Previous generation – World War Z

	Render resolution	Target FPS
Xbox One / PS4	960x1080 – 1920x1080	30
Xbox One X	2880x2160 – 3840x2160	30
PS4 Pro	1920x1080 – 3200x1800	30

- We used fixed-height dynamic resolution on consoles
- TAA as an antialiasing solution without upscaling on consoles
- FXAA for low-preset settings on PC

- Much later we integrated FSR2 in the end of the frame, and it was a starting point of our AA/upscale journey on newer generation

Platforms

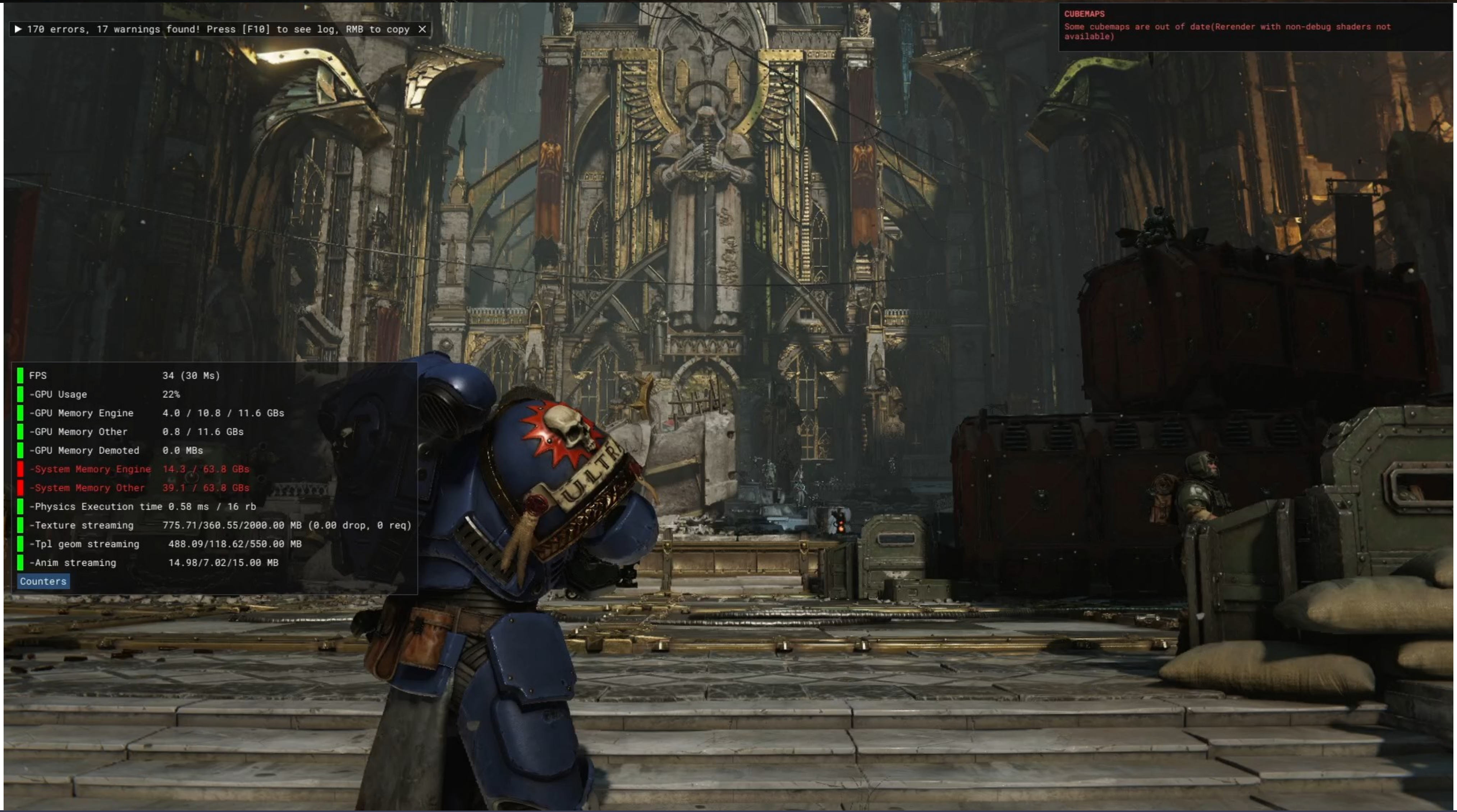
We use much more temporal techniques on different platforms now:

	FSR2	FSR3	FSR4	XESS*	DLSS	MFSR	TAA + FSR1
PC	✓	✓	AMD RDNA4	✓	NVIDIA 2000+ series		✓
XBOX SERIES X/S	✓						
PS5	✓						
PS5 PRO						✓	
Steam Deck	✓	✓					✓

TAA + FSR1 is intended as a cheap fallback solution. In practice, TAA is mostly used as an auxiliary temporal pass for several resources like circle of confusion, bloom mask, etc.

*XeSS is only in RoadCraft for now

Why we don’t use FXAA

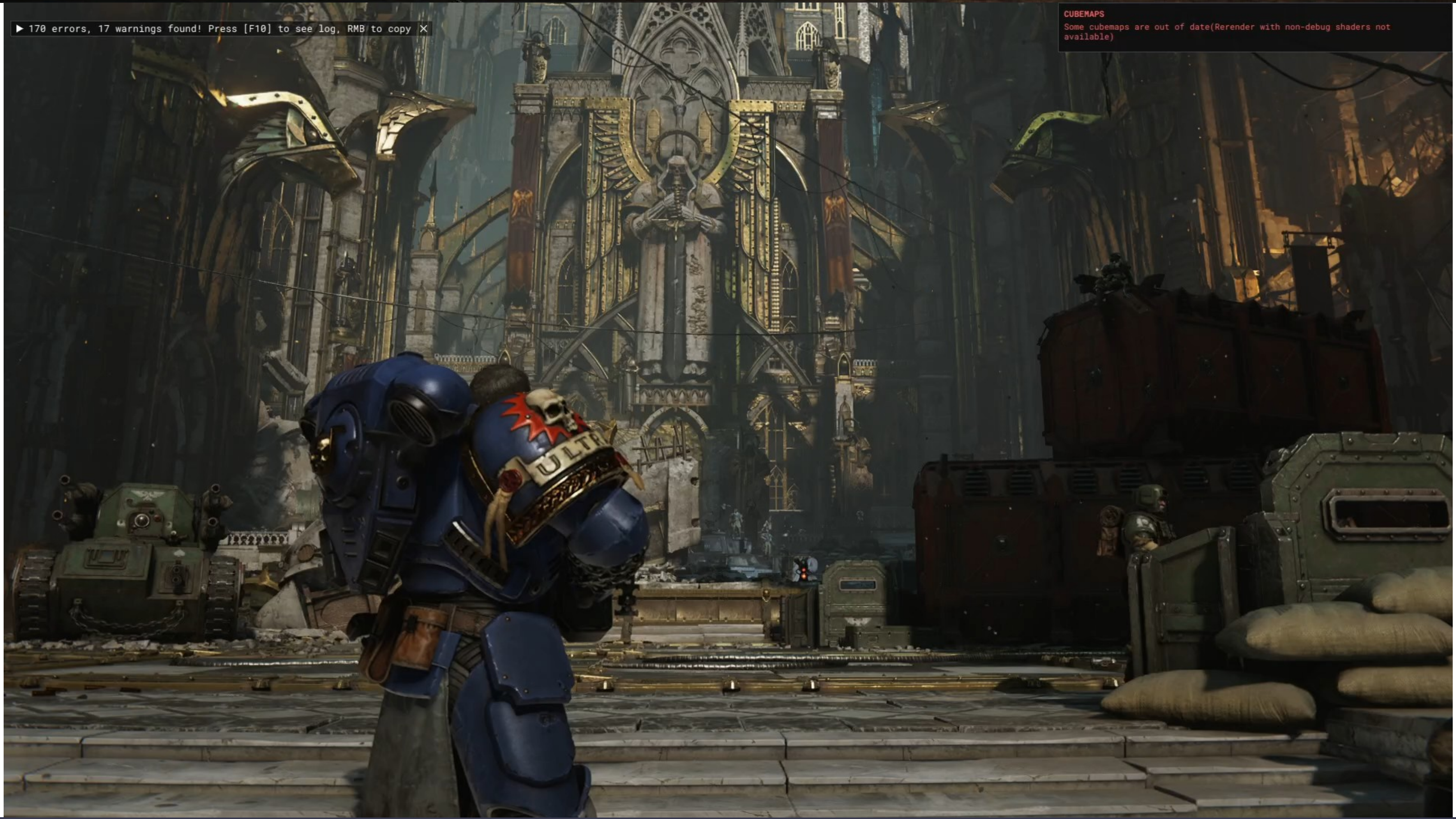


```
▶ 170 errors, 17 warnings found! Press [F10] to see log, RMB to copy ✕
```

```
CUBEMAPS
Some cubemaps are out of date(Rerender with non-debug shaders not
available)
```

```
FPS                     34 (30 Ms)
-GPU Usage              22%
-GPU Memory Engine      4.0 / 10.8 / 11.6 GBs
-GPU Memory Other       0.8 / 11.6 GBs
-GPU Memory Demoted     0.0 MBs
-System Memory Engine   14.3 / 63.8 GBs
-System Memory Other    39.1 / 63.8 GBs
-Physics Execution time 0.58 ms / 16 rb
-Texture streaming      775.71/360.55/2000.00 MB (0.00 drop, 0 req)
-Tpl geom streaming      488.09/118.62/550.00 MB
-Anim streaming          14.98/7.02/15.00 MB
Counters
```

TAA



```
▶ 170 errors, 17 warnings found! Press [F10] to see log, RMB to copy ✕
```

```
CUBEMAPS
Some cubemaps are out of date(Rerender with non-debug shaders not
available)
```

Resolution presets

Consoles

	Render resolution	Target resolution	Target FPS
Xbox Series X / PS5 **Quality** mode	1080p – 1440p	2160p	30
Xbox Series X / PS5 **Perf** mode	720p – 1440p	2160p	60
Xbox Series S	720p – 1080p	1440p	30
PS5 Pro **Quality** mode	1080p – 2160p	2160p	30
PS5 Pro **Perf** mode	1080p – 1440p	2160p	60

PC

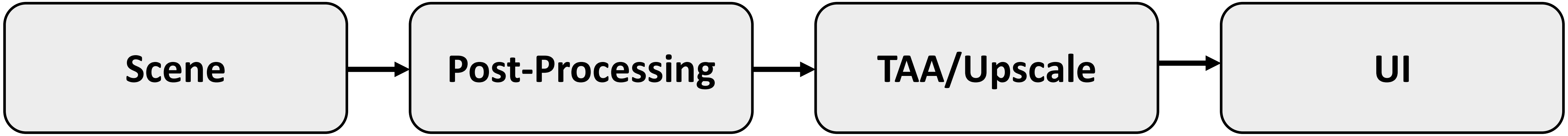
Mode	Resolution Scale
Native	1
Quality	1.5
Balanced	1.7
Performance	2
Ultra performance	3

XeSS

Mode	Resolution Scale
AA	1
Ultra quality	1.5
Quality	1.7
Balanced	2
Ultra performance	3

Temporal AA/Upscale in rendering pipeline

We tried to place it after post-proc like in WWZ, but rejected it



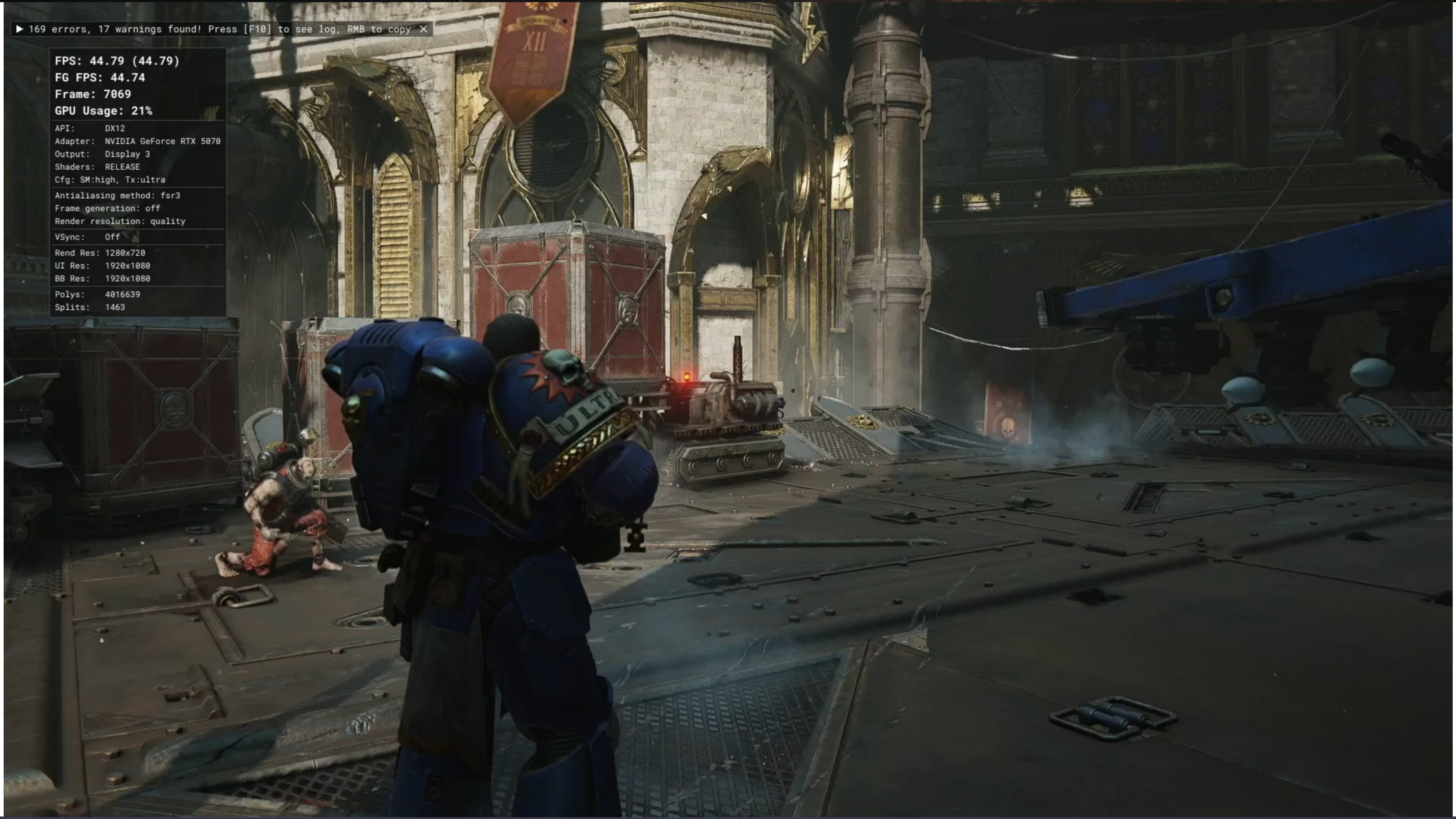
Pros:

- We win some performance in post-processing depending on resolution scale

Cons:

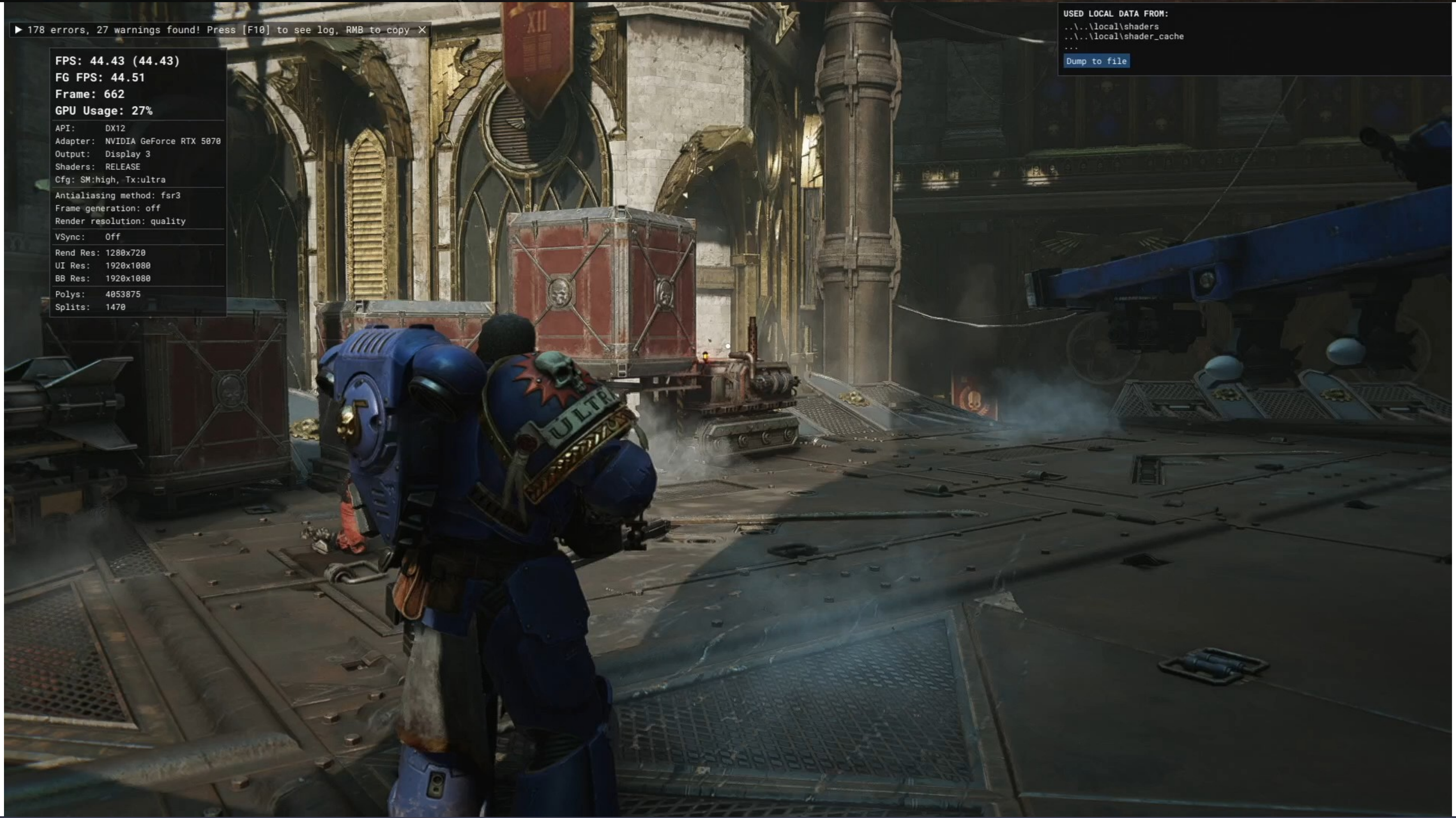
- Post-processing effects such as **motion blur, depth of field, bloom, barrel distortion** change or distort color buffer, so we get ghosting due to mismatch between color and corresponding depth/velocity values.
- Some color details are lost + potential artifacts due to nonlinear LDR input

