

Rainbow Frogs

HDR + Colour Management in Gamescope



Rainbow Frogs

HDR + Colour Management in Gamescope

[ABRIDGED]



Hi! I'm Josh



What is Gamescope?



What is Gamescope?

- Wayland compositor for gaming. Used on Steam Deck.



What is Gamescope?

- Wayland compositor for gaming. Used on Steam Deck.
- Low latency



What is Gamescope?

- Wayland compositor for gaming. Used on Steam Deck.
- Low latency
- Low power consumption



HDR Display timing Low Latency VRR
Colour Management Session



Gamescope does EVERYTHING!

10-bit output Nested Low Power “Looks”
Console-ish UX Linear blending Planes

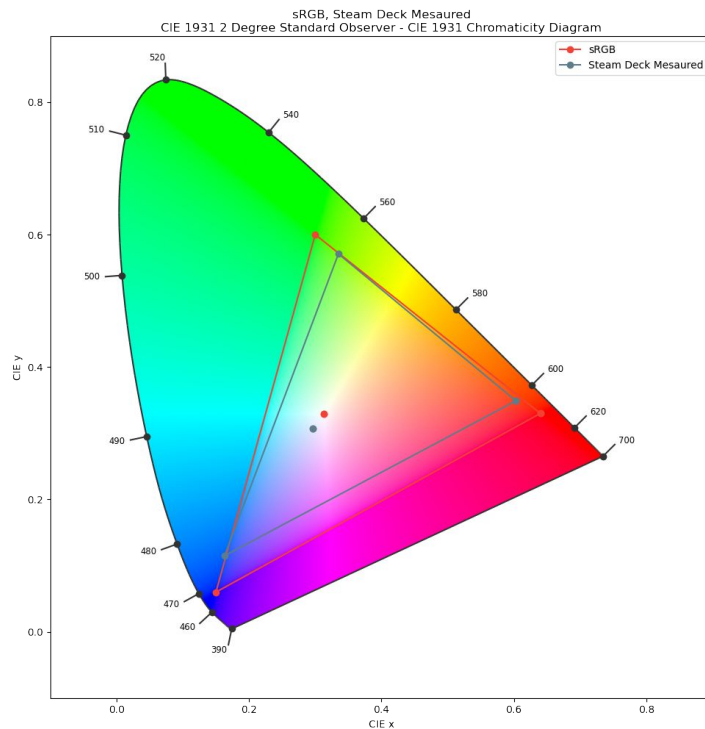
What is Gamescope?

- Wayland compositor for gaming. Used on Steam Deck.
- Low latency
- Low power consumption
- ...
- **Now includes Colour Management!**
 - (Blending, HDR, 10 bits, gamut mapping)



Gamut Mapping on Steam Deck

- Internal display has a modest gamut (Only ~72% sRGB)
- Customers expect more vibrancy



Gamut Mapping on Steam Deck

Naïve CTM approach:

Use a CTM (color transform matrix) to go from Rec.709 \rightarrow XYZ \rightarrow Native Display Primaries.

Pro: Makes it more vibrant!

Cons: Horrible clipping!
(we tried this, and chose not to ship it)





AMD DisplayCore is cool!



AMD DisplayCore is cool!

You'd know this if you saw Melissa's talk just before mine ;)
<https://indico.freedesktop.org/event/4/contributions/186/>

AMD DisplayCore is cool

Lots of features!

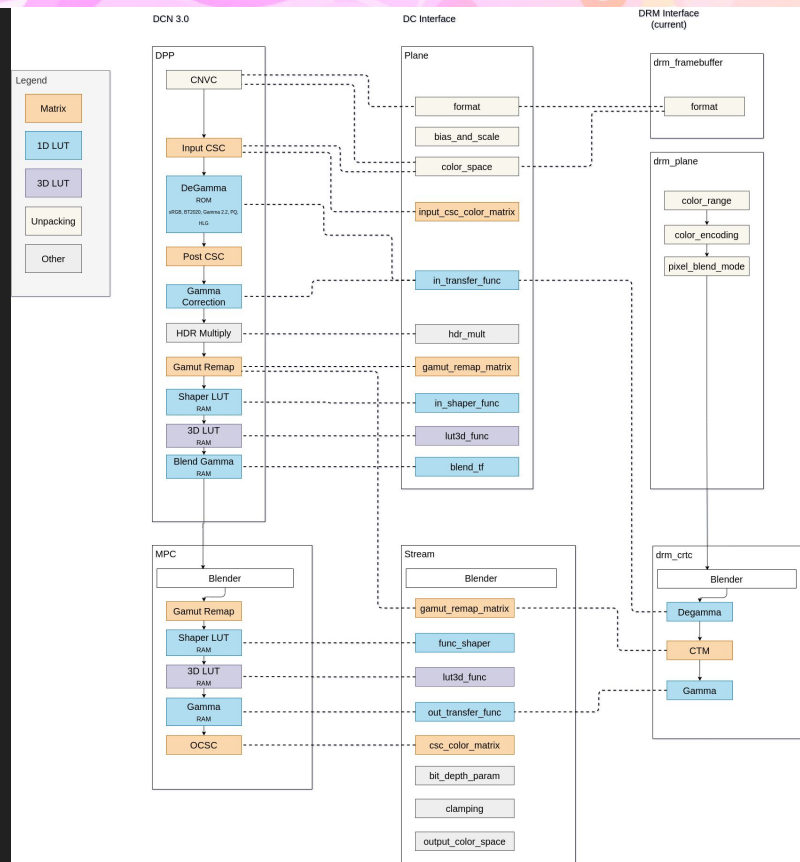
Planes:

