

Open in app ↗

Sign up

Sign in



Search



A guide to a good docked gaming experience on the Steam Deck



Wasib · Follow

7 min read · Nov 30, 2022



Listen



Share

The Steam Deck is a great handheld device that handily beats anything and everything within its class. But what about a good docked gaming experience? Does the tiny 15w APU have the power to deliver a good docked gaming experience on a 4k TV? These were the questions I had set out to answer when I was planning to use my Steam Deck as my only computer for a while.

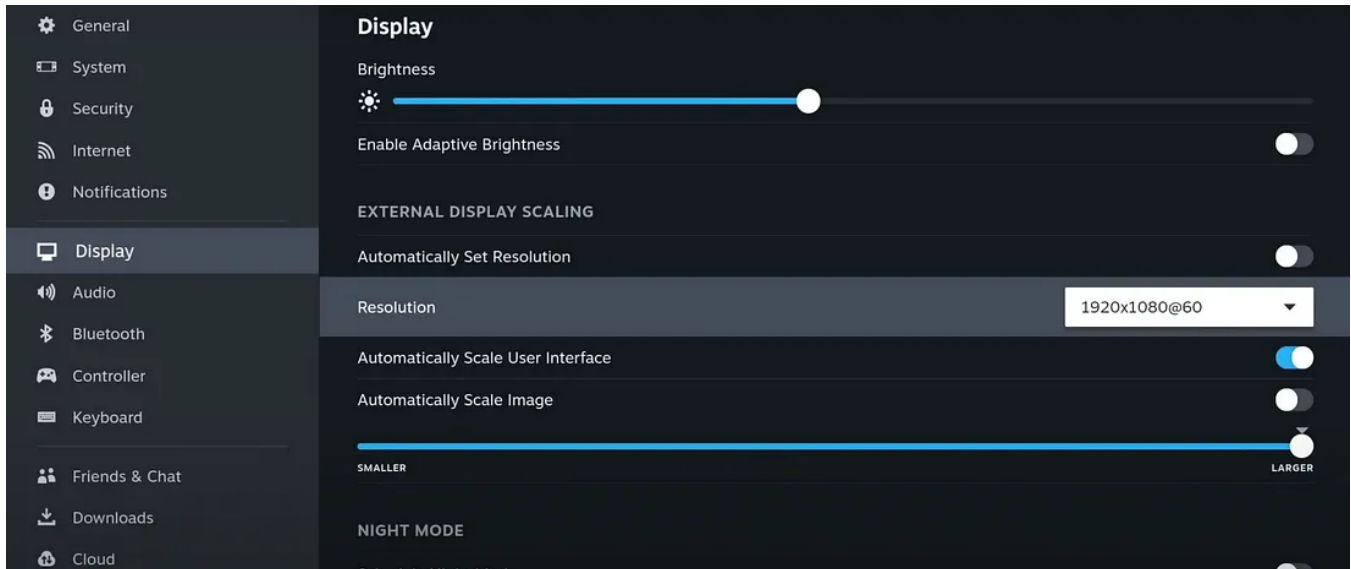
To answer it straight : No, there's no way the tiny 15w RDNA2 APU can output games at Native 4k. But it does not have to. Thanks to the latest innovations in upscaling technologies like FSR 2.0, FSR 1.0, Temporal upscalers like Unreal's TAAU/TSR, we are now able to get a very good visual experience even at a fraction of the actual native resolution.

The steam deck is a 40hz/40fps machine at its native resolution of 800p when it comes to newer, demanding games. It can handle older games fairly well at resolutions like 1080p, 1440p and even 4k in certain cases.

So our goal here would be to render games at the same resolution of 720p and try and upscale them to 4k while maintaining a good image quality with minimal blur. And coming to framerate, we would be targetting 30fps just like many of the consoles out there given our performance constraints.

