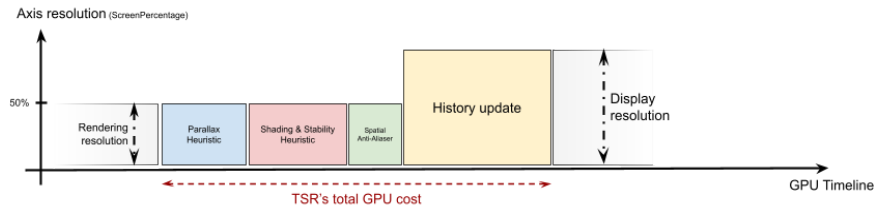
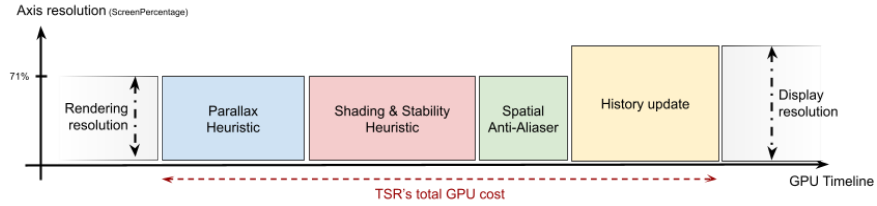


TSR's Upscaling GPU Cost

The primary goal of TSR is upscaling. The majority of its GPU work scales based on the resolution it is given. That is due to some of the GPU cost of TSR needing to be done at a higher display resolution than the rendering resolution.



Open the GPU statistical display from the console using `stat GPU`. With this open, you can adjust the primary Screen Percentage to see the difference in performance when rendering at 100 percent versus 50% screen percentage, like the examples taken below from the Valley of the Ancients sample project.



~0.79ms at r.ScreenPercentage=100	~0.43ms at r.ScreenPercentage=50
<i>Click the image for full size.</i>	<i>Click the image for full size.</i>

The parallax heuristic of TSR depends on the depth and velocity buffer rather than the scene color and translucencies of the frame because the depth and velocity buffers often finish on the GPU much earlier than the scene color and translucency buffers do. This allows the entire TSR parallax heuristic to compute asynchronously on the GPU, filling in the gaps where the GPU is underused by other rendering algorithms with `r.TSR.AsyncCompute=2`.

In Fortnite Chapter 4 on PlayStation 5 and Xbox Series X, this is offsetting approximately 0.5ms of TSR's total GPU cost when testing performance over an entire Battle Royal performance replay. In the replay, about 0.1ms is saved, bringing down the effective TSR cost to 1.5ms and critical path GPU cost to finish rendering the frame to 1.1ms.