FSR before post-processing



```
▶ 169 errors, 17 warnings found! Press [F10] to see log, RMB to copy ✕

FPS: 44.79 (44.79)
FG FPS: 44.74
Frame: 7069
GPU Usage: 21%
API:      DX12
Adapter:  NVIDIA GeForce RTX 5070
Output:   Display 3
Shaders:  RELEASE
Cfg: SM:high, Tx:ultra
Antialiasing method: fsr3
Frame generation: off
Render resolution: quality
VSync:    Off
Rend Res: 1280x720
UI Res:   1920x1080
BB Res:   1920x1080
Polys:    4016639
Splits:   1463
```

FSR after post-processing

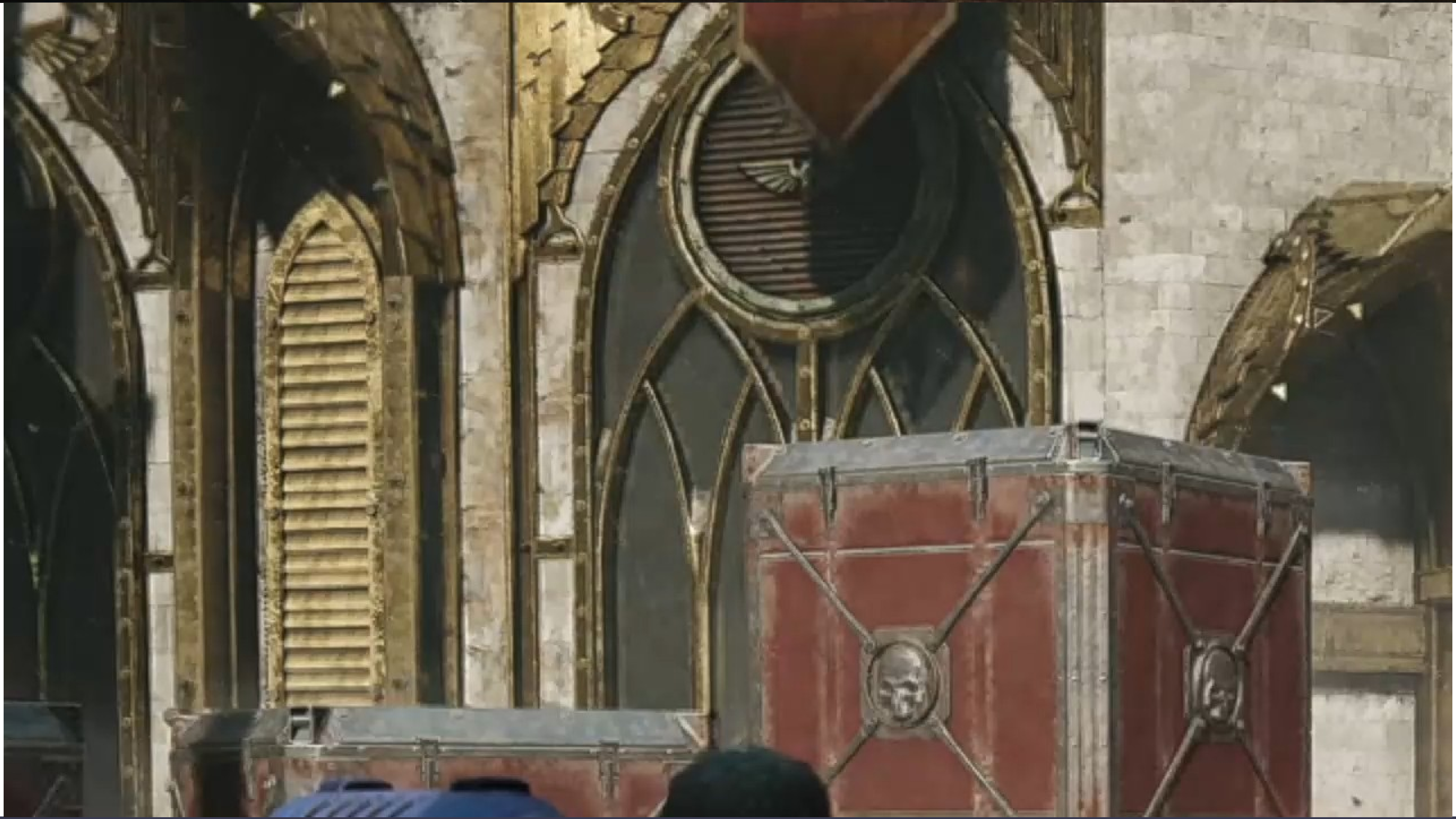


```
▶ 178 errors, 27 warnings found! Press [F10] to see log, RMB to copy ✕
```

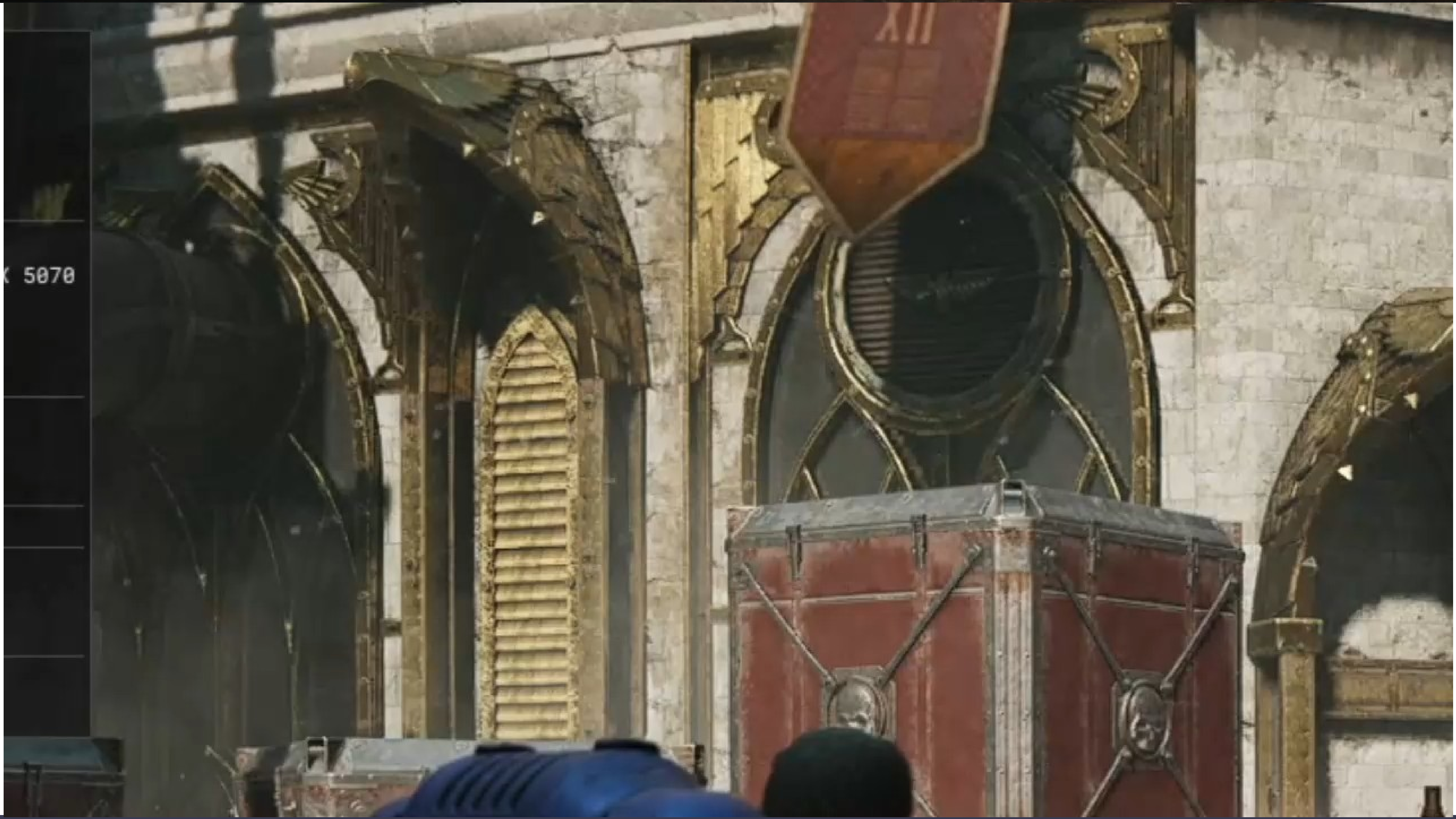
```
FPS: 44.43 (44.43)
FG FPS: 44.51
Frame: 662
GPU Usage: 27%
API:       DX12
Adapter:   NVIDIA GeForce RTX 5070
Output:    Display 3
Shaders:   RELEASE
Cfg: SM:high, Tx:ultra
Antialiasing method: fsr3
Frame generation: off
Render resolution: quality
VSync:     Off
Rend Res:  1280x720
UI Res:    1920x1080
BB Res:    1920x1080
Polys:     4053875
Splits:    1470
```

```
USED LOCAL DATA FROM:
..\..\local\shaders
..\..\local\shader_cache
...
```

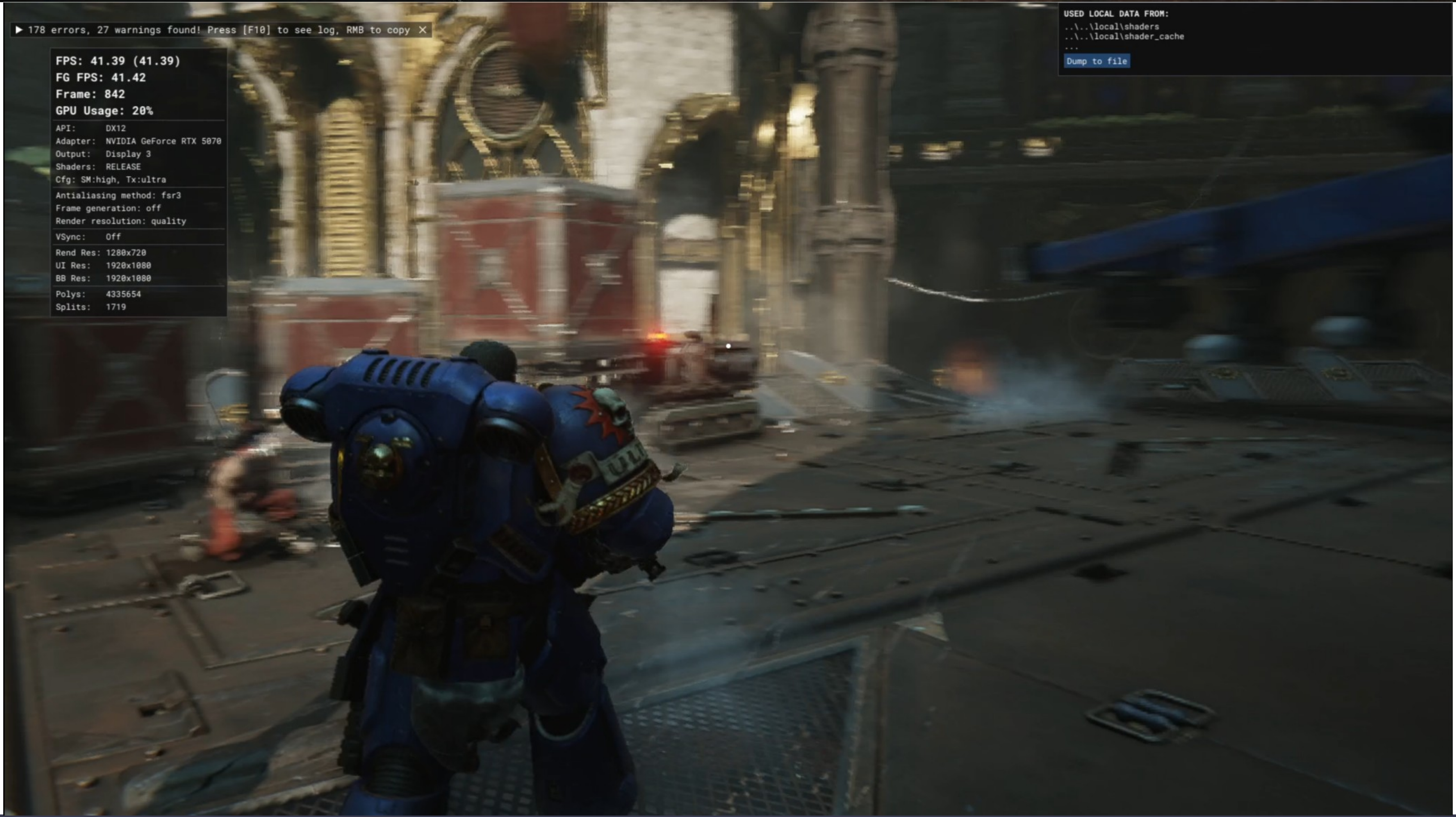
FSR after post-processing



FSR before post-processing



FSR after post-processing

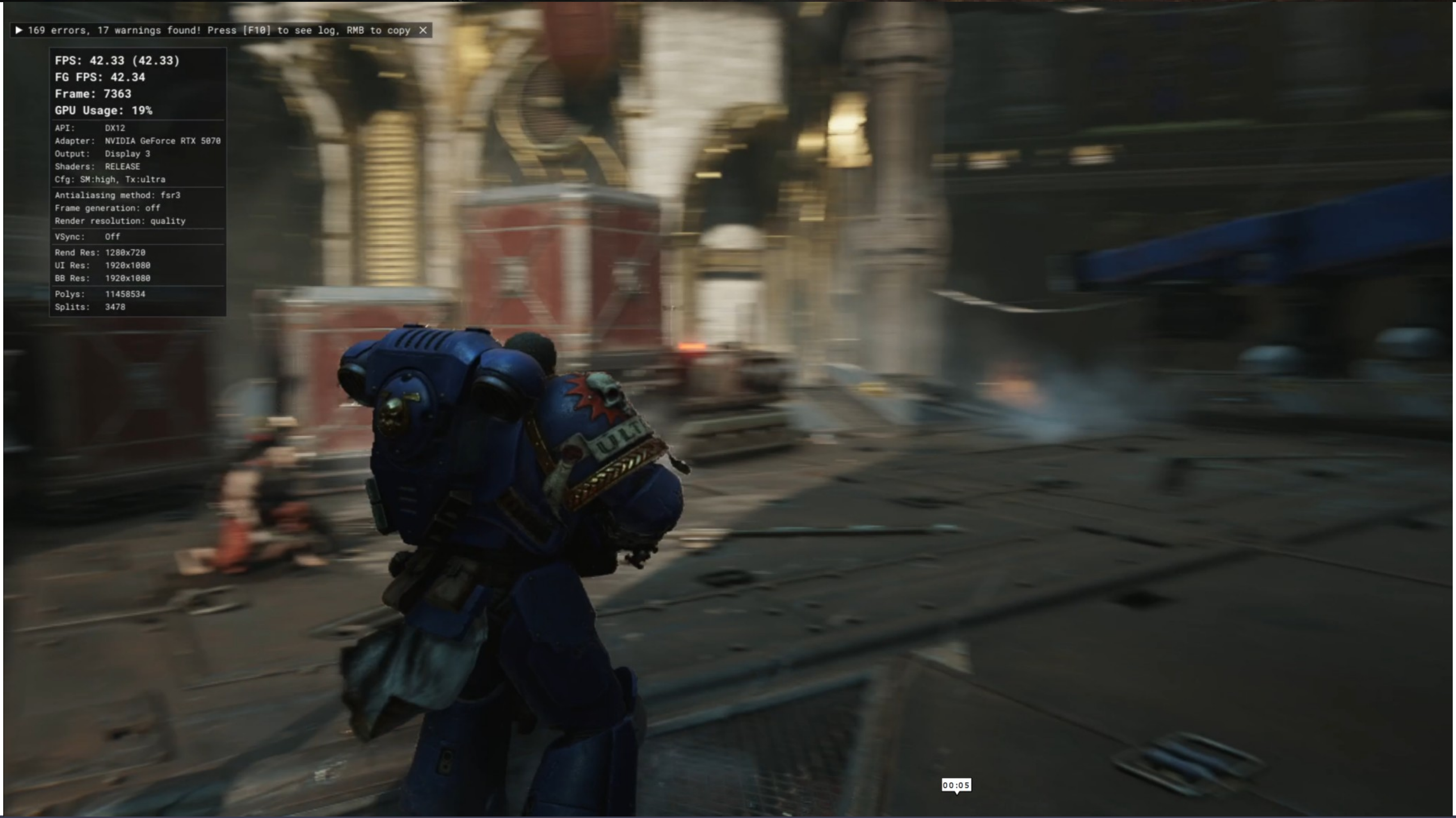


```
▶ 178 errors, 27 warnings found! Press [F10] to see log, RMB to copy ✕
```

```
FPS: 41.39 (41.39)
FG FPS: 41.42
Frame: 842
GPU Usage: 20%
API:       DX12
Adapter:   NVIDIA GeForce RTX 5070
Output:    Display 3
Shaders:   RELEASE
Cfg: SM:high, Tx:ultra
Antialiasing method: fsr3
Frame generation: off
Render resolution: quality
VSync:     Off
Rend Res:  1280x720
UI Res:    1920x1080
BB Res:    1920x1080
Polys:     4335654
Splits:    1719
```

```
USED LOCAL DATA FROM:
..\..\local\shaders
..\..\local\shader_cache
...
```