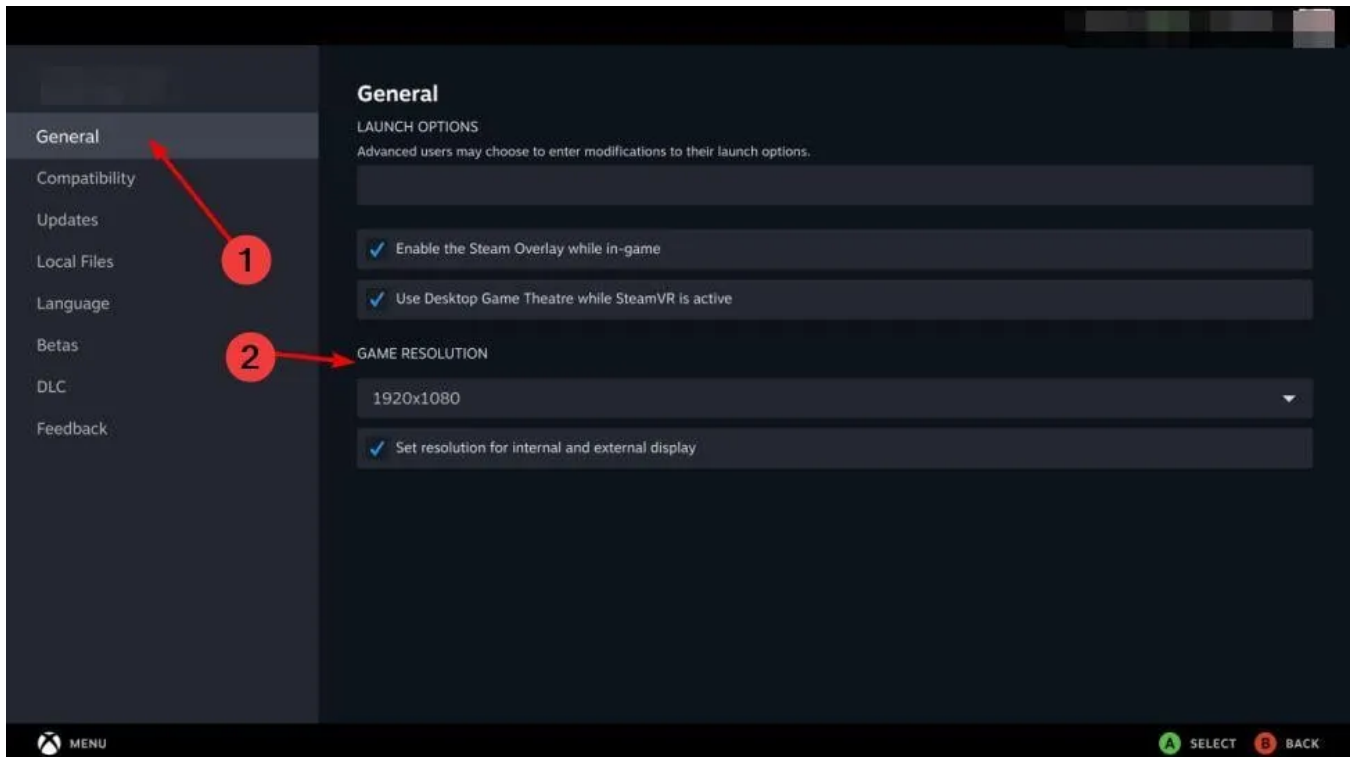
Before jumping into the games, I would want to bring up the various set of places where a resolution can be set within the game mode of SteamOS. There are two different places where we can set the output resolution of the steam deck and both of them behave differently.

