gamescope update

Pierre-Loup A. Griffais, Valve
steamcompmgr 2.0

- Embedded gaming session with automatic fullscreen scaling
- GLX -> Wayland
- DRM output for zero-copy in most cases
- Vulkan async compute if it needs to composite
- Aligns the game to output vblank and presents newest content available at the tightest possible deadline before vblank, no getting ahead
wlroots

- Pretty quick to get started
- Level of abstraction seemed low enough to match our domain-specific needs
- Just a couple of impedance mismatches at first
- Started from rootston, grafted steamcompmgr’s output pipeline on it, then removed rootston bit by bit.
Desktop applications

- “X namespaces” have been a goal of ours for a long time
- Had all sorts of plans to be able to apply the same sort of input/output transformations robustly in a general way on a desktop
- Nested desktop scenario was initially just a useful development tool, but starting to have some potential uses for real desktop gaming now
Some desktop demos

- Output size abstraction
- Display abstraction: ultrawide / multi-monitor
- Refresh rate abstraction
- Unified Vulkan-based output pipeline
TODO

- More polish around the nested experience, more mimicry of the game window, etc
- Get DRM planes working more consistently
High-level stack takeaways

- Modifiers badly need plumbing across the stack
- Implicit sync is not ideal - would like a fence to be paired with incoming DMA-BUFs instead, that we could wait on the GPU for late-latch
- Display drivers in the kernel remain buggy, and any non-trivial paths are not currently well exercised
- Need tighter control over the application’s cadence to further reduce latency. The compositor needs to be optionally able to dictate when glXSwapBuffers() and vkAcquireNextImageKHR() returns. This can be hacked by locking buffers now but not ideal.
Thank you!

- https://github.com/Plagman/gamescope/