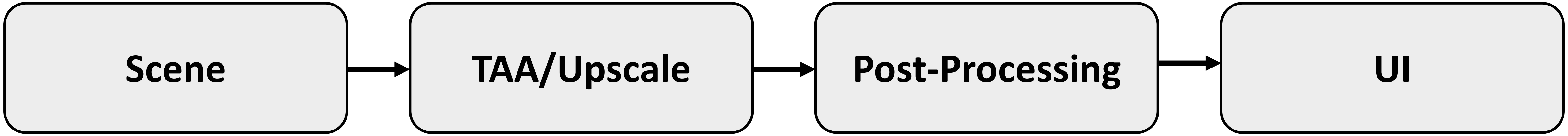
FSR before post-processing



```
▶ 169 errors, 17 warnings found! Press [F10] to see log, RMB to copy ✕

FPS: 42.33 (42.33)
FG FPS: 42.34
Frame: 7363
GPU Usage: 19%
API:      DX12
Adapter:  NVIDIA GeForce RTX 5070
Output:   Display 3
Shaders:  RELEASE
Cfg: SM:high, Tx:ultra
Antialiasing method: fsr3
Frame generation: off
Render resolution: quality
VSync:    Off
Rend Res: 1280x720
UI Res:   1920x1080
BB Res:   1920x1080
Polys:    11458534
Splits:   3478
```

Temporal AA/Upscale in rendering pipeline

Standard practice – placing before post-process


Scene

TAA/Upscale

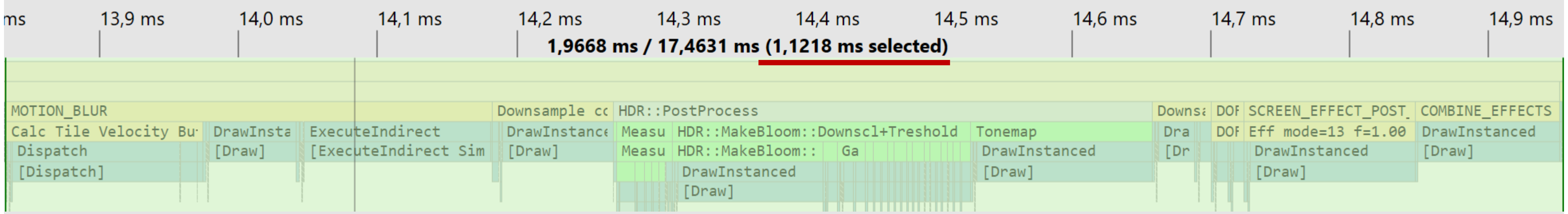
Post-Processing

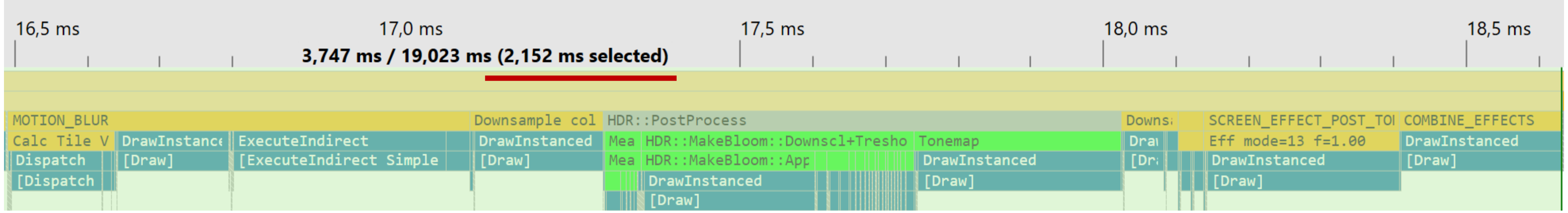
UI

But in this case we need to perform post-processing in upscaled resolution:

1.122 vs 2.152 ms

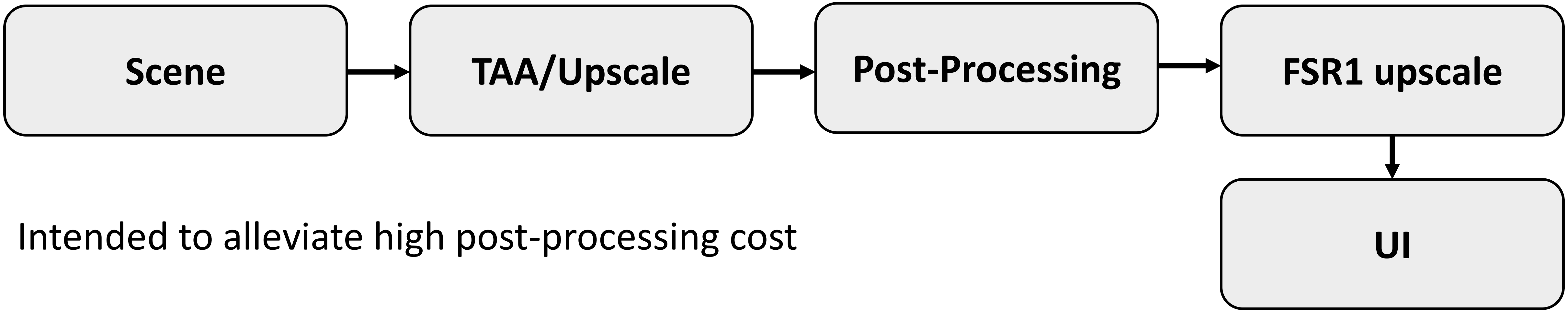
Xbox SX **quality mode**
(1440p -> 2160p)


1,9668 ms / 17,4631 ms (1,1218 ms selected)


3,747 ms / 19,023 ms (2,152 ms selected)

Temporal AA/Upscale in rendering pipeline

Combined upscale: temporal + spatial in the end



Intended to alleviate high post-processing cost

Rejected due to:

- FSR1 makes sense only with small resolution scale like <= 1.1 otherwise there is a quality degradation compared to the classic temporal upscaling scheme
- With small FSR1 scale there is no performance win

Temporal problems

Main tools to deal with problems

Algorithm	Masks	
FSR	Transparency & composition mask	Narrows color clamping window, relaxes locks contribution and luminance instability factor
	Reactive mask	Directly affects current frame weight when blending with history
XeSS	Responsive mask	
MFSR	Reactive mask	
TAA	Reactive mask	
DLSS	No masks	

Mask values are integrated into material system – artists are able to control them

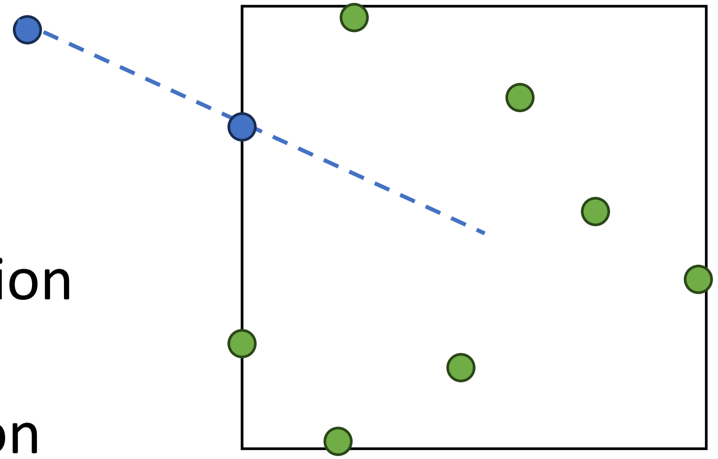
Transparency

VFXs that don’t write to velocity buffer

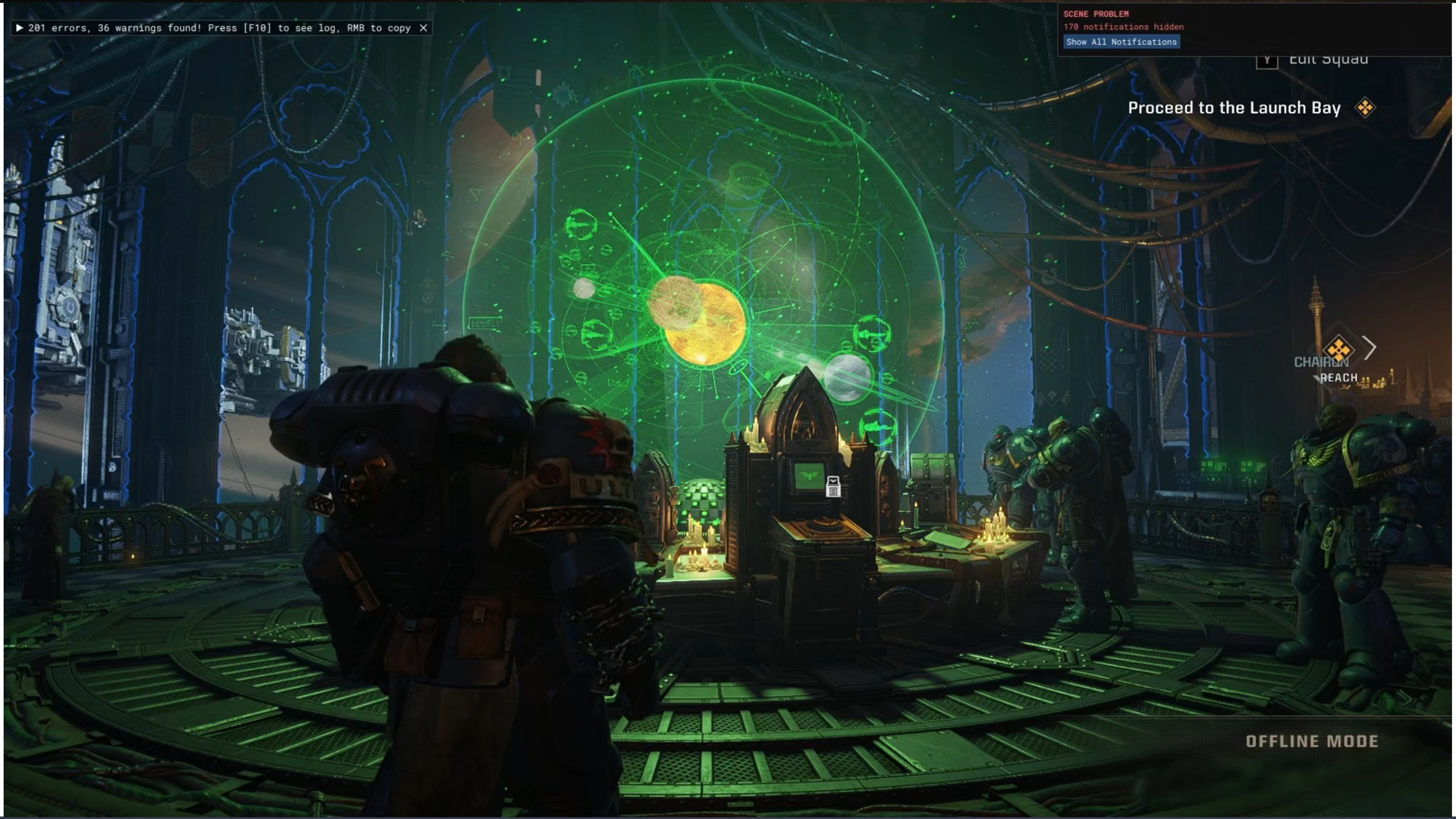
Here we rely on color clamping that clips history color if its significantly different from 3x3 neighborhood colors

- TAA – color clamping is enough to deal with it
- FSR – need VFXs to be written to transparency & composition mask
- XeSS – we write VFXs into responsive mask. Lower values on lower render resolution presets

Current frame neighborhood AABB and **history** clamping



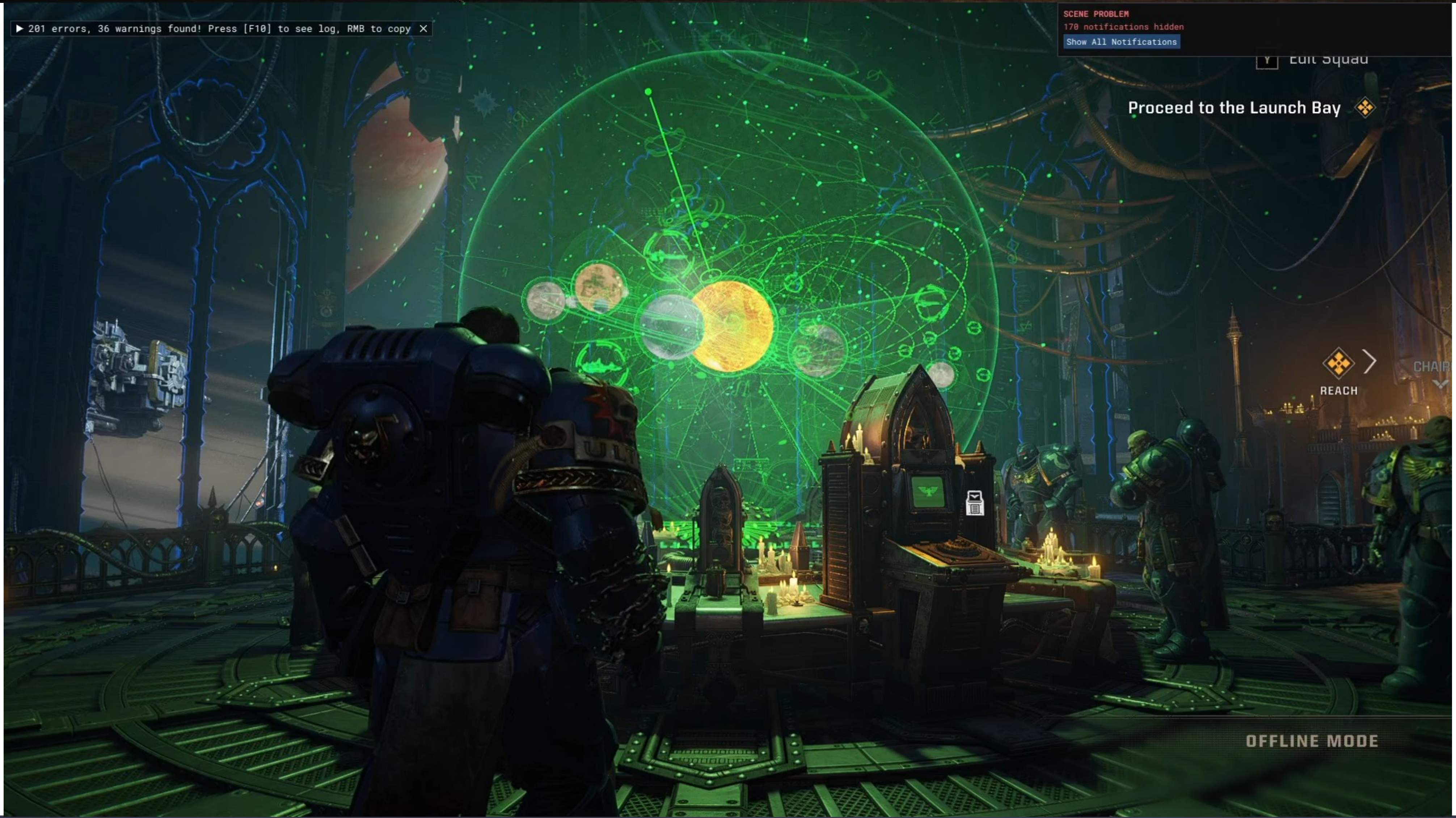
Without transparency mask



```
▶ 201 errors, 36 warnings found! Press [F10] to see log, RMB to copy ✕
```

```
SCENE PROBLEM
170 notifications hidden
Show All Notifications
```

With transparency mask



```
▶ 201 errors, 36 warnings found! Press [F10] to see log, RMB to copy ✕
```

```
SCENE PROBLEM
170 notifications hidden
Show All Notifications
```