The resolution option within game mode's display settings:



Lets call this one the Gamescope Output Resolution

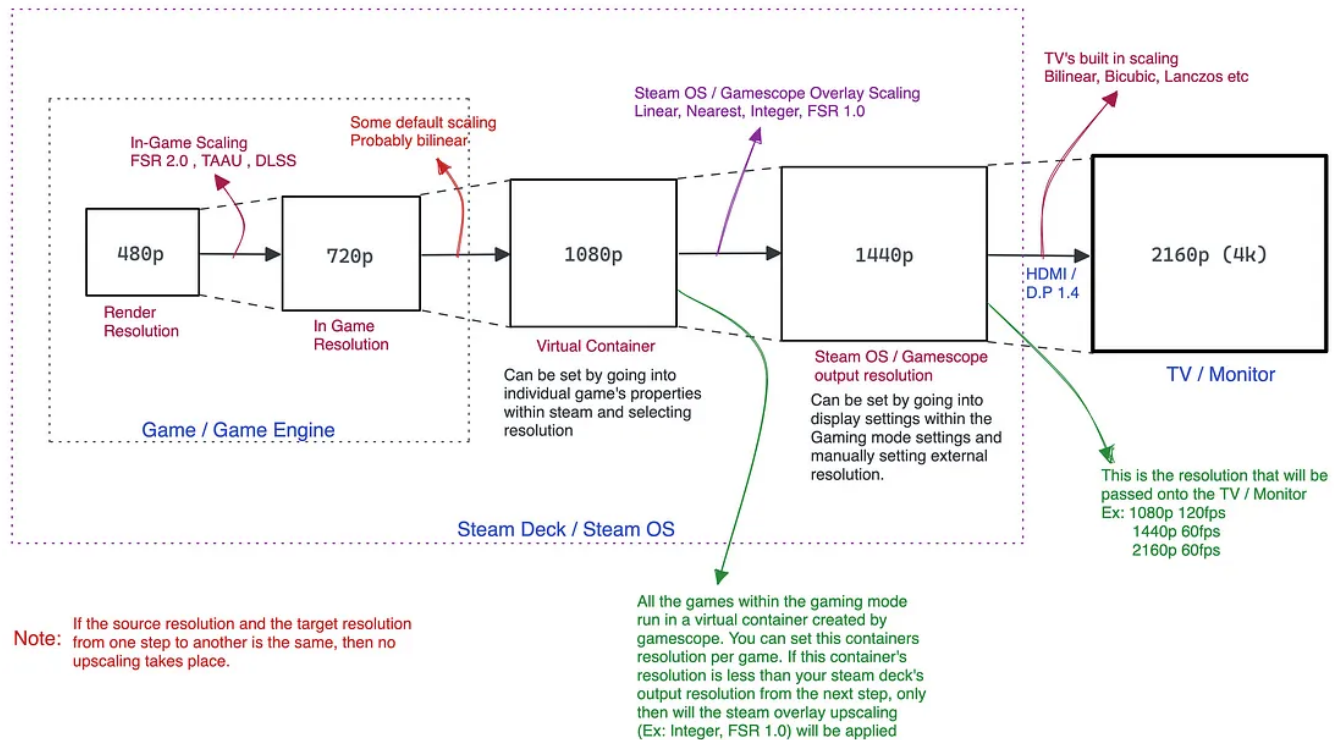
The resolution that can be set within individual game's properties by clicking on the gear icon:



Lets call this one the Virtual Container Resolution

The gamescope output resolution is the resolution that is sent to the monitor or the TV. If you set this to 1080p 120fps, the TV will receive a 1080p 120fps signal. Coming to the second setting — the Virtual Container, for some reason Gamescope creates a virtual container and runs the game within it and it then upscales or downscales the virtual container to match the resolution of your output resolution. Yes, you can set the virtual container to run at even higher than native resolutions and then downscale them to match your native resolution (supersampling). The upscaling between the virtual container and the output resolution happens according to the algorithm you have selected within the steam overlay (brought up by pressing the ‘..’ button).

Apart from these two places, you can also set the resolutions within your games, after starting them. These are the in game resolution and the internal rendering resolution that we are all aware of. Here is where all the temporal upscaling algorithms come into play. I have summarised all these different resolutions into a handy diagram representing the order of upscaling that happens.



