

OSS Driver Dev Funding

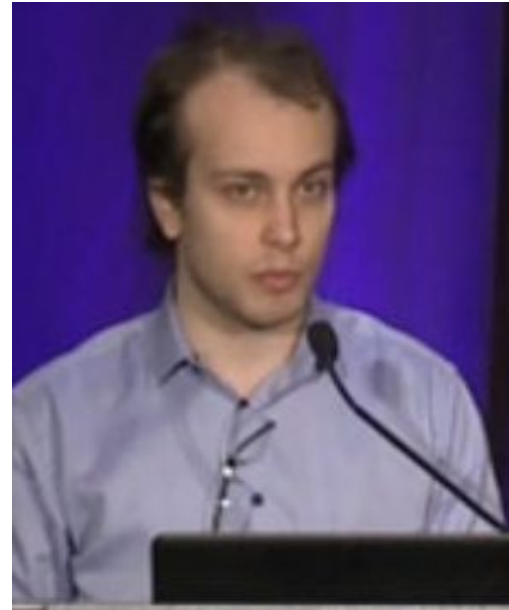
Hooking up the Money Hose

Jens Owen [Google], Pierre-Loup Griffais [Valve]

Introduction



Jens from Google



Pierre-Loup from Valve

Overview

Benefits open source 3D driver brings to game development, and...

Benefits these drivers will ultimately bring to gamers themselves

It's About Scale

As the gaming industry grows to exceed \$100B in revenue in 2018, the opportunity for open platforms and open drivers continues to grow with it

- It's about scale
- Even the largest, most well funded, driver support teams can't address all the needs of AAA studios and the long tail of Indie developers
- We believe an open approach enables DIY that can scale to meet this challenge, but we will need a good understanding of what practices scale to these large ecosystems with multiple GPU architectures supporting thousands of titles

Pierre-Loup's ramblings on open drivers

- Benefits from a game/app developers' perspective
- Same benefits and their side-effects from the hardware vendor's perspective
- Platform vendors considerations
- The Mesa model

Game developer using open drivers

Well-known set of benefits, pretty straightforward:

- Possible to diagnose game-side bug causing driver crash
- Easier to profile slow path triggered by game-side behavior
- Can even hack around a codepath or apply test patches; it does happen in practice!
- In general, more tools to unblock yourself

HW vendor perspective

All these same points are much more valuable to the HW vendor:

- Vast majority of app/game developers are not able (or even willing) to have a direct line to your driver team, nor are you able to support them all
- If the rootcause isn't immediately obvious, they will disable the problematic code path for your HW and move on to the next burning thing
- You will often find out after the game has launched; they might or might not be in a position to do post-launch support
- **Open drivers increase the likelihood of your HW being tested, and well-exercised, across larger swaths of content**
- Reduces cost of triaging; more information upfront in bugs
- Recruiting

Platform vendors

Open drivers increase the likelihood of your HW winning platform designs:

- Even if most of the driver is directly funded by the HW vendor, the platform vendor can supplement their own resources and accelerate development
- They can prototype features without third-party resourcing and schedule considerations, removes a great deal of risk from their perspective
- **Just like with content, closed drivers might make you miss platform opportunities you might not even know existed until it's too late**

The Mesa model

Some platforms have unilateral “driver models”; Mesa/DRM gets to the same point organically:

- Drivers self-organize and converge around common systems like Gallium/NIR/etc if it makes sense for them
- These common pieces of code are very well-exercised, code quality is high due to large amount of eyes on the code
- Bringing up new HW becomes much less work as tons of work can be leveraged
- **Reduces driver development cost**; arguably not as much as sharing 90% of the stack wholesale from your primary platform, but end-result is a better fit
- Share smaller self-contained modules instead? Shader compiler, addrlib, etc.

What open drivers don't do for you

- Still need a full team of in-house devs
 - New HW support needs coordinated effort behind the scenes
 - Same with unreleased content support, new API features, etc
- **QA/testing is still extremely important**, you'll want to be very hands-on there, community-driven development isn't great at avoiding regressions

Back to Jens

Vision

A world where every engine developer utilizes your driver to ensure they can deliver a great gaming experience

- What would you need from these devs to ensure you can deliver fixes before the game launched?
- What would you need from a regression test suite to ensure the existing back catalogue of games continues to render well?

What Do Engine Devs Need from OSS Drivers?

- Reach: driver adoption on the most popular platforms
- Reach: support for older devices that make up a big percentage of their target market
- Timeliness: ability for driver fixes to propagate to gamers before new title launch

Challenges

There are tough engineering challenges we can work on together, as a community

- How do we ensure happy game devs when using our drivers?
- How do we ensure a healthy community of 3D driver devs?
- What are the most effective regression tests? Which ones cause unwanted noise?
- How do we capture game work loads across GPU architectures?
- How do we capture engine bugs to improve validation?
- How do we capture driver bugs to improve conformance?

Conclusion

We face many challenges pushing open source drivers to scale, but we have fundamental advantages with our open, community based approach