Particles

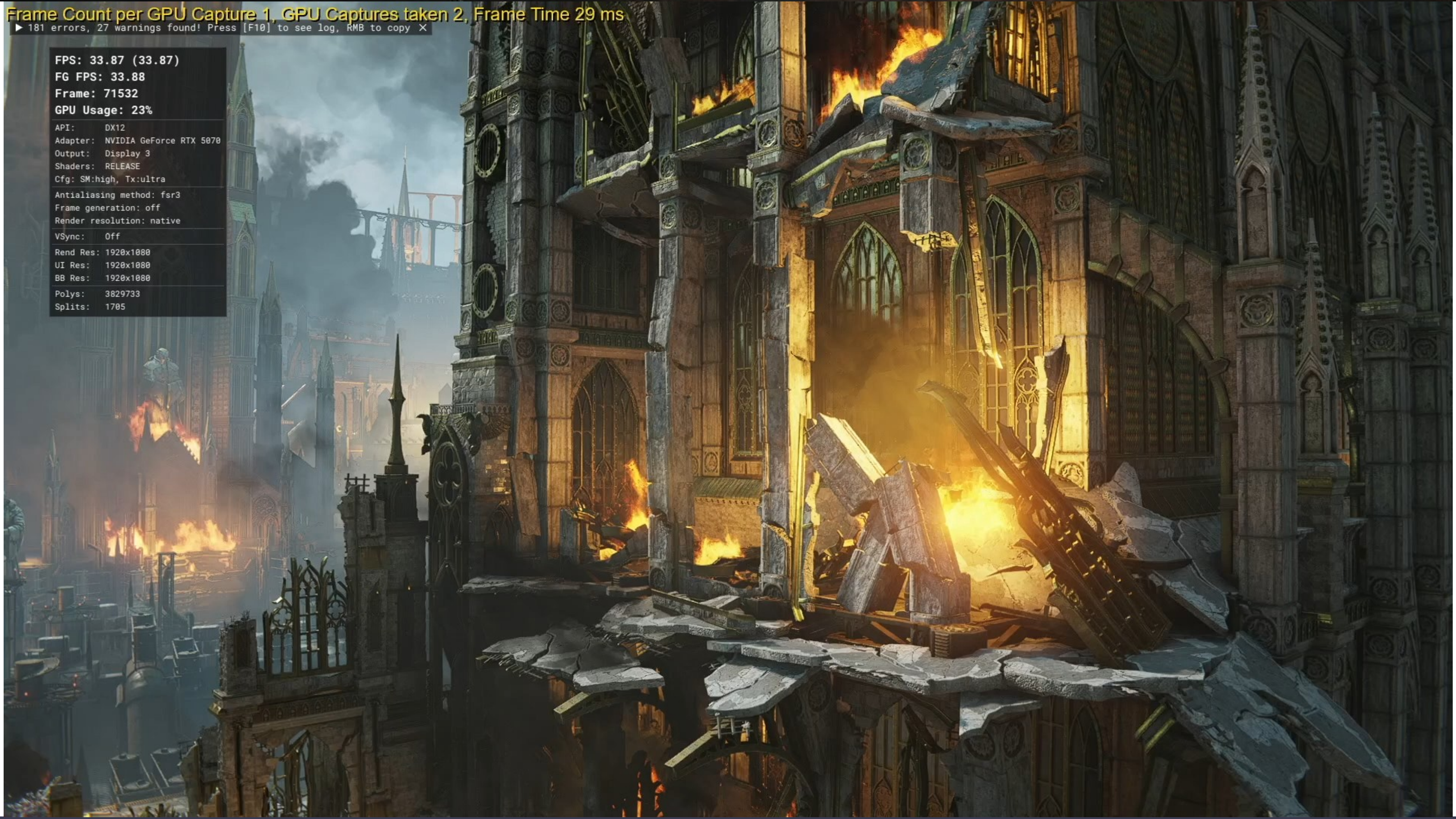
Fast particles that don't write to velocity buffer

- Need to increase current frame weight to avoid ghosting and diminishing in size
- Color clamping still helps on noisy background

We locally increase current frame weight using reactive mask

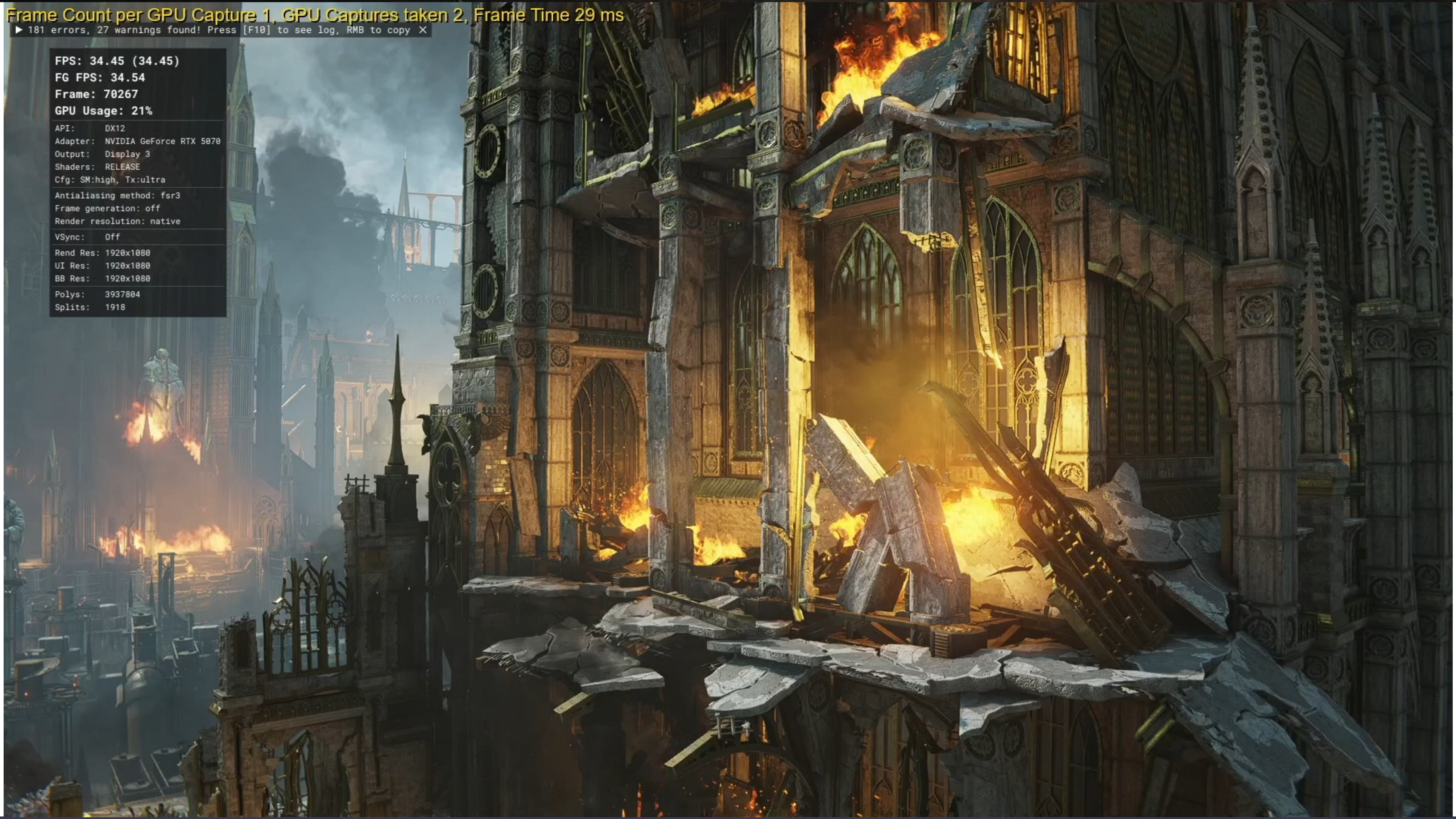
- TAA – particles write to reactive mask only
- FSR – particles write to reactive mask and transparency & composition to narrow color clamping AABB and relax locks for noisy background
- MFSR – same as FSR, plus we combine previous frame reactive mask with the current one
- XeSS – particles write to responsive mask

Without masks



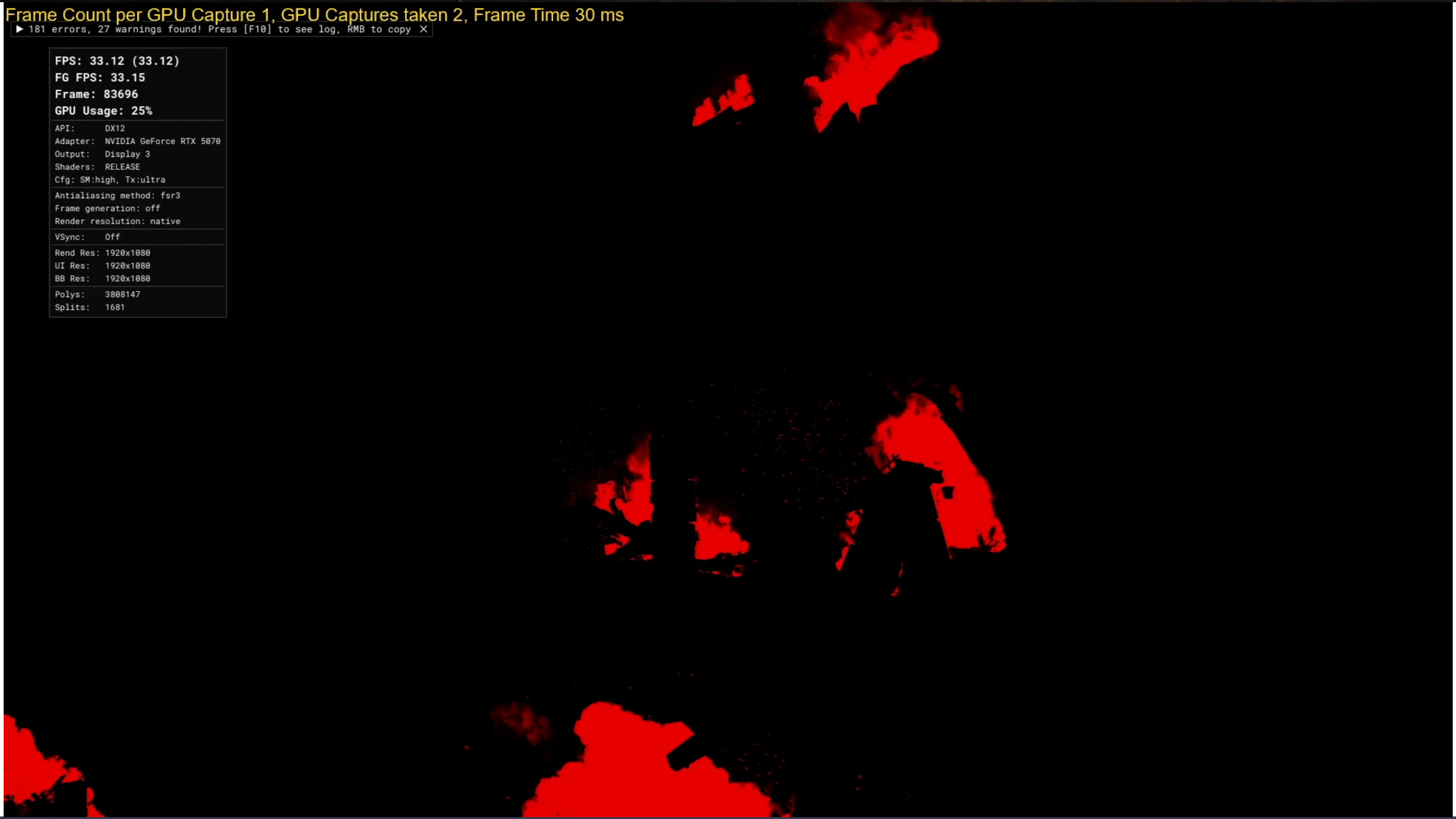
```
Frame Count per GPU Capture 1, GPU Captures taken 2, Frame Time 29 ms
▶ 181 errors, 27 warnings found! Press [F10] to see log, RMB to copy ✕

FPS: 33.87 (33.87)
FG FPS: 33.88
Frame: 71532
GPU Usage: 23%
API:      DX12
Adapter:  NVIDIA GeForce RTX 5070
Output:   Display 3
Shaders:  RELEASE
Cfg: SM:high, Tx:ultra
Antialiasing method: fsr3
Frame generation: off
Render resolution: native
VSync:    Off
Rend Res: 1920x1080
UI Res:   1920x1080
BB Res:   1920x1080
Polys:    3829733
Splits:   1705
```

With masks



```
Frame Count per GPU Capture 1, GPU Captures taken 2, Frame Time 29 ms
▶ 181 errors, 27 warnings found! Press [F10] to see log, RMB to copy ✕

FPS: 34.45 (34.45)
FG FPS: 34.54
Frame: 70267
GPU Usage: 21%
API:       DX12
Adapter:   NVIDIA GeForce RTX 5070
Output:    Display 3
Shaders:   RELEASE
Cfg: SM:high, Tx:ultra
Antialiasing method: fsr3
Frame generation: off
Render resolution: native
VSync:     Off
Rend Res:  1920x1080
UI Res:    1920x1080
BB Res:    1920x1080
Polys:     3937804
Splits:    1918
```

Reactive mask



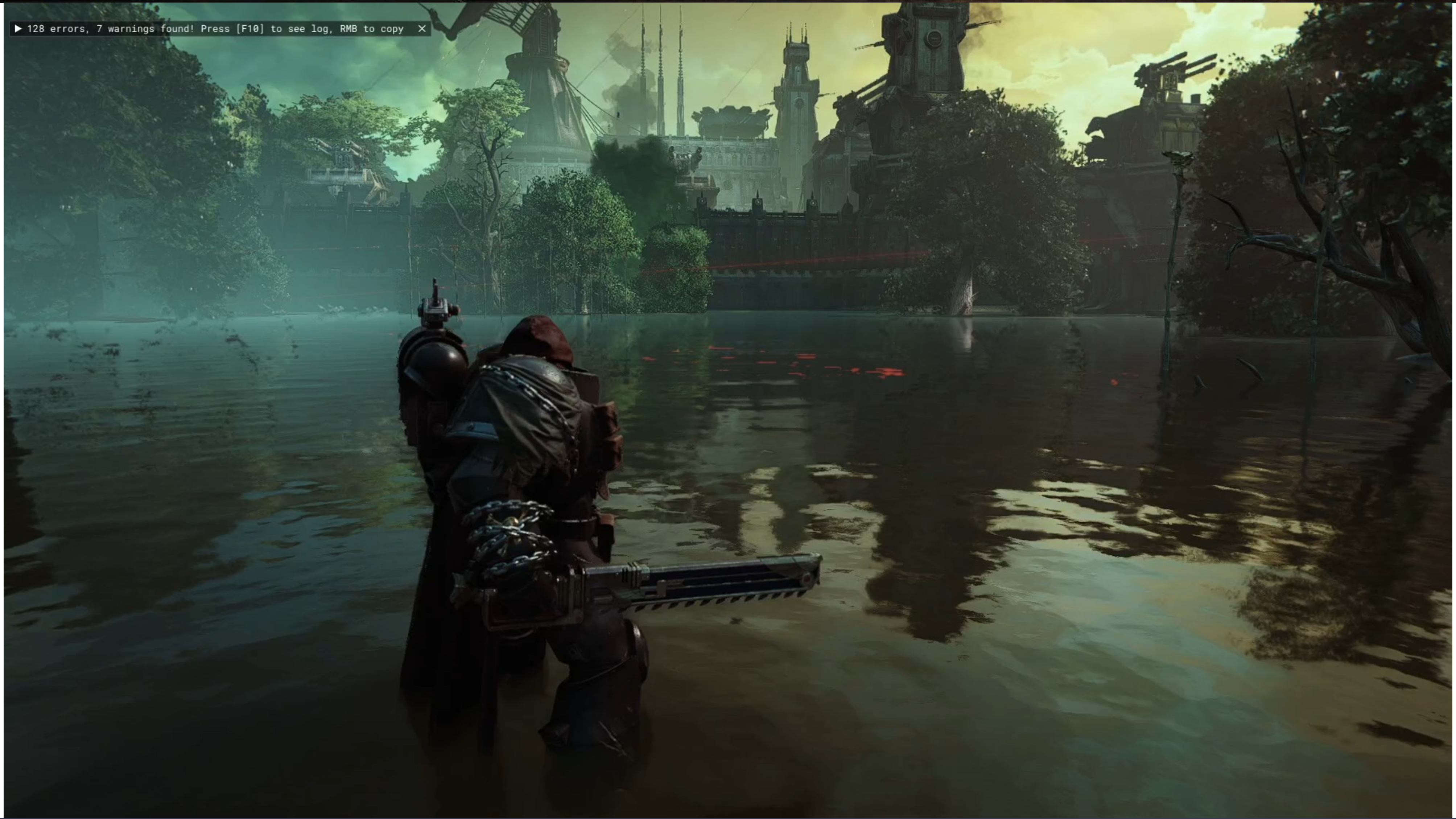
```
Frame Count per GPU Capture 1, GPU Captures taken 2, Frame Time 30 ms
▶ 181 errors, 27 warnings found! Press [F10] to see log, RMB to copy ✕

FPS: 33.12 (33.12)
FG FPS: 33.15
Frame: 83696
GPU Usage: 25%
API:       DX12
Adapter:   NVIDIA GeForce RTX 5070
Output:    Display 3
Shaders:   RELEASE
Cfg: SM:high, Tx:ultra
Antialiasing method: fsr3
Frame generation: off
Render resolution: native
VSync:     Off
Rend Res:  1920x1080
UI Res:    1920x1080
BB Res:    1920x1080
Polys:     3808147
Splits:    1681
```