All the different possible places where the resolution can be set

Now that we have a good understanding of how and where the steam deck handles all the upscaling, we can make intelligent decisions to render our games at smaller resolutions and efficiently upscale them using techniques like Integer scaling, FSR 1.0 without causing any blur usually caused by other algorithms.

From my experience the best place to upscale would be the rendering resolution of the game using techniques like FSR 2.0, TAAU/TSR in games which support them and mods like CyberFSR in games which do not have temporal upscaling built in. I would not suggest using ingame FSR 1.0 upscaling as I have found the steam deck's built in FSR to be doing a better job at that.

The cost of these upscalers is also a factor to be considered.

Ingame — **FSR2.0** costs around 10fps in performance.

Ingame — **TAAU/TSR** costs around 5fps in performance.

Ingame — **FSR1.0** costs around 5fps in performance.

Ingame — **Simple upscaling**(bilinear) costs nothing but is the worst.

SteamOS Upscaling — **FSR** costs the highest at around 5-10fps in performance but tends to generate results sharper and better than Integer scaling with no blur.

SteamOS Upscaling — **Integer** looks the next best at integer scaling resolutions (1:2, 1:3) and costs nothing.

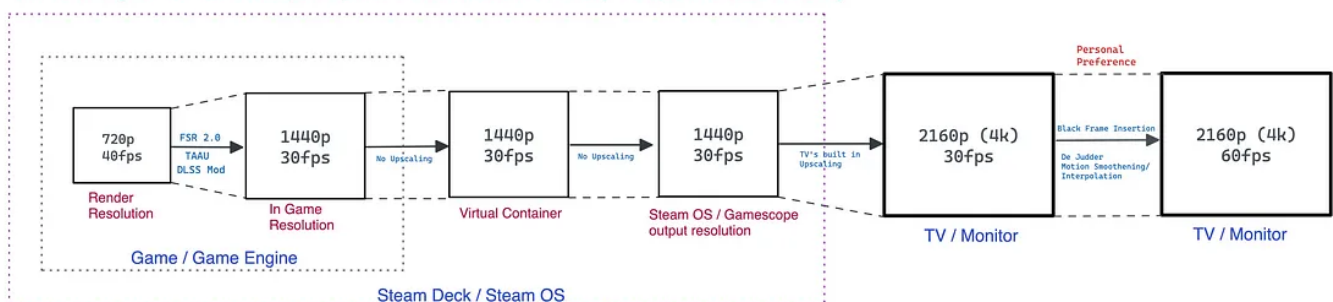
SteamOS Upscaling — **Nearest** is the same as Integer at integer resolutions, but not as good at non integer resolutions. Cost is minimal too.

SteamOS Upscaling — **Linear** looks the worst and introduces blur at low resolutions, but costs nothing.

Now that we know how and where the images are upscaled and the cost of the same, lets take a look at a few recommendations I landed for myself after extensive testing. These tend to produce the best looking visuals while playing on a 4k TV and in many cases, I couldnt believe that all this was running on that tiny handheld. It was a smooth 4k 30fps experience in many cases simulating the performance of a PS4.

Case 1: I am currently following this approach for games like Horizon Zero Dawn (using CyberFSR), God of War (TAAU), and Spiderman Remastered (FSR 2.1). All these games run well at 40fps on the deck natively and when upscaled smartly, we can still get a very good looking output while still maintaining a locked 30fps at 4k.