- Degamma ROM and Degamma LUT
- Lots of CTMs (3x4!)
- HDR Multiply
- **Shaper + 3D LUT**
- Blend LUT

CRTC:

- **Shaper + 3D LUT**
- Regamma LUT

Literally everything you could ever want! :D



Gamut Mapping on Steam Deck

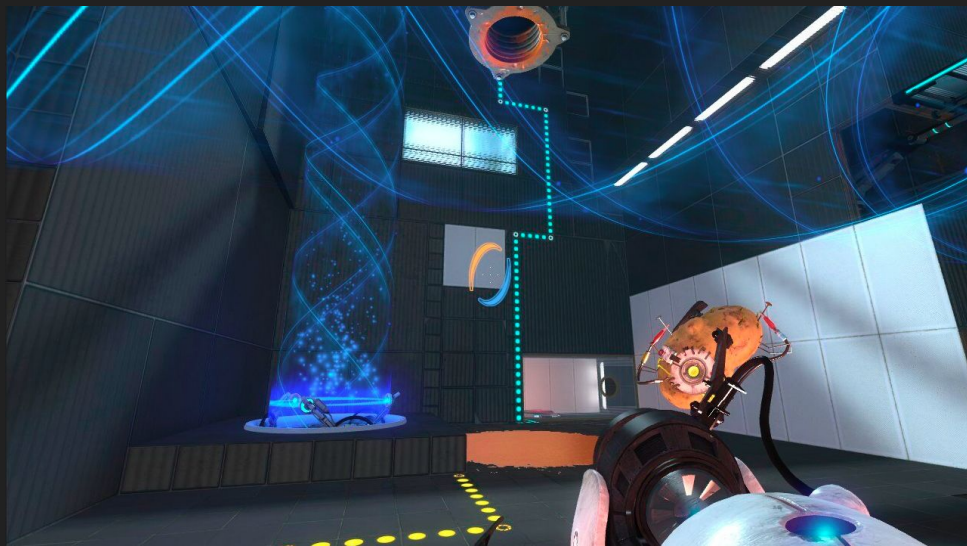
3D LUTs allow much greater control!

Use CTM near gray axis

Smoothly decrease influence near colour gamut edges (aka. gamut mapping)

Image matches expectations, no clipping!

We shipped this as part of the
Steam OS 3.5 Update! :D



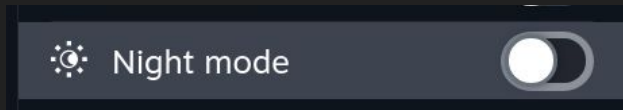
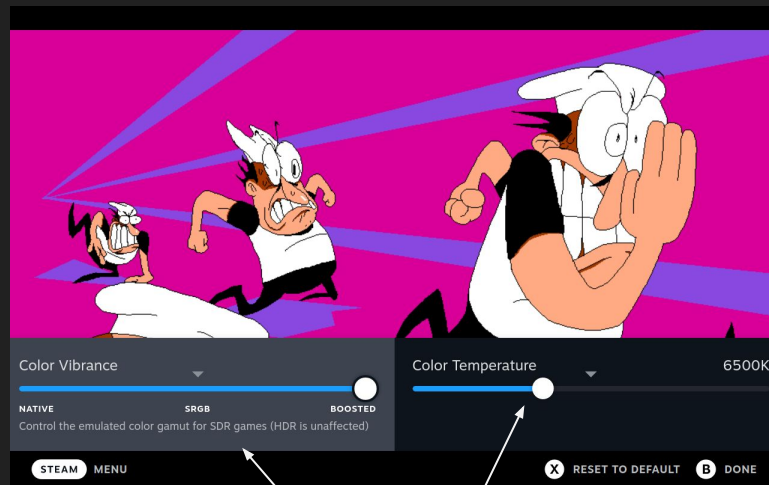
Using a 3D LUT for Colour Management

Gamescope *now* uses per-plane 3D LUTs to concatenate *all* colour processing.

Performance is constant, regardless of the whatever we want to do.

No latency increase!

(Could do HDR + Tonemapping + Night Mode all at the same time with no extra cost!)



All color controls turn into a single 3D LUT!

// TODO: Insert image of HDR on a TV



HDR is hard?



HDR is hard?



Not for us!

What's hard about HDR?