Trails

Since such objects are tiny and not written to depth and velocity

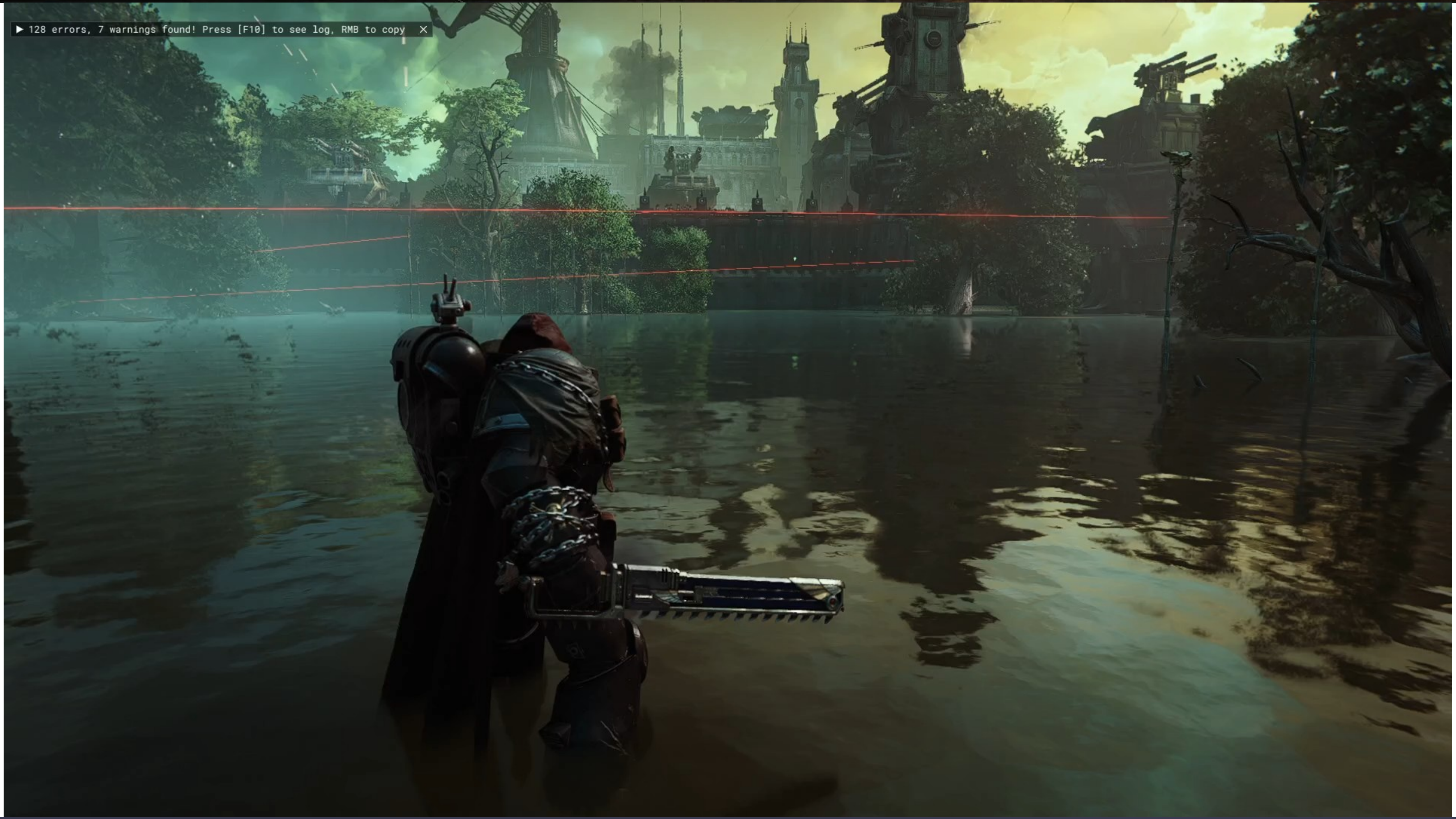
- Need to adaptively broaden trails to make them 1-2 pixel to avoid excessive blend with background
- Need to conservatively apply reactive mask just a little in case of moving trail to avoid ghosting

Lasers before fixes



```
▶ 128 errors, 7 warnings found! Press [F10] to see log, RMB to copy ✕
```

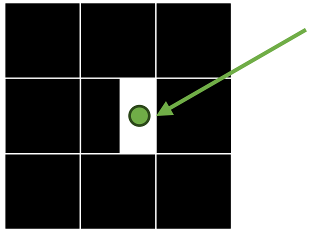
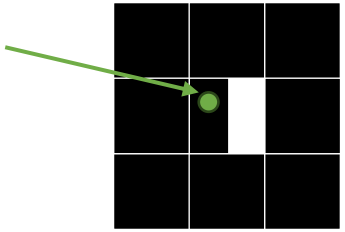
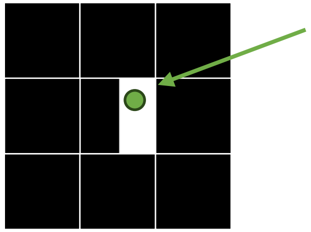
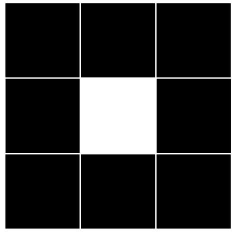
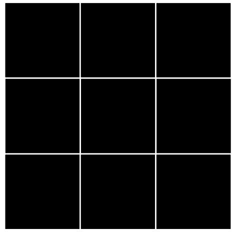
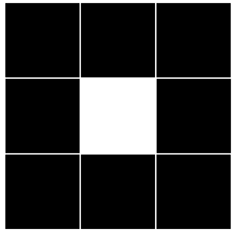
Lasers after fixes



```
▶ 128 errors, 7 warnings found! Press [F10] to see log, RMB to copy ✕
```

Subpixel shading details

Jitter + color clamping = flickering

	0-th frame	1-st frame	2-nd frame
Current jitter pos **on original signal**			
Current frame shading			
History sample	■	History clamp! □	■
Output	□	■	□

Subpixel shading details

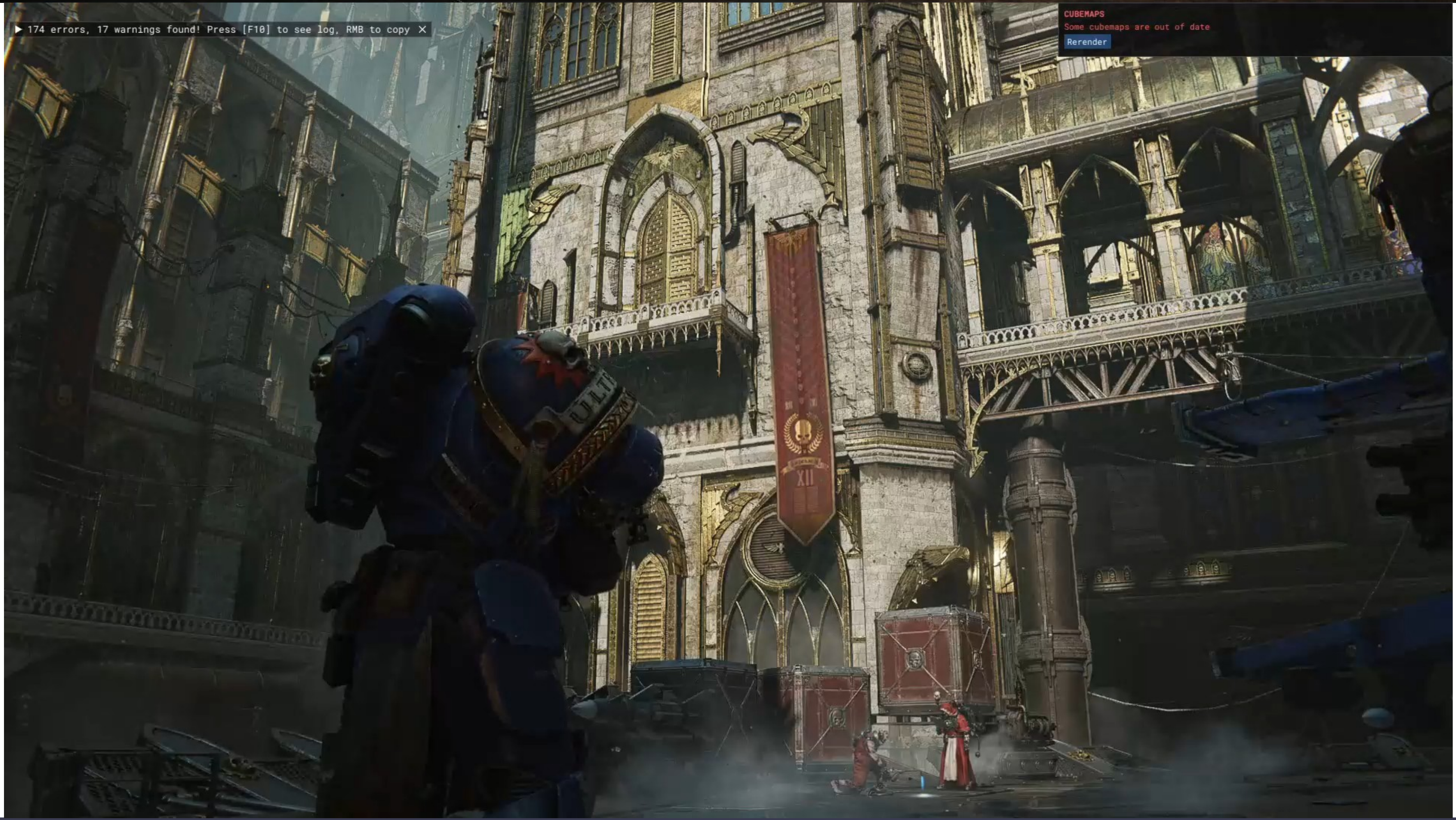
We can only alleviate this problem

- Relax color clamping – reduce mask values where possible
- Decrease resolution of normal maps for some grainy materials
- FSR specific – disable velocity factor (FFX_API_CONFIGURE_UPSCALE_KEY_FVELOCITYFACTOR)

Potential to-do:

- Introduce luminance aware filter to make fireflies less bright, currently we did it only in TAA

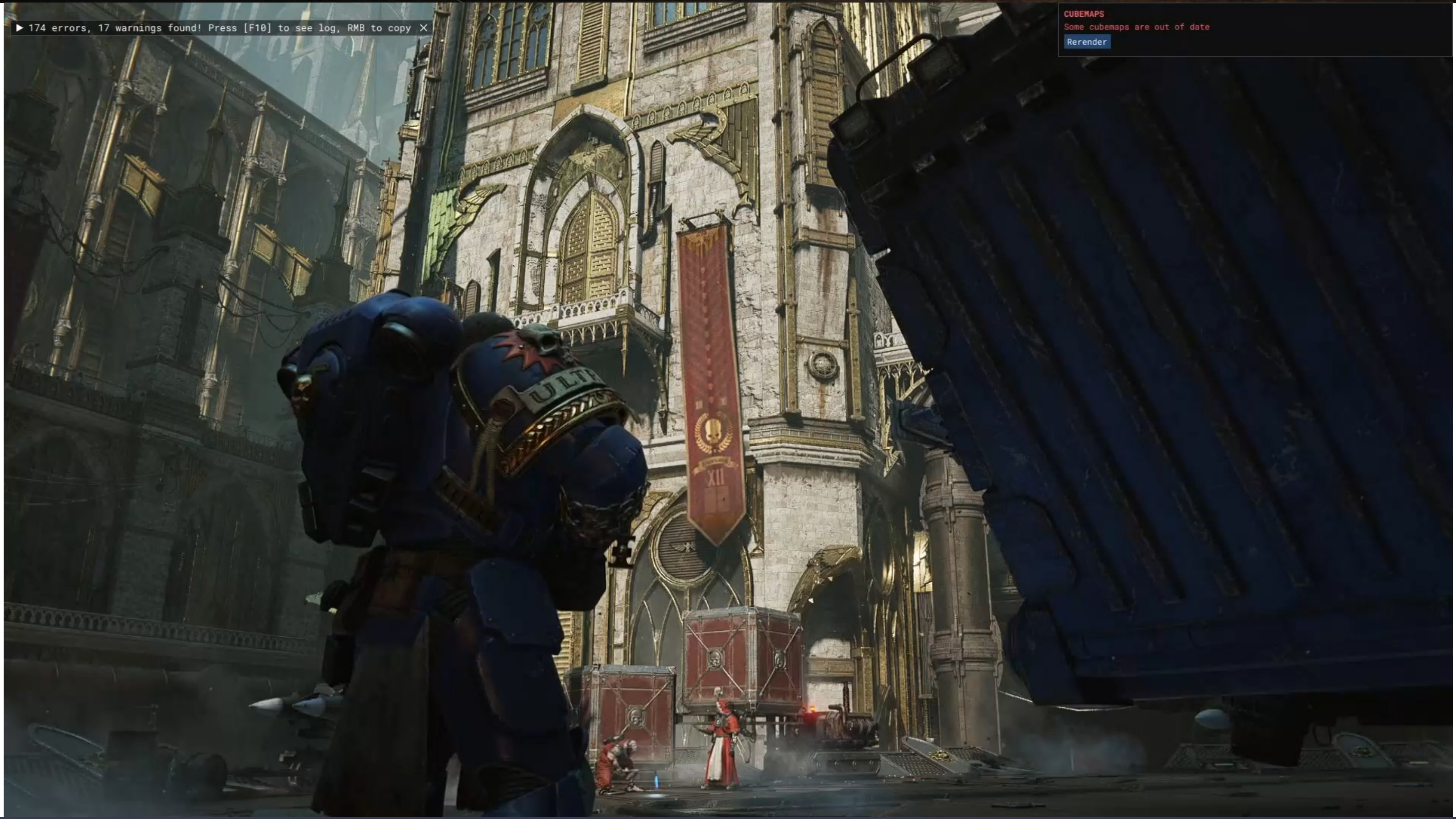
Velocity factor default



```
▶ 174 errors, 17 warnings found! Press [F10] to see log, RMB to copy ✕
```

```
CUBEMAPS
Some cubemaps are out of date
```

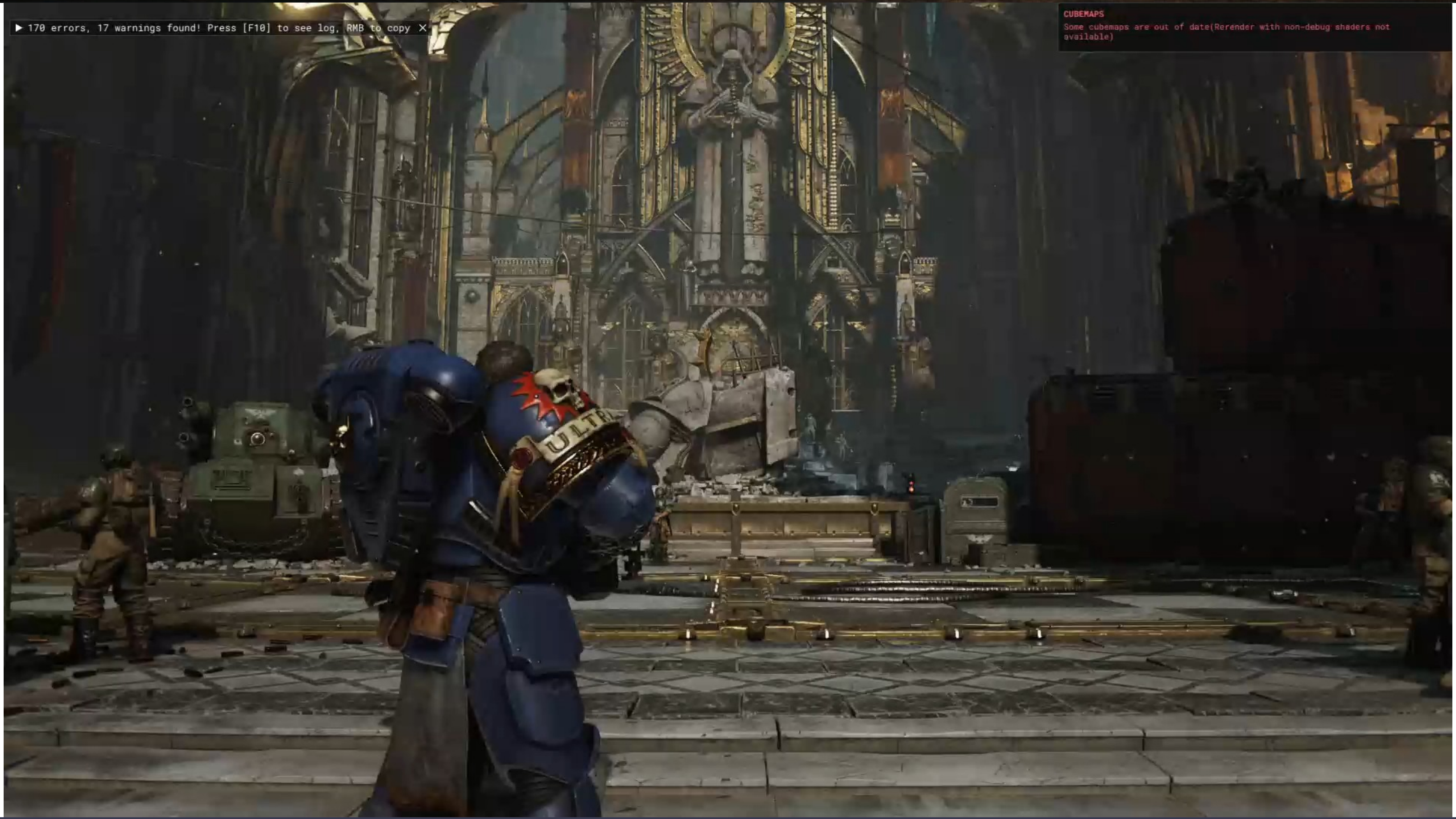
Velocity factor off



```
▶ 174 errors, 17 warnings found! Press [F10] to see log, RMB to copy ✕
```

```
CUBEMAPS
Some cubemaps are out of date
```