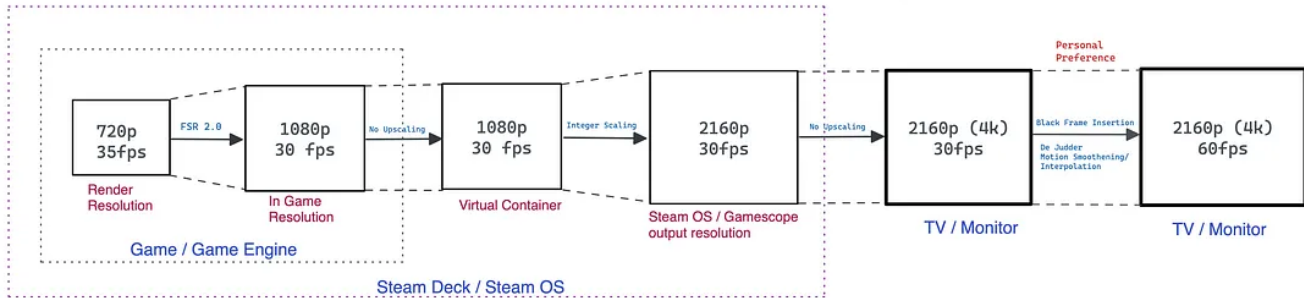
Case 1: If a game has built in temporal upscalers and runs at atleast 720p \geq 40fps on the deck natively



Case 1: If a game has built in temporal upscalers and runs at atleast 720p \geq 40fps on the deck natively

Case 2: If the game struggles to hit consistent 30fps after temporal upscaling to 1440p (FSR 2.0 performance mode), try rendering at 1080p and use FSR 2.0 Quality mode instead. Then upscale it using integer scaling to 4k (1:4) or using FSR 1.0 from the deck if you have some more performance to spare.

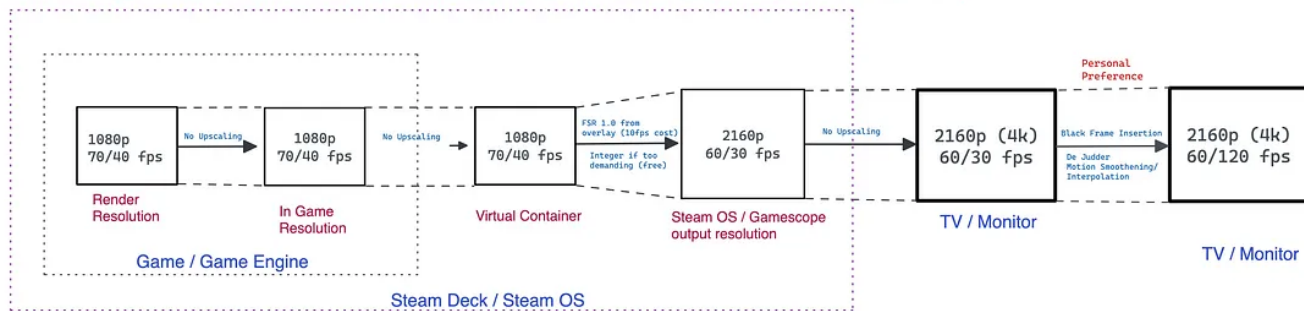
Case 2: If a game has built in temporal upscalers and runs at atleast 720p >30fps & <40fps on the deck natively



Case 2: If a game has built in temporal upscalers and runs at atleast 720p >30fps & <40fps on the deck natively

Case 3: If it is an older game and does not have built in FSR 2.0 or similar, but can run fine at 1080p, then Deck's FSR 1.0 or Integer scaling(1:4) is your friend.