- You can have mixed HDR/SDR windows visible (even on Deck)
- Multitude of HDR formats (PQ/HLG/scRGB)
- Internal color processing requires $> 8\text{bpc}$
- Parsing and Patching EDIDs
- All our HDR content is Win32 (X11!)
- Undocking HDR games
- Supporting HDR through the full stack is just a lot of work! (DXVK/ Vulkan / Proton)



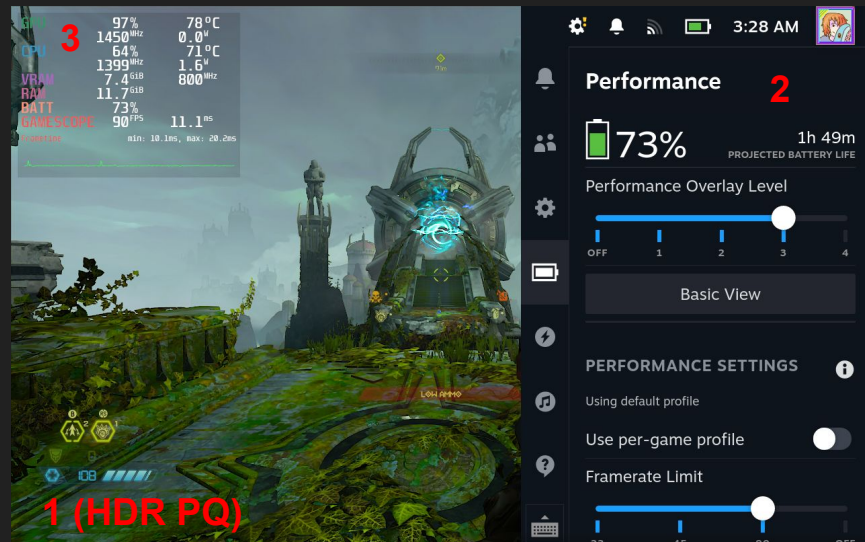
Handling mixed SDR + HDR content

Gamescope, despite showing a single window still can have mixed blended content:

- Steam overlays
- Performance overlay (mangoapp)
- Dropdowns/popups/modals

Solution:

- Shaper + 3D LUT for Gamma 2.2 → PQ
- Blend LUT → Linearized display referred
- CRTC Regamma → PQ



HDR Game + Perf Overlay + QAM
All scanned out!

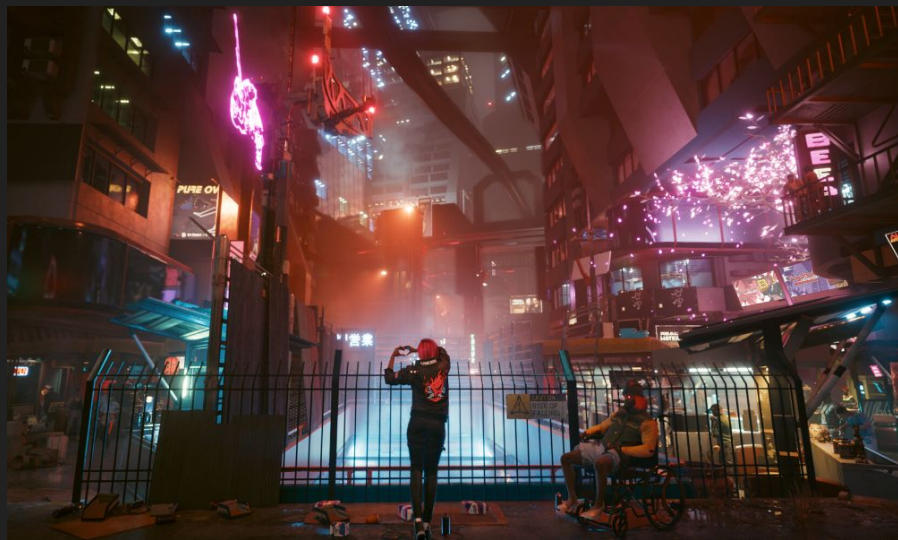
Handling scRGB (on HDR)

Some games deliver scRGB, which relies on negative components.

This would not work with our 3D LUT model!

Use a CTM before to convert from Rec.709 → Rec.2020.

Shaper takes us from scRGB → PQ!



Cyberpunk 2077 supports scRGB HDR

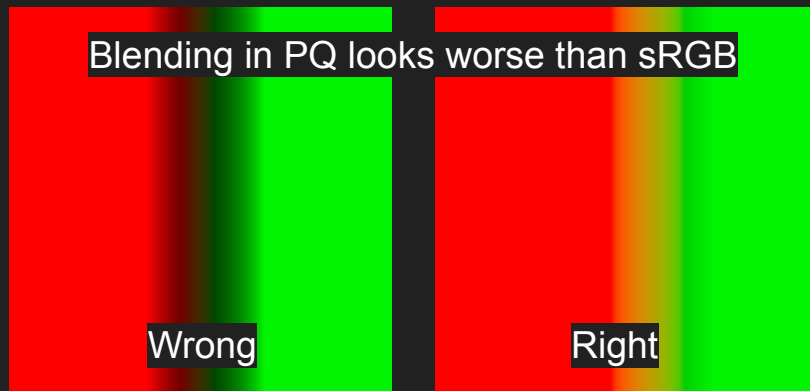
Making scanout identical to composite

Translated Tetrahedral 3D LUT interpolation from OpenColorIO to shader with a 3D texture

We also use this for blending 'looks' into final 3D LUT

AMDGPU applies Degamma for each tap, so do bilinear manually.

- *textureGather* to get each four texels for each colour, apply to each sample, blend based on each pixel's weight.
- Important as most people will be using 720p scaled up





End-to-end HDR is a lot!

Supporting HDR is a lot of work

Initially, didn't really know anything, so decided to try bring up in VK_KHR_display.

Hans-Kristian made a Granite sample for HDR which I used to test with and compare against Windows behaviour.

I should come back to this at some point, but no consumers yet.