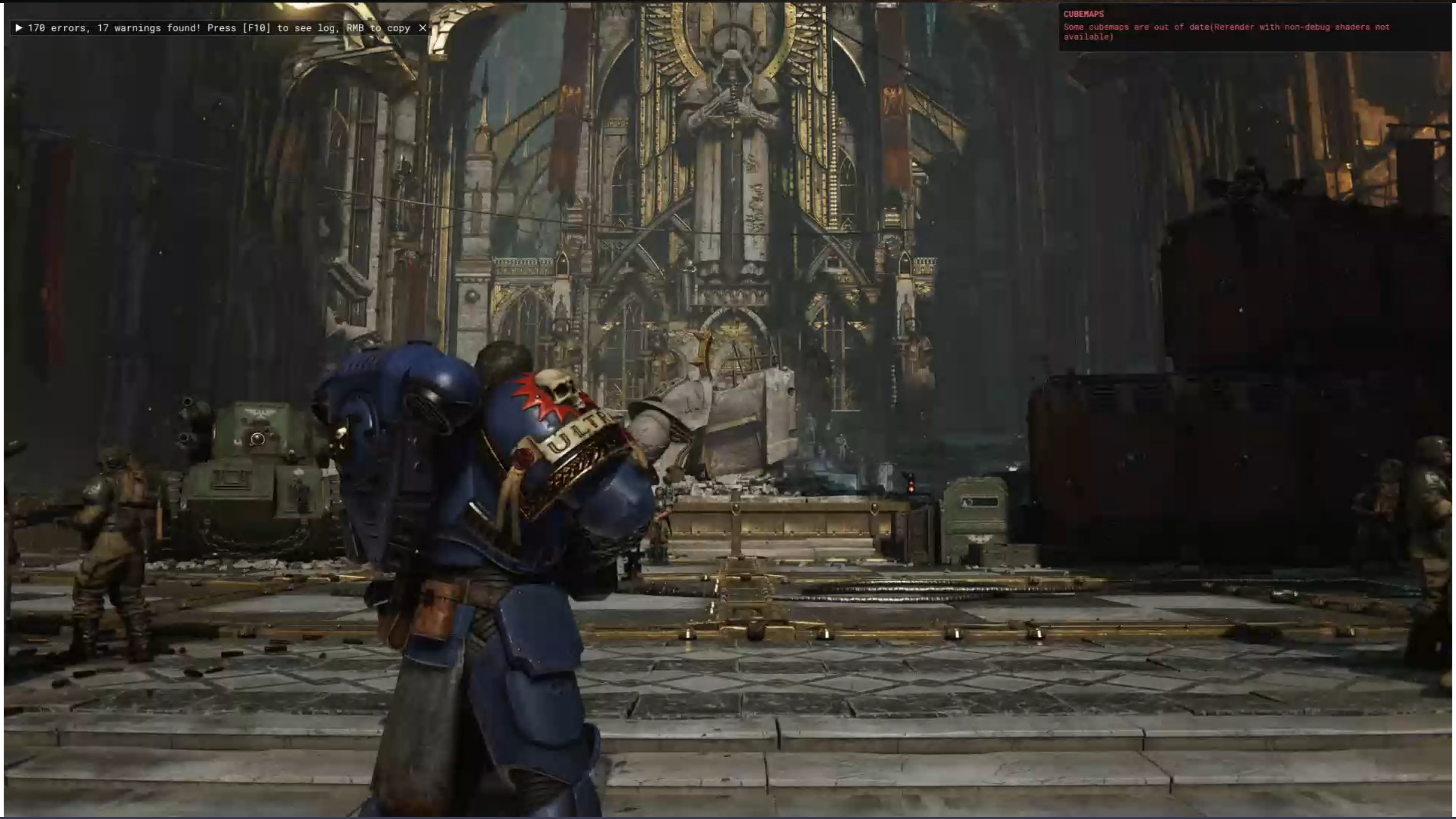
Luminance filter off



```
▶ 170 errors, 17 warnings found! Press [F10] to see log, RMB to copy ✕
```

```
CUBEMAPS
Some cubemaps are out of date(Rerender with non-debug shaders not
available)
```

Luminance filter on



```
▶ 170 errors, 17 warnings found! Press [F10] to see log, RMB to copy ✕
```

```
CUBEMAPS
Some cubemaps are out of date(Rerender with non-debug shaders not
available)
```

Jittered input problem

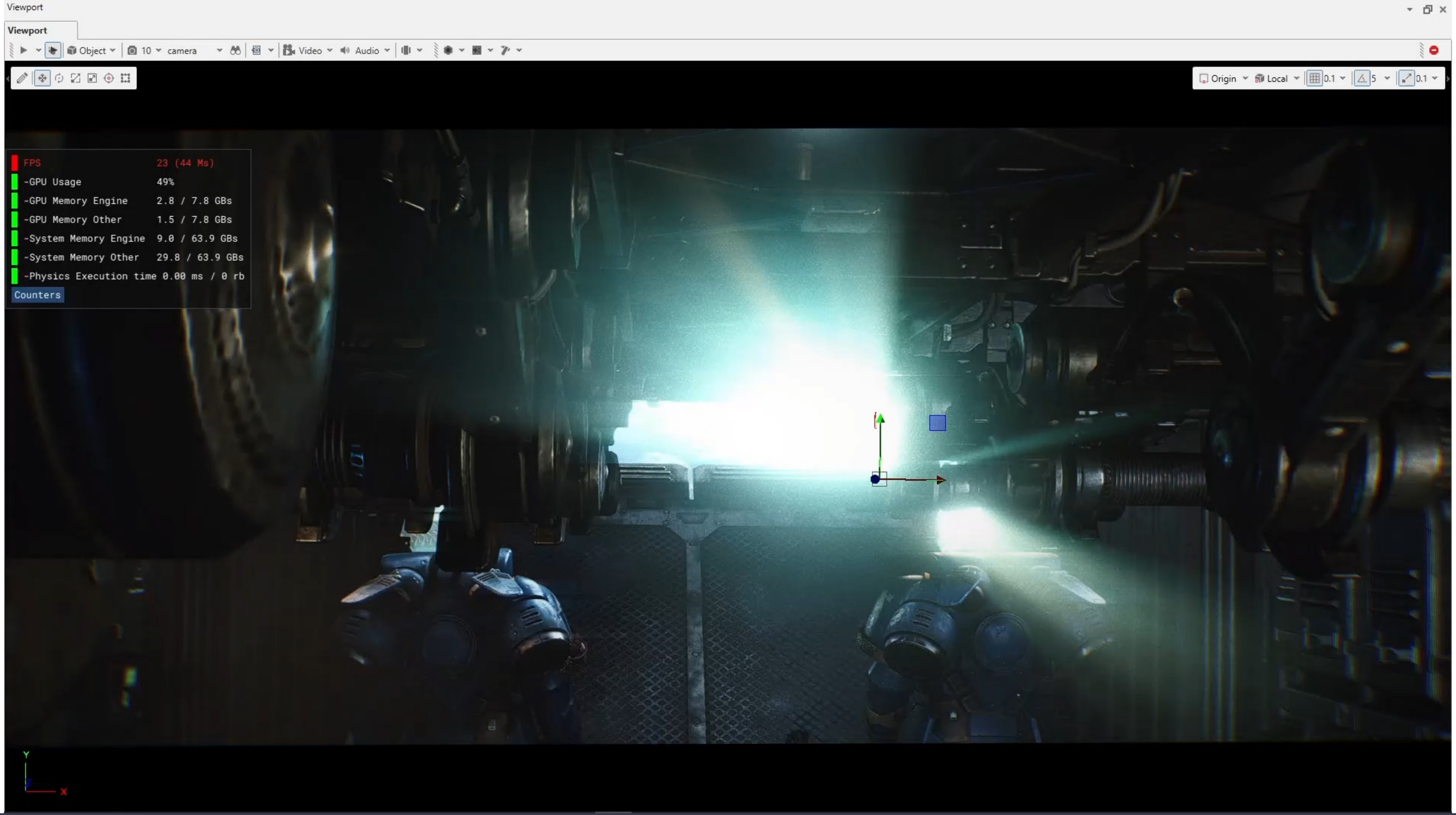
Common examples are depth and velocity and all resources that depend on it

Jitter and large effect’s footprint caused flickering in:

- Screen-space artistic lightshafts due to depth
- Screen-space artistic flares due to depth
- Bloom due to artistic bloom mask – mask is written by geometry that needs artificial bloom
- Depth of field due to circle of confusion and half res color buffer

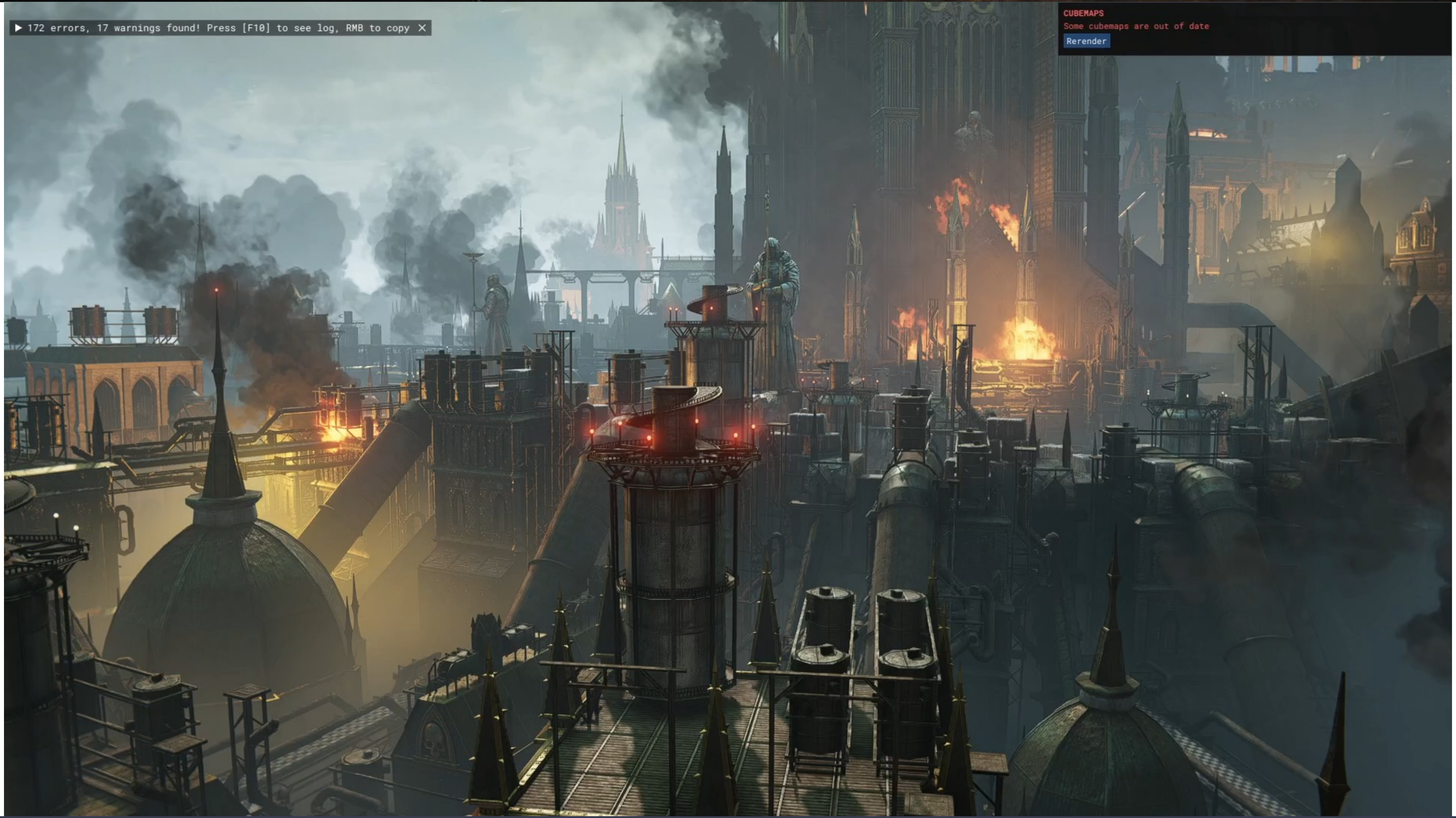
Solution: apply lightweight TAA or some hysteresis to depth-dependent resources

Lightshafts TAA on/off



```
FPS                          23 (44 Ms)
-GPU Usage                   49%
-GPU Memory Engine           2.8 / 7.8 GBs
-GPU Memory Other            1.5 / 7.8 GBs
-System Memory Engine        9.0 / 63.9 GBs
-System Memory Other         29.8 / 63.9 GBs
-Physics Execution time 0.00 ms / 0 rb
Counters
```

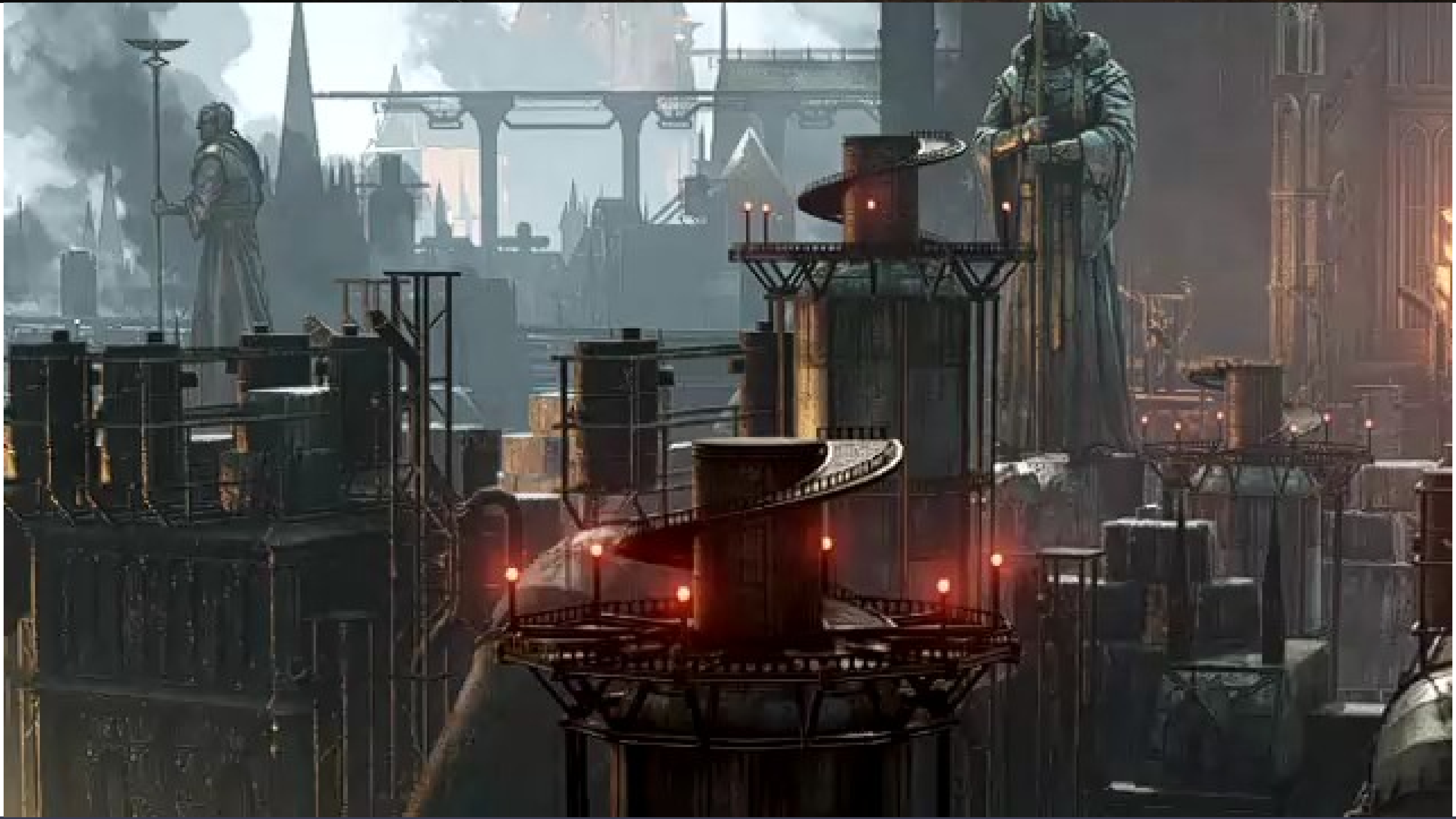
Bloom mask TAA off



```
▶ 172 errors, 17 warnings found! Press [F10] to see log, RMB to copy ✕
```

```
CUBEMAPS
Some cubemaps are out of date
```