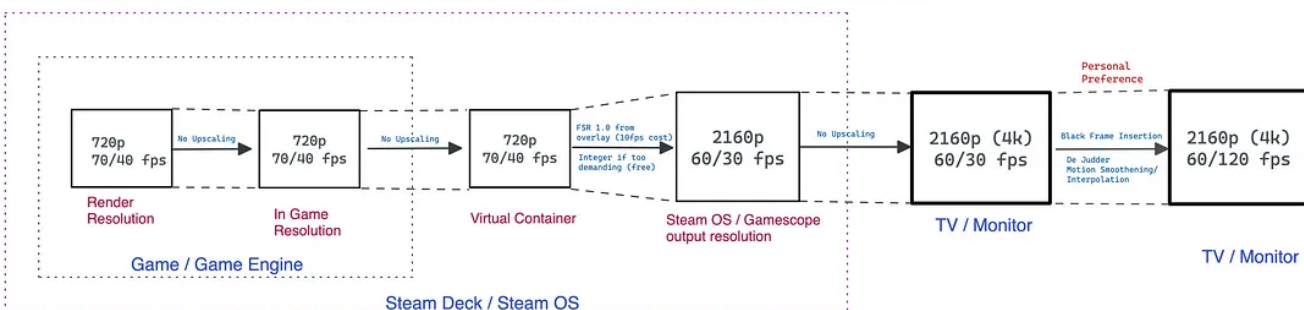
Case 3: If a game does not have built in temporal upscalers and runs at atleast 1080p ≥40fps on the deck natively



Case 3: If a game does not have built in temporal upscalers and runs at atleast 1080p ≥40fps on the deck natively

Case 4: If it is a more demanding game without temporal upscaling, running it at 720p and scaling it using Deck's FSR or Integer scaling(1:9) still produces fairly good looking results for a 4k TV from a few feet away.

Case 4: If a game does not have built in temporal upscalers and runs at atleast 720p ≥40fps on the deck natively



Case 4: If a game does not have built in temporal upscalers and runs at atleast 720p >=40fps on the deck natively

The above cases do not take into account all the possible scenarios, all tv/monitor resolutions, but can help act as a source of reference for whoever is trying to run games on their docked steam deck. You can tweak and play around and find what works best for you.

Other tips i use to extract the most performance out of my Deck:

1. Use [CryoByte33's](#) VRAM tweaks and Swap file tweaks to increase Vram to 4gb and Swap file to 16gb with swappiness set to 1. This increases minimum framerates in many games.
2. Try different Proton versions including ProtonGE and also by setting `DXVK_ASYNC=1` in the game's launch commands.
3. Set the Texture setting in game to "High" and Ambient Occlusion to "Medium". These generally tend to give the most bang for the buck in terms of display quality while costing very little performance. The steam deck has a good enough vram buffer to handle high textures.
4. Lock the framerate to 30 using steam overlay and disable any ingame vsync. If you are on steamOS 3.4 or above, enable the "Allow Tearing" option in the overlay to reduce the input lag by 16ms.
5. Many UE4 games support TAAU but they do not show it on the game settings. You can enable this by manually entering this line in the DefaultEngine.ini file of the game — `r.TemporalAA.Upsampling=1 r.screenPercentage=50`
6. Use the [CyberFSR mod](#) to add FSR 2.0 in games that do not have native support for FSR2.0 — Their quality is sometimes even better than the ingame FSR2 implementation.
7. In games that are CPU heavy, try to give CPU some more power budget by

locking the GPU frequency to a lower value like *1200mhz*.

8. If you have a OLED or a newer LCD TV, use the motion interpolation (dejudder) and Black Frame Insertion (BFI) to double the output visual fps. Not everyone prefers this but it can makes 30fps output much smoother and platable which introducing a tiny amount of input lag and minor artifacting. A worthy exploration if you want to get the smoothest experience out of your deck. [Here is a good tutorial/example of that.](#)
9. If you are on a TV or Monitor that supports 120hz, and want to set the fps limit to 40, but cannot find the 40fps option in the overlay slider, then you can set this command in the steam game launch options to lock it to 40fps.
mangohud MANGOHUD_CONFIG=fps_limit=40,no_display %command%

Let me know if you have any other tips and I can add them to the list above.

Thank you for reading through the entire post. I will make a follow up to this one by capturing some screenshots of actual gameplay and showing how they appear with all the different upscaling methods and how they affect the final image quality.

Steam Deck

Gaming

Hardware

Linux

Image Upscaling



Follow