- Maybe for HDR + VR in the future?

https://gitlab.freedesktop.org/mesa/mesa/-/merge_requests/20220

Supporting HDR is a lot of work

Need to add some kernel support so we can get Rec.2020 on the output

- “Colorspace” property was added and entirely untested/unused by userspace
 - Bruh
 - It’s what is supposed to be marking stuff as Rec.2020 colorimetry over DP/HDMI.
- No support in AMDGPU previously, but Harry had some patches!
 - Yay!!
- Worked with Harry and others to clean those up and find an amicable solution to the problems.
 - Thanks everyone! <3

Supporting HDR is a lot of work

Need support for doing all sorts of display stuff at scanout.

- We already had Shaper + 3D LUT from before (thanks Melissa)
- Worked with Melissa to split and implement other features to userspace
 - Plane Degamma TF + LUT (so we can use ROMs for better power + bw + accuracy)
 - Plane CTMs (including 3x4)
 - Plane HDR Multiplier
 - Plane Blend TF + LUT
 - CRTC Regamma TF + LUT
- Worked closely with Harry with accuracy/precision issues we were seeing wrt. banding when using all these features.
 - Thanks a lot for helping us there! Really really appreciate it. :D

Supporting HDR is a lot of work

Putting the display in HDR mode

- Grab the EDID from DRM
- Check if it supports PQ + BT.2020 using libdisplay_info
- Set the “Colorspace” DRM property for the connector to BT.2020
- Set the “HDR_OUTPUT_METADATA” property to one with the PQ EOTF selected
 - Currently using some sane defaults for SDR/HDR without app metadata:
 - `max_display_mastering_luminance = desired_content_max_luminance`
 - `min_display_mastering_luminance = desired_content_min_luminance`
 - `max_cll = desired_content_max_frame_avg_luminance`
 - `max_fall = desired_content_max_frame_avg_luminance`
 - Don't set the display's max luminance as the default max content light level or some TVs will start aggressively tonemapping and look very broken.

Supporting HDR is a lot of work

Need some way to read the EDID in both Gamescope + DXVK

- Opted for libdisplay-info by Simon Ser and others
 - Really nice API, works well, does what it says. Awesome!
 - <https://gitlab.freedesktop.org/emersion/libdisplay-info>
- No Windows or MSVC support! :(ul>- DXVK runs on Windows, and is a PE Win32 dll
- Had to do a fork to support that
 - Simon pls I beg u!

Supporting HDR is a lot of work

HDR from DXVK/VKD3D-Proton side (DXGI Output)

- Gamescope provides an EDID via a path in the `GAMESCOPE_DISPLAY_EDID_PATH` property on the root window.
 - XWayland provides some dummy fake EDID – this also allows us to expose the proper one to the game with the right size, name, etc.
 - We also patch it to hide Deck screen rotation (games are stoopid!).
- Read EDID to determine support for HDR, and colorimetry info to expose via `DXGIOutput6::GetDesc1`
 - Using my bodged `libdisplay-info` that compiles for Windows.

Supporting HDR is a lot of work

HDR from DXVK/VKD3D-Proton side (DXGI Swapchain)

- Enable `VK_EXT_swapchain_colorspace` for HDR10/scRGB colorspaces
 - PQ: `VK_COLOR_SPACE_HDR10_ST2084_EXT`
 - scRGB: `VK_COLOR_SPACE_EXTENDED_SRGB_LINEAR_EXT`
- Forward HDR metadata using `VK_EXT_hdr_metadata`
 - Convert from weird MS format/scaling for metadata to Vulkan (nits in floats)
 - Not the same as HDMI CTA 861.G wire format! (Microsoft samples were broken!)
 - `vkSetHdrMetadataEXT` on the swapchain

Supporting HDR is a lot of work

Supporting NVAPI/AMD AGS HDR for older games:

These (terrible) old vendor APIs don't work in terms of buffer/swapchain tagging. Instead they work by setting the entire screen to 'PQ' and still sending content untagged. Everything SDR looks broken on Windows.

I opted to implement this differently, making it so when an app uses this, any swapchains they have automatically get tagged as PQ.

AMD AGS is especially painful as it has no stable ABI.

Have to manage some horrible global state using a DXVK extension



PHEW!



But, *Josh*? How can we do HDR when Gamescope only supports X11 clients?





But, *Josh*? How can we do HDR when Gamescope only supports X11 clients?

Isn't Wine/Proton still targeting X11?



I don't know



I don't know

Thanks for coming to my talk. Bye.

Gamescope WSI Layer!



Who needs XWayland anyway?

1. Create a Wayland surface in the layer
2. Tell Gamescope to replace an X11 window with our own surface
3. Present to our Wayland surface
4. ???
5. Profit



Other benefits of the WSI layer!

- Lower latency! (No XWayland!)

Other benefits of the WSI layer!

- Lower latency! (No XWayland!)
- Not limited by X11 visual IDs...

Other benefits of the WSI layer!

- Lower latency! (No XWayland!)
- Not limited by X11 visual IDs...
- Can implement and iterate on *anything* we want!



A flawless plan!

A flawless plan!