Bloom mask TAA off



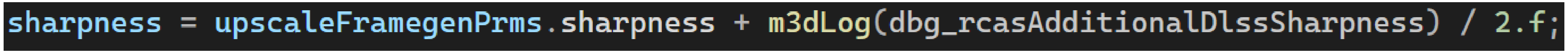
Bloom mask TAA on

Sharpness

We use AMD RCAS everywhere

- Problem with different base sharpness level for each upscaler

  Solution: we've applied additional base sharpness for DLSS:

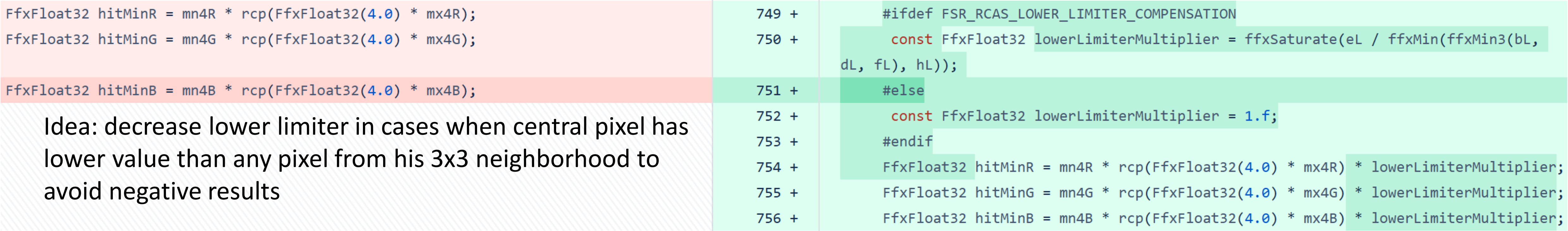


```
sharpness = upscaleFramegenPrms.sharpness + m3dLog(dbg_rcasAdditionalDlssSharpness) / 2.f;
```

`dbg_rcasAdditionalDlssSharpness` – base sharpness to make up for less sharp output from DLSS and XeSS
`upscaleFramegenPrms.sharpness` – sharpness from game settings

- Problem with dark pixels in some setups (sharpness close to 1)

  Solution: we've slightly patched limiters part of RCAS algorithm:

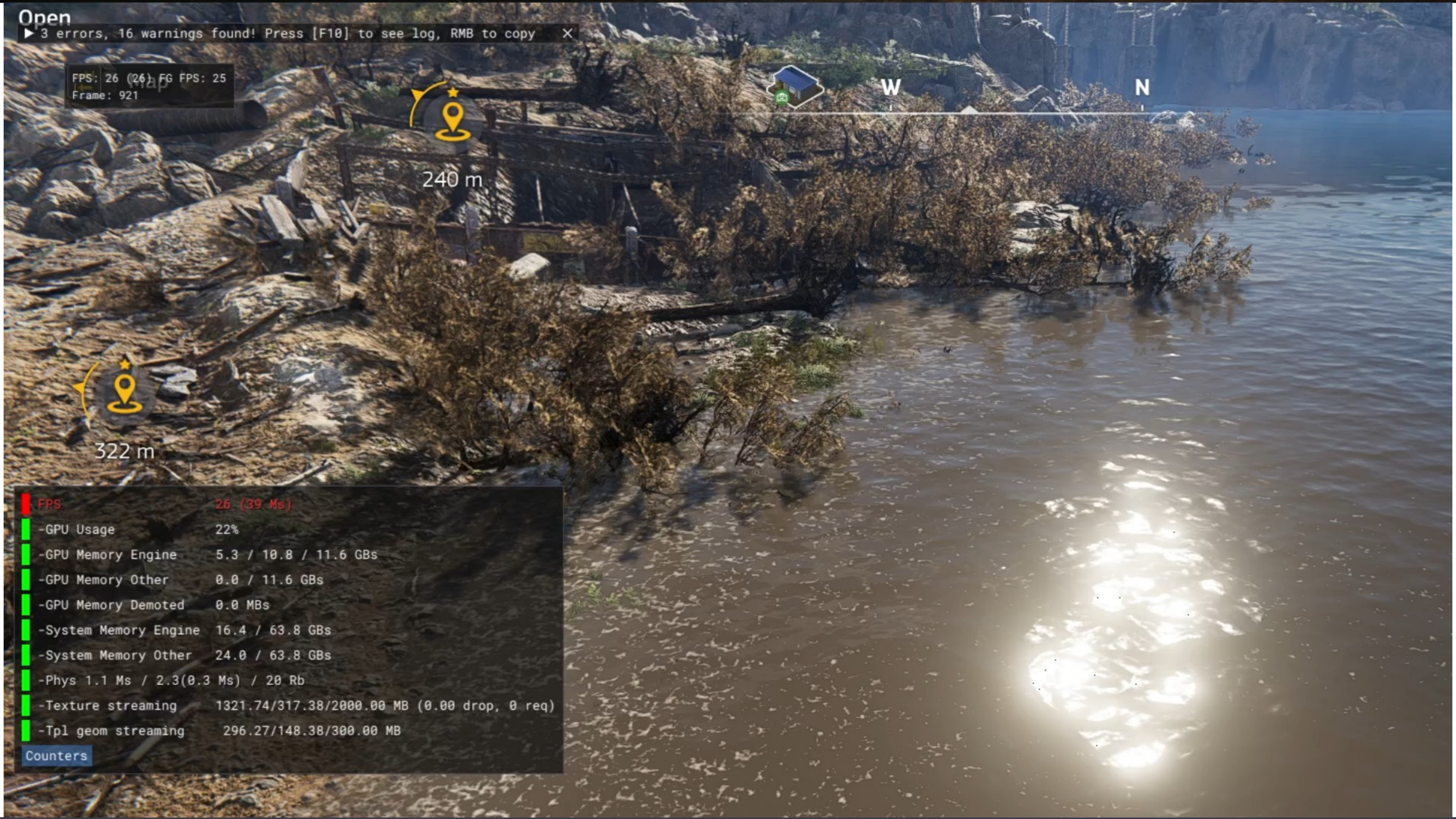


```
FfxFloat32 hitMinR = mn4R * rcp(FfxFloat32(4.0) * mx4R);
FfxFloat32 hitMinG = mn4G * rcp(FfxFloat32(4.0) * mx4G);

FfxFloat32 hitMinB = mn4B * rcp(FfxFloat32(4.0) * mx4B);
```

```
+    #ifdef FSR_RCAS_LOWER_LIMITER_COMPENSATION
+     const FfxFloat32 lowerLimiterMultiplier = ffxSaturate(eL / ffxMin(ffxMin3(bL, dL, fL), hL));
+    #else
+     const FfxFloat32 lowerLimiterMultiplier = 1.f;
+    #endif
+    FfxFloat32 hitMinR = mn4R * rcp(FfxFloat32(4.0) * mx4R) * lowerLimiterMultiplier;
+    FfxFloat32 hitMinG = mn4G * rcp(FfxFloat32(4.0) * mx4G) * lowerLimiterMultiplier;
+    FfxFloat32 hitMinB = mn4B * rcp(FfxFloat32(4.0) * mx4B) * lowerLimiterMultiplier;
```

Idea: decrease lower limiter in cases when central pixel has lower value than any pixel from his 3x3 neighborhood to avoid negative results

Sharpness black dots



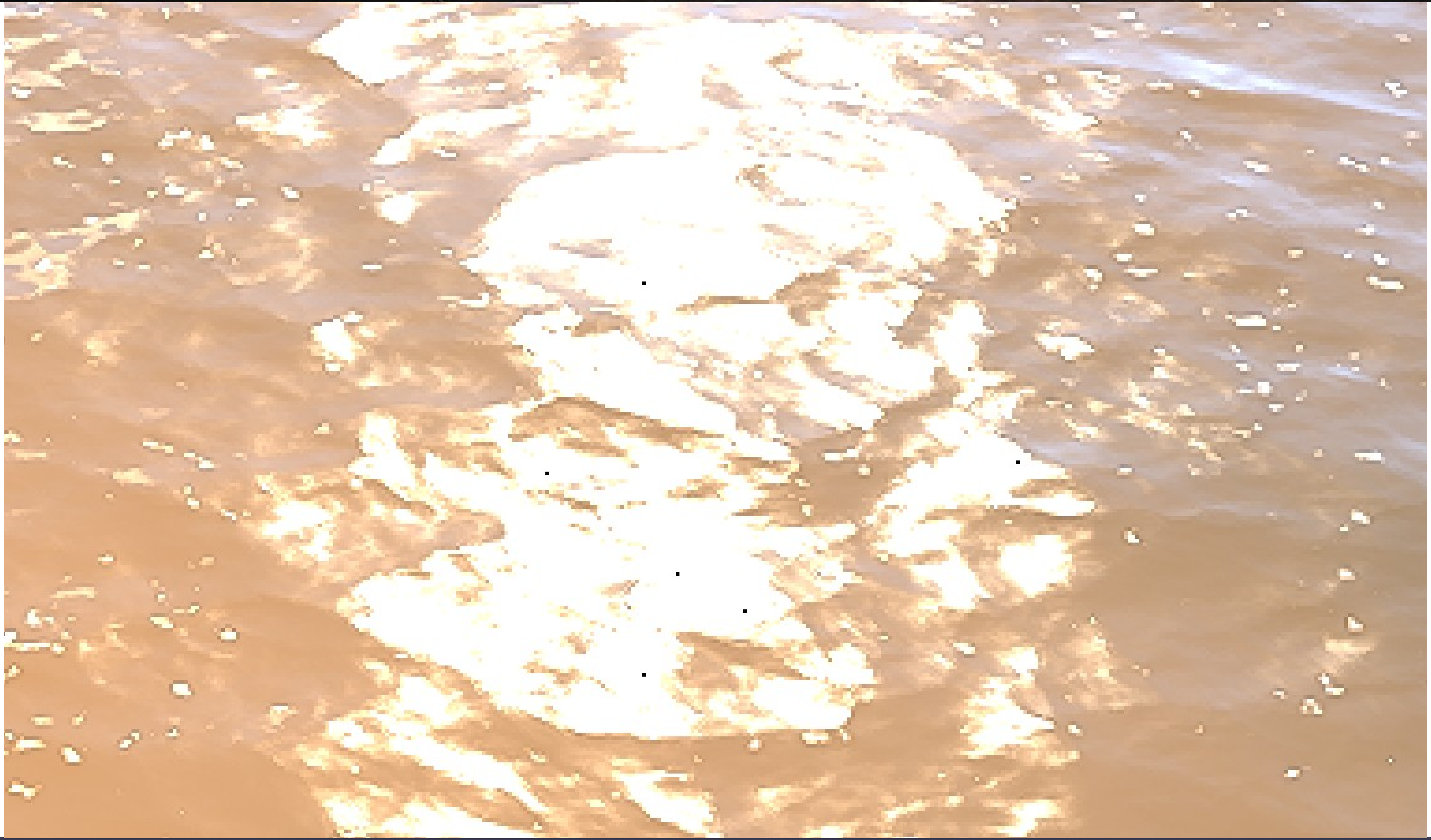
Open

```
▶ 3 errors, 16 warnings found! Press [F10] to see log, RMB to copy   X
```

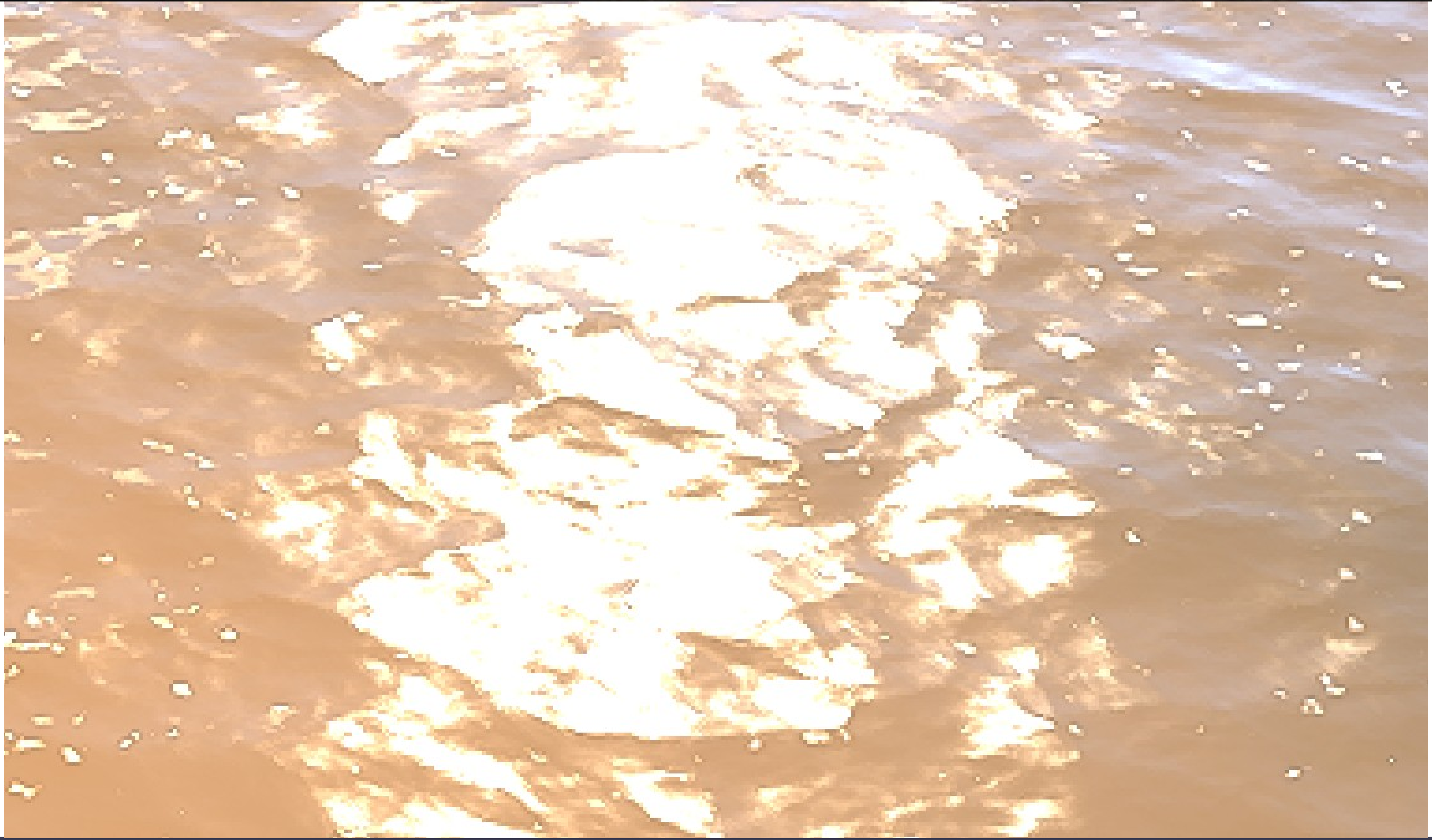
```
FPS: 26 (26) FG FPS: 25
Frame: 921
```

```
FPS                    26 (39 Ms)
-GPU Usage             22%
-GPU Memory Engine     5.3 / 10.8 / 11.6 GBs
-GPU Memory Other      0.0 / 11.6 GBs
-GPU Memory Demoted    0.0 MBs
-System Memory Engine  16.4 / 63.8 GBs
-System Memory Other   24.0 / 63.8 GBs
-Phys 1.1 Ms / 2.3(0.3 Ms) / 20 Rb
-Texture streaming     1321.74/317.38/2000.00 MB (0.00 drop, 0 req)
-Tpl geom streaming     296.27/148.38/300.00 MB
```

Sharpness black dots



After lower limiter fix



Ghosting due to sudden changes

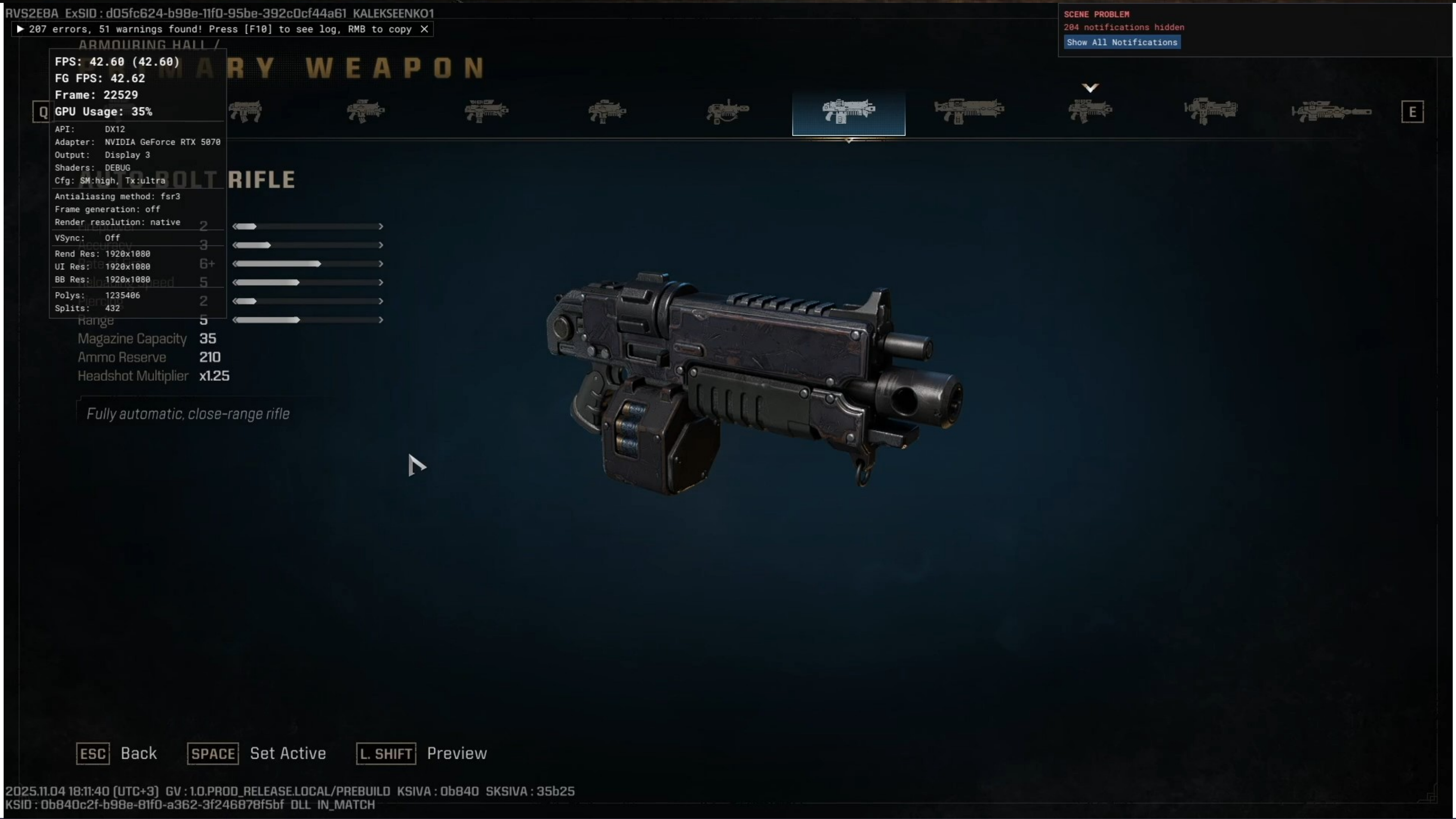
Sometimes there are situations when some object suddenly disappears or appears again
For example, armor/weapon change in inventory menu

We don’t use `reset` flag available in each upscalers because it completely resets history and produce unpleasant convergence effect just after applying

Reactive mask is ideal choice for a partial history reset.

