards are operating pretty close to the edge.

The next issue is lack of debugging tools. When the GPU or driver crashes, you lose your display, meaning no debug output on-screen. Added to that, we have no remote debugger, so I can't use a separate machine to step through code and analyze what's happening. This means inserting a lot of DebugPrintF() statements, and then going through crash and reboot cycles.

Even if we had a good debugger, it doesn't debug GPU shader code, and documentation can be wrong. This means that the only feedback you have is "the GPU locked up," with a few status registers giving a vague idea of what bits locked up. With so little data on what's wrong, debugging descends into a painstaking "guess and check" routine. Guess, test, crash. Guess again, test again, crash again. Guess again... well, you get the picture. Sometimes you get lucky. Other times it can take days or even longer until you stumble on the root cause and get it working.

8. What would you like the community as a whole to better understand about video driver development?

There's only so much that one developer can do. Graphics driver development is normally a team effort. Given our limited resources (basically, it's just me working on it), priorities have to be made. So, as much as I'd love to see the entire Radeon HD series to be fully supported, it's just not feasible with only one developer working on it.

The same goes for adding features such as video decoding, GART support, power management, etc. Priorities have to be made on what to tackle first. I've seen people insist that the one feature they want most is what everyone wants. However, just because it's most important to you, doesn't mean that it's most important for everybody.

9. Can you provide an overview of some of the issues that you encounter while developing drivers for the Amiga that might not be encountered on other platforms?

Lack of a working remote debugger (or just a working debugger) is the biggest one. I've also dealt with UBoot/CFE/BIOS problems that don't exist on other platforms. Other platforms also have more modern graphics driver APIs and systems. I'm having to pull tricks to work around Picasso96's ageing design.

Finally, most of those developing drivers for other platforms work for the companies that design the GPUs.** That means they:

- Get to talk to the hardware designers
- Have access to hardware debugging tools we can't afford
- Work together in teams to write the drivers
- Get paid a lot more to work on it

Nevertheless, we are making progress. AmigaOS' graphics capabilities are lightyears ahead of where it was back in 2008, when I started.

** An AMD employee told me that I'm one of the few people in the world who can write graphics drivers that don't already work for them or one of their competitors.

10. Other than purchasing the drivers and participating in your annual survey how can the average user assist you in your development, if possible?

Buy software/games from other developers, and generally encourage them to develop more for AmigaOS. Graphics drivers are pretty boring unless there's other software that uses them. This is especially true of Warp3D Nova/GLES2. We need more software that takes advantage of the new capabilities.

Enjoy using your AmigaOS machines, use it creatively, and tell others about it. The more AmigaOS users we have, the larger the market, the more people can develop software of all types.

Oh, and if you happen to also be a developer and would like to give writing graphics drivers a shot, then consider talking to A-EON.

11. There always seems to be a lot of discussion and sometimes some confusion about benchmarks. Can you talk about benchmarking as a whole as well as specifically how it relates to the Amiga? What is the best software to use to obtain accurate benchmarks on the Amiga and how does a user interpret those results? What's the best way to compare those Amiga based benchmark results with benchmarks obtained from other platforms? How does a user compare "apples to apples"?

Ah, benchmarks. So many arguments around that one. I've seen some insist that synthetic benchmarks like GfxBench2D are invalid because they "don't reflect real-world performance." Respectfully, I disagree. "Real-world" benchmarks are no more valid or informative. For example, running the Quake III timedemo only tells you how that time demo performed; you can't infer what the frame-rate would be for another game, or even a different Quake III demo/level. They do give you relative performance of different hardware, though.

I prefer simple benchmarks such as the ones I used in GfxBench2D, because they probe the performance and limits of specific parts of the hardware. That's data that could be used to identify driver bottlenecks, and fix them. App/game developers could also use them to choose which operations they use (e.g., avoiding the ones you know will be slow). If I were to write benchmarks for 3D, then I'd construct benchmarks to probe things such as: draw-ops/s, vertices/s, fill rate, shader ops/s. That kind of data is useful for developers.

I don't have a list of benchmarks people can/should use, but do advise using multiple benchmarks. Drawing big conclusions from one benchmark is a bad idea. For example, the famous Cow3D test was affected by a bottleneck that made Warp3D Nova look bad:

<https://keasigmadelta.com/blog/warp3d-...ially-hidden-by-lazy-cow/>

Even with that issue removed, Cow3D's performance is still hitting one particular bottleneck (most

likely draw-ops/s), and says nothing about the performance of other software. So, run multiple benchmarks and don't rely on any particular one.

As far as comparing Amiga benchmarks to other platforms, I suggest testing a large range of different benchmarks, and be careful about benchmark versions. For example, there's no point in comparing Blender 2.48 render times vs Blender 2.79. Also be aware that compiler optimizations can play a big role, and that benchmarks can be optimized for specific hardware (or have hand-coded optimizations for multiple platforms).

Ultimately, people need to accept the results. If machine A performs H.264 decoding faster than machine B, then machine A performed faster than B. The exact reasons for that, and what could be done to make machine B perform better is another matter...

12. Is there anything you would like to add that wasn't covered?

No. Can't think of anything.