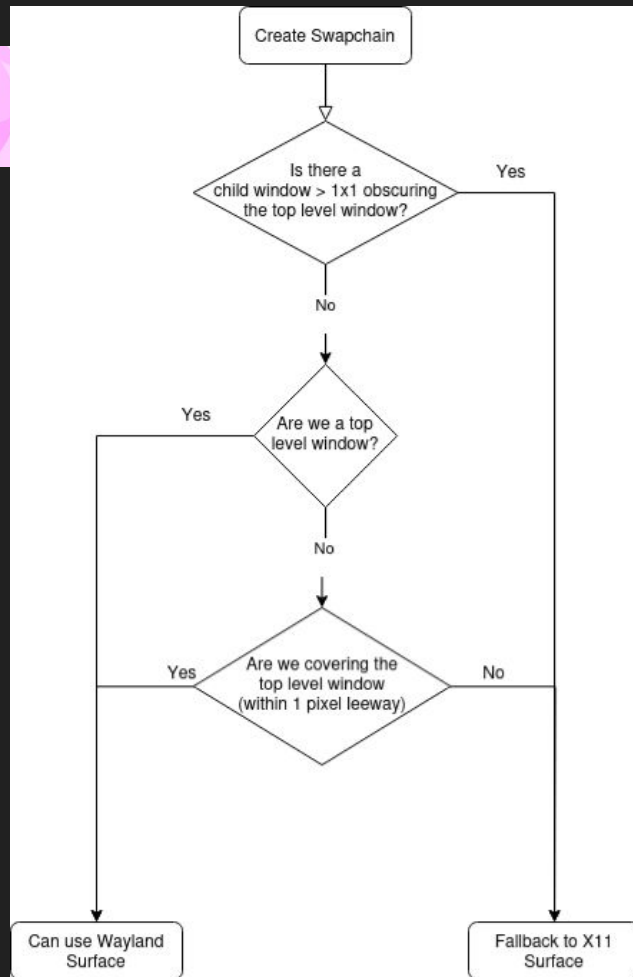
*aside from child windows



Solving child windows

- If we are a top level window, check if there are any valid (not 1x1) child windows and fallback to the X11 surface if so.
- If we are a child window, check if any others are covering, and fallback to the X11 surface.

It is not a perfect or “compliant” solution, but it is good enough! (for now at least :D)



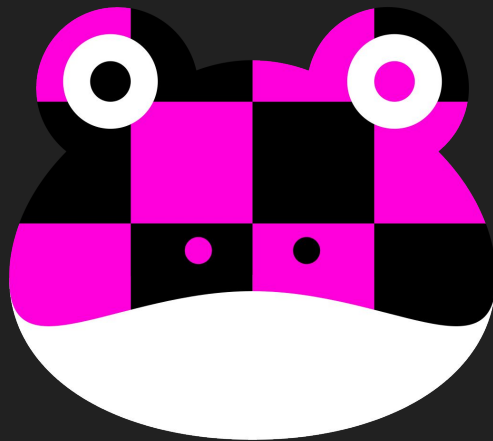
Now what?

We need to talk to the compositor!

- Upstream Wayland colour management/HDR efforts are very slow.
- Almost 4 years old with little progress on implementation. :(
- *Lots of spec arguing and bikeshedding.*

...

- Just going to do my own thing for now...



***gamescope_swapchain* protocol**

- Custom protocol (*gamescope_swapchain*) to talk to the compositor

***gamescope_swapchain* protocol**

- Custom protocol (*gamescope_swapchain*) to talk to the compositor
- It implements the X11 window replacement logic (*override_window_content*)

gamescope_swapchain protocol

- Custom protocol (*gamescope_swapchain*) to talk to the compositor
- It implements the X11 window replacement logic (*override_window_content*)
- Simplest thing ever, literally sends *VkColorSpaceKHR* enums over Wayland wire.

gamescope_swapchain protocol

- Custom protocol (*gamescope_swapchain*) to talk to the compositor
- It implements the X11 window replacement logic (*override_window_content*)
- Simplest thing ever, literally sends *VkColorSpaceKHR* enums over Wayland wire.
- Could be smarter and do protocol enums... but no reason yet.

gamescope_swapchain protocol

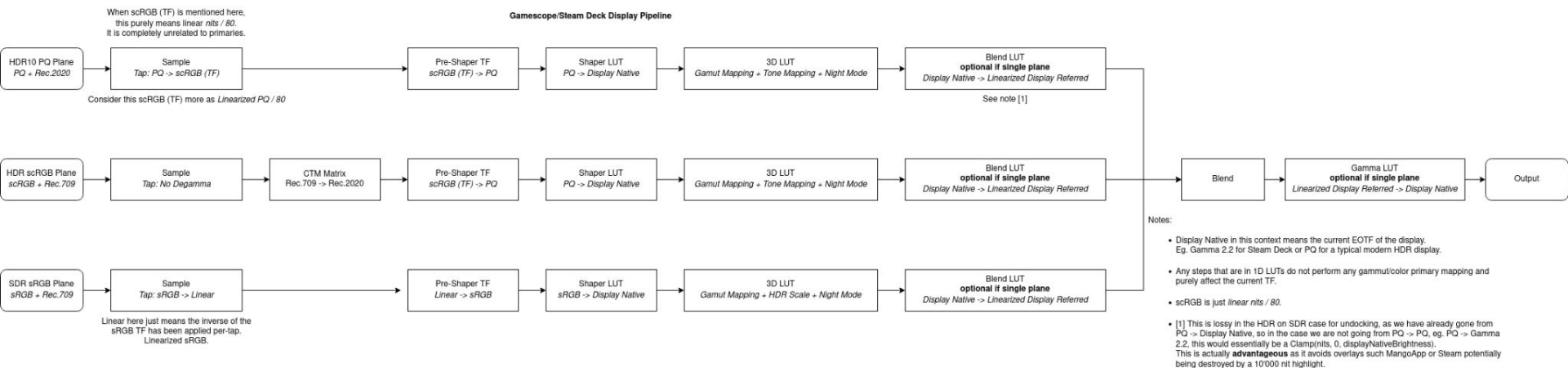
- Custom protocol (*gamescope_swapchain*) to talk to the compositor
- It implements the X11 window replacement logic (*override_window_content*)
- Simplest thing ever, literally sends *VkColorSpaceKHR* enums over Wayland wire.
- Could be smarter and do protocol enums... but no reason yet.
- Convert and send HDR metadata in CTA 861.G format on wire.
 - tl;dr is that this is just fixed point vs float (vulkan)
 - also HDR metadata sucks anyway lol

gamescope_swapchain protocol

- Custom protocol (*gamescope_swapchain*) to talk to the compositor
- It implements the X11 window replacement logic (*override_window_content*)
- Simplest thing ever, literally sends *VkColorSpaceKHR* enums over Wayland wire.
- Could be smarter and do protocol enums... but no reason yet.
- Convert and send HDR metadata in CTA 861.G format on wire.
 - tl;dr is that this is just fixed point vs float (vulkan)
 - also HDR metadata sucks anyway lol

<https://github.com/ValveSoftware/gamescope/blob/master/protocol/gamescope-swapchain.xml>

Color Pipeline



Future Work: Kernel uAPI Needs



Right now we are using AMD Private DRM props that Melissa and I PRed.

Need proper color management uAPI

- Neeeeeed 3x4 CTMs!
- Splitting up setting and enabling of LUTs.

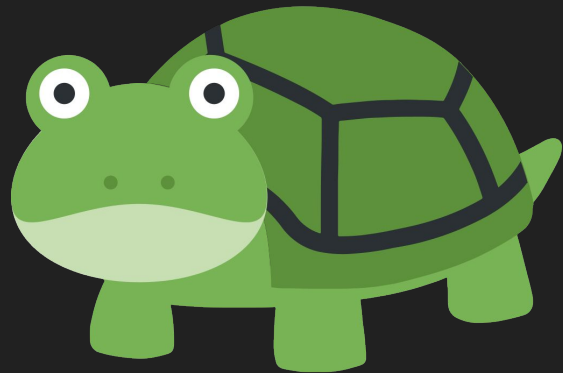
Future Work: Kernel uAPI Needs



Right now we are using AMD Private DRM props that Melissa and I PRed.

Need proper color management uAPI

- Neeeeeed 3x4 CTMs!
- Splitting up setting and enabling of LUTs.



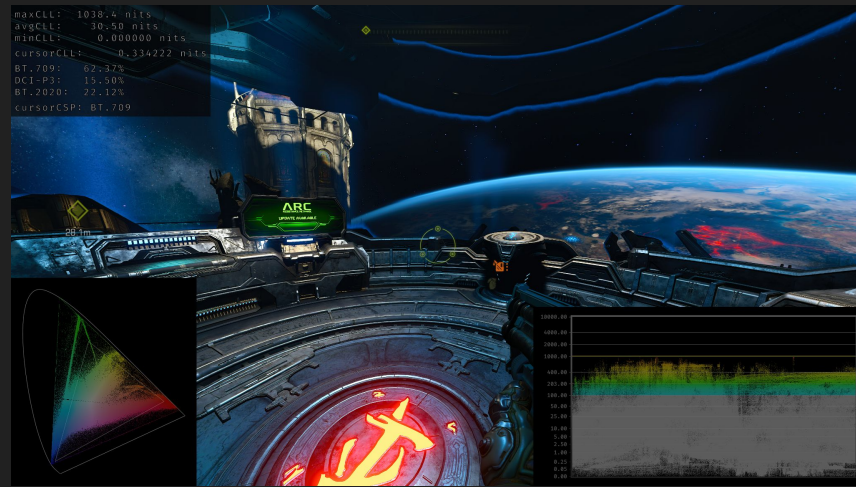
● **LIVE:** Image of updating 3D LUT on AMDGPU

Tools for HDR Developers and HDR Debugging

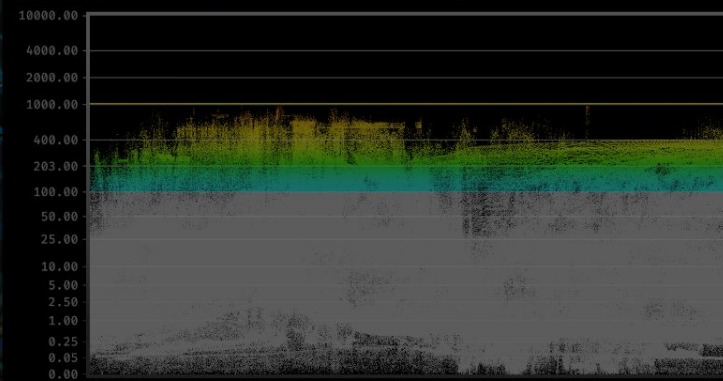
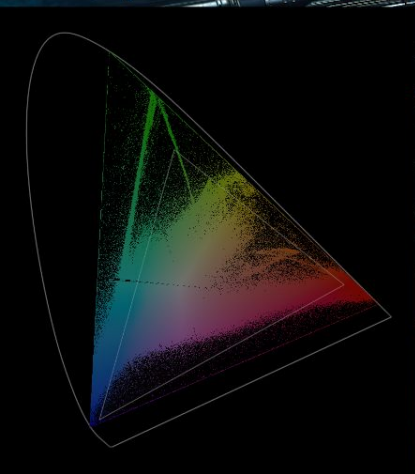
Gamescope/SteamOS has lots of tools to help you investigate HDR apps!

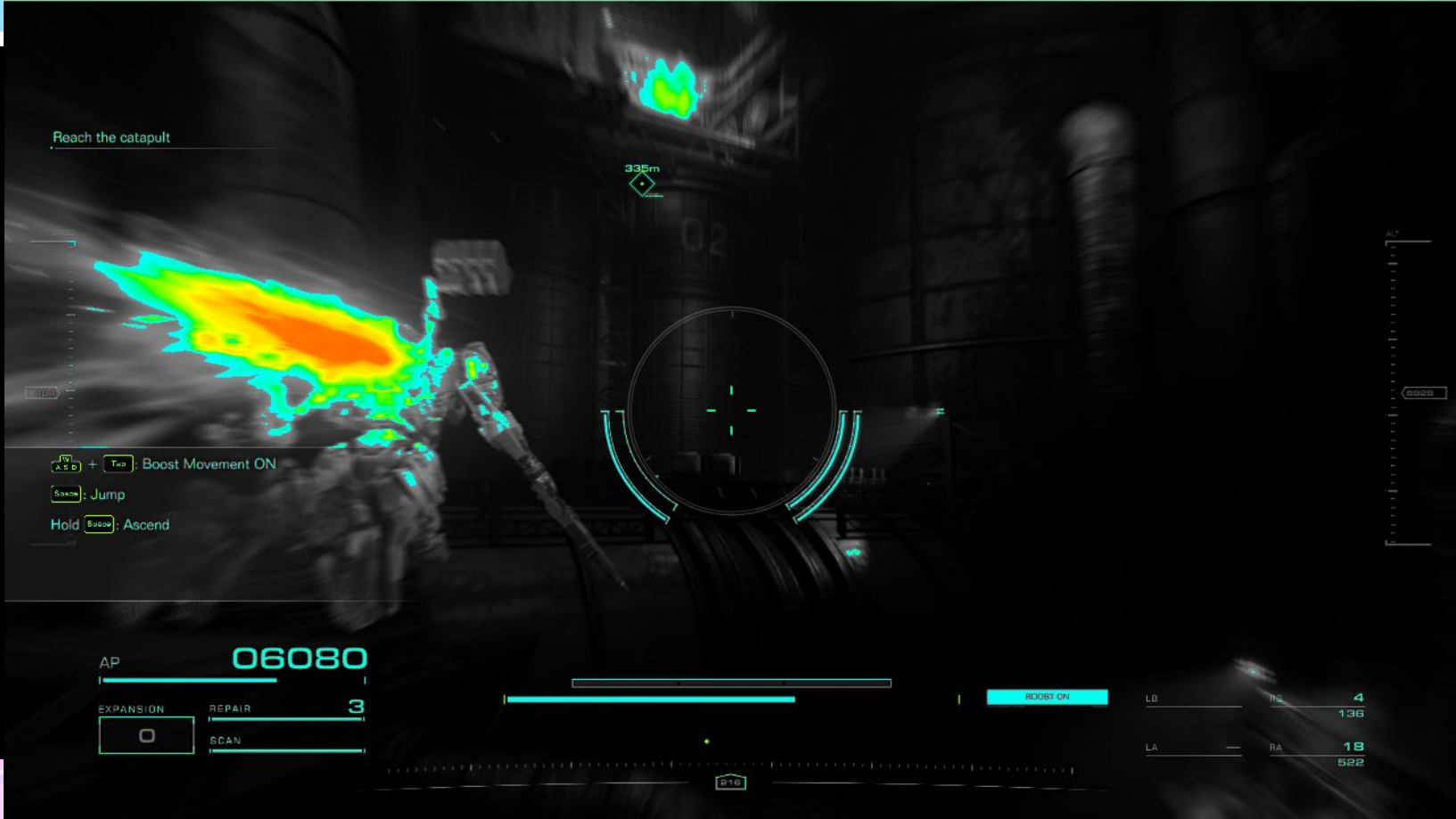
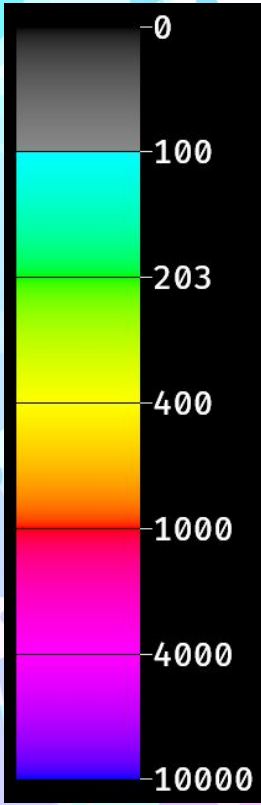
All of these are in SteamOS Developer Options today!

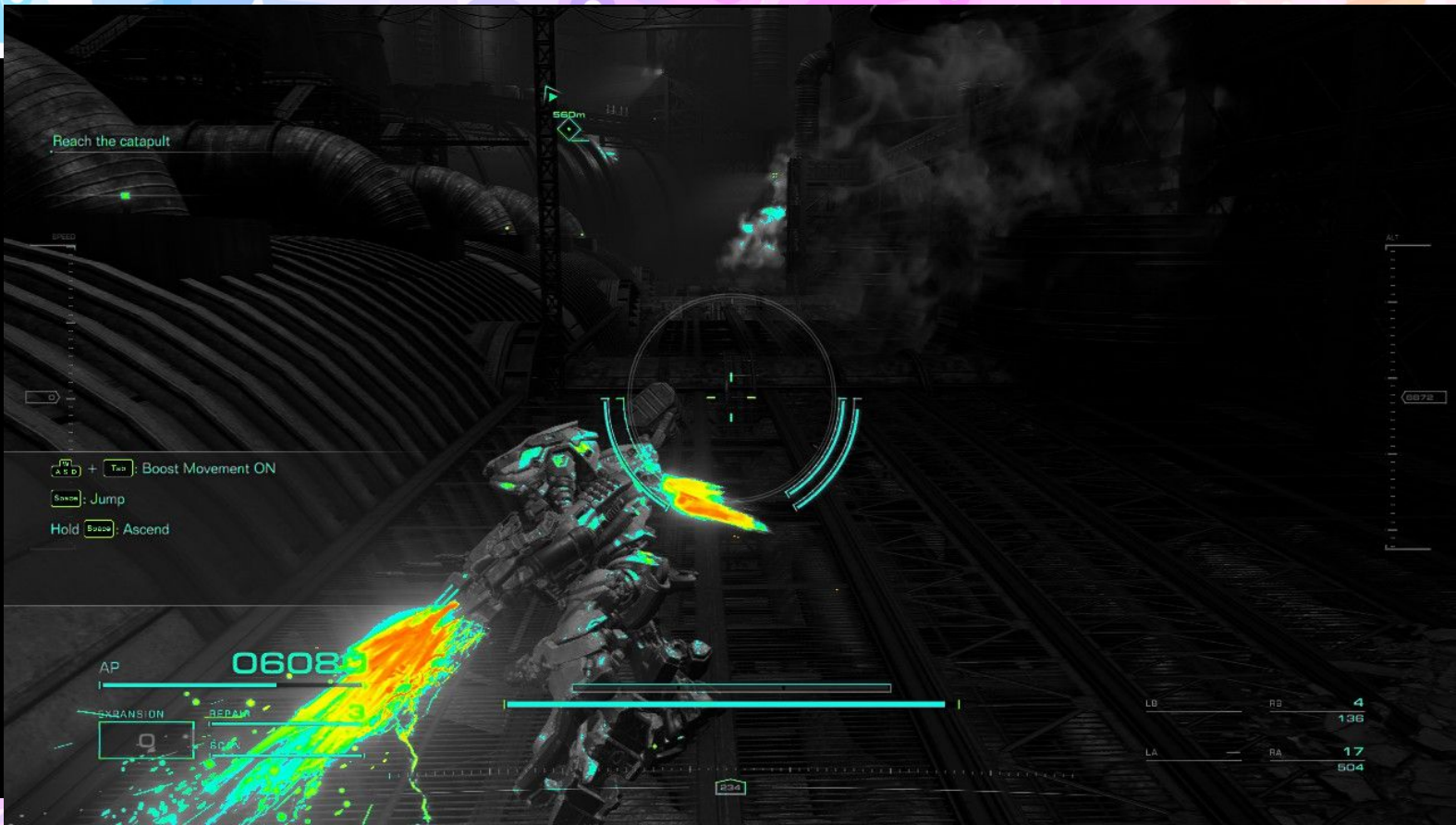
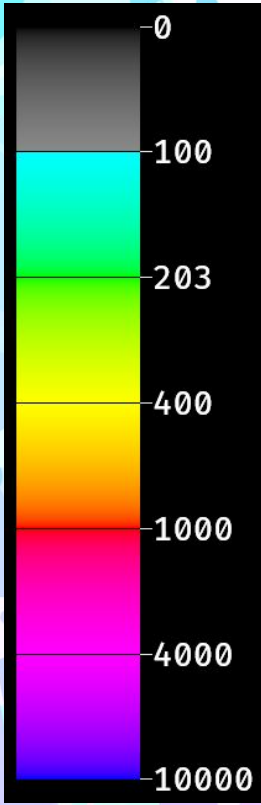
- HDR Heatmap for both Lilium + MSWCG targets.
- Full HDR Analysis suite by Lilium available in Developer Options



maxCLL: 1038.4 nits
avgCLL: 30.50 nits
minCLL: 0.000000 nits
cursorCLL: 0.334222 nits
BT.709: 62.37%
DCI-P3: 15.50%
BT.2020: 22.12%
cursorCSP: BT.709









Thanks for coming!

Questions?

e-mail me: joshua@froggi.es

special thanks: harry wentland, melissa wen, jeremy selan, lilium