When armor/weapon changes game logic script triggers writing to the reactive mask there are no any ghosting.

Weapon switch before



RVS2E8A ExSID : d05fc624-b98e-11f0-95be-392c0cf44a61 KALEKSEENKO1

```
▶ 207 errors, 51 warnings found! Press [F10] to see log, RMB to copy ✕
```

```
SCENE PROBLEM
204 notifications hidden
Show All Notifications
```

ARMOURING HALL /

PRIMARY WEAPON

```
FPS: 42.60 (42.60)
FG FPS: 42.62
Frame: 22529
GPU Usage: 35%
API:       DX12
Adapter:   NVIDIA GeForce RTX 5070
Output:    Display 3
Shaders:   DEBUG
Cfg: SM:high, Tx:ultra
Antialiasing method: fsr3
Frame generation: off
Render resolution: native
VSync:     Off
Rend Res: 1920x1080
UI Res:   1920x1080
BB Res:   1920x1080
Polys:     1235406
Splits:    432
```

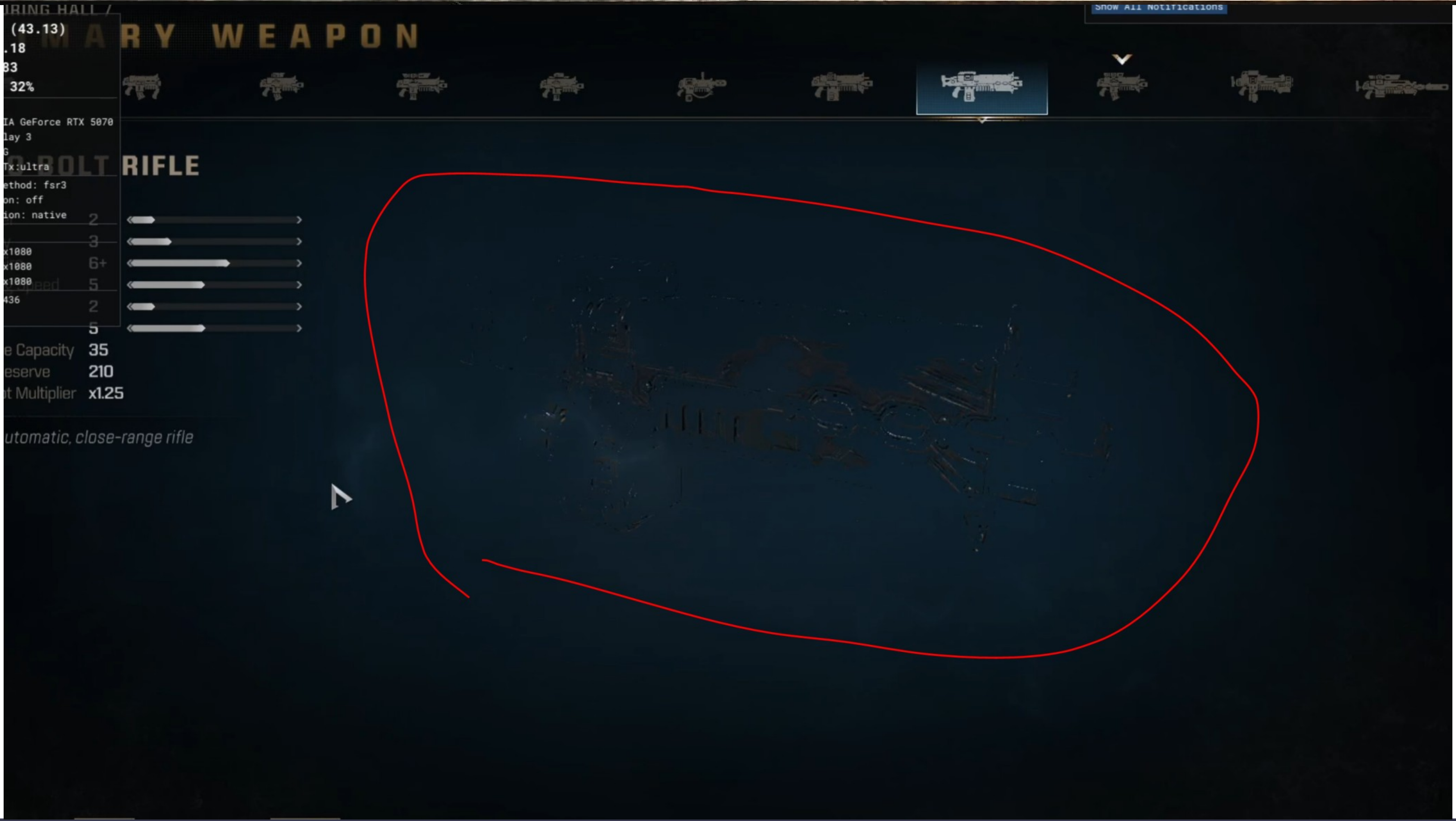
AUTO BOLT RIFLE

Firepower	2
Accuracy	3
Rate of Fire	6+
Reloading Speed	5
Piercing	2
Range	5
Magazine Capacity	35
Ammo Reserve	210
Headshot Multiplier	x1.25

Fully automatic, close-range rifle

2025.11.04 18:11:40 (UTC+3) GV : 1.0.PROD_RELEASE.LOCAL/PREBUILD KSIVA : 0b840 SKSIVA : 35b25
KSID : 0b840c2f-b98e-81f0-a362-3f246878f5bf DLL IN_MATCH

Weapon switch before

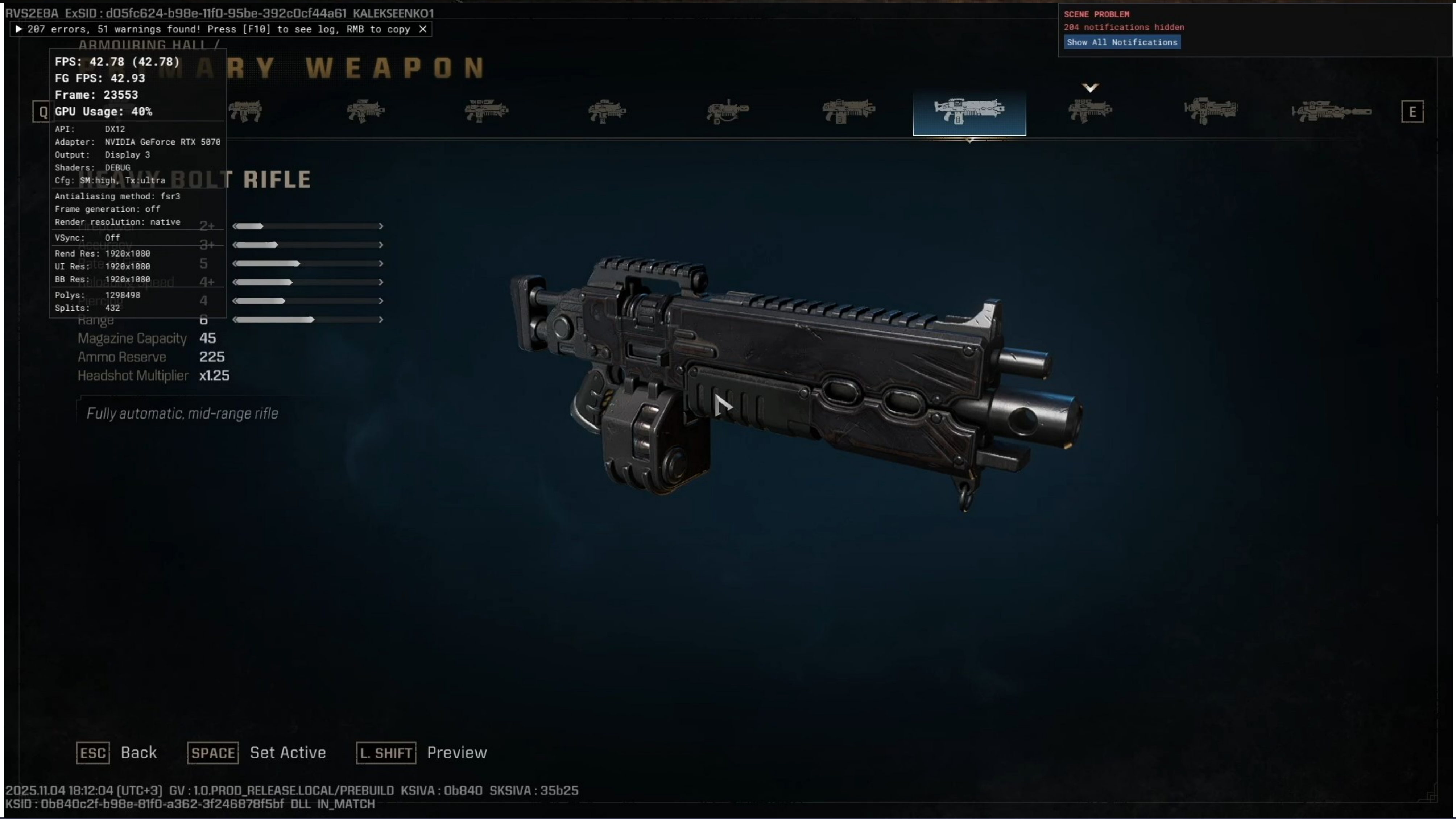


IRING HALL /
(43.13)
.18
83
32%
IA GeForce RTX 5070
lay 3
G
Tx:ultra
ethod: fsr3
on: off
ion: native
x1080
x1080
x1080
436

	2
	3
	6+
peed	5
	2
	5
e Capacity	35
eserve	210
t Multiplier	x1.25

utomatic, close-range rifle

Weapon switch after



```
RVS2E8A  ExSID : d05fc624-b98e-11f0-95be-392c0cf44a61  KALEKSEENKO1
▶ 207 errors, 51 warnings found! Press [F10] to see log, RMB to copy ✕
```

```
SCENE PROBLEM
204 notifications hidden
Show All Notifications
```

```
FPS: 42.78 (42.78)
FG FPS: 42.93
Frame: 23553
GPU Usage: 40%
API:       DX12
Adapter:   NVIDIA GeForce RTX 5070
Output:    Display 3
Shaders:   DEBUG
Cfg: SM:high, Tx:ultra
Antialiasing method: fsr3
Frame generation: off
Render resolution: native
VSync:     Off
Rend Res: 1920x1080
UI Res:   1920x1080
BB Res:   1920x1080
Polys:     1298498
Splits:    432
```

ARMOURING HALL /

PRIMARY WEAPON

HEAVY BOLT RIFLE

	2+
	3+
	5
	4+
	4
Range	6
Magazine Capacity	45
Ammo Reserve	225
Headshot Multiplier	x1.25

Fully automatic, mid-range rifle

```
2025.11.04 18:12:04 (UTC+3)  GV : 1.0.PROD_RELEASE.LOCAL/PREBUILD  KSIVA : 0b840  SKSIVA : 35b25
KSID : 0b840c2f-b98e-81f0-a362-3f246878f5bf  DLL  IN_MATCH
```

Frame generation

Our Frame generation integration

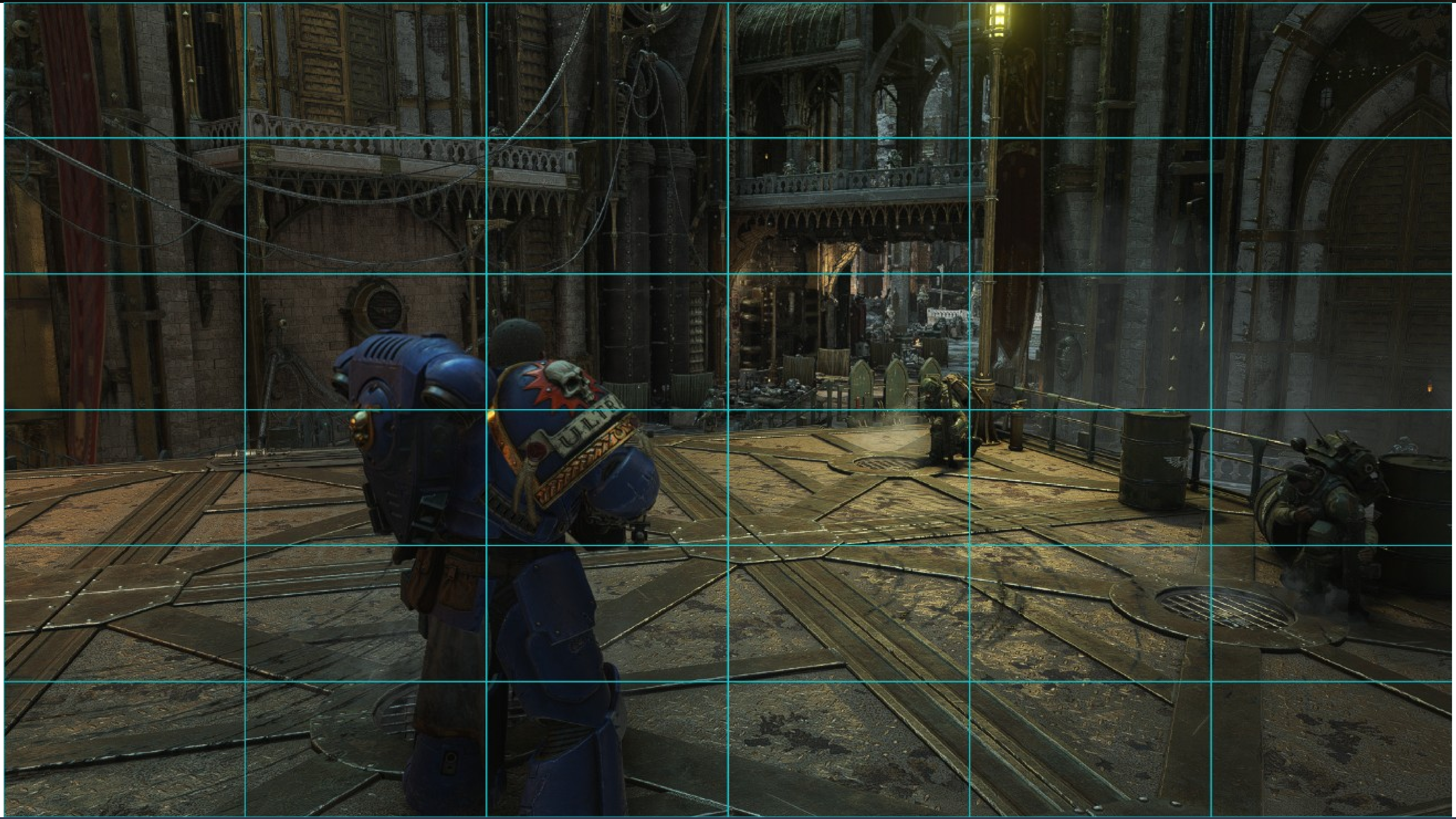
Algorithms and platforms:

- FSR3 FG – anywhere on PC
- DLSS FG – NVIDIA, from 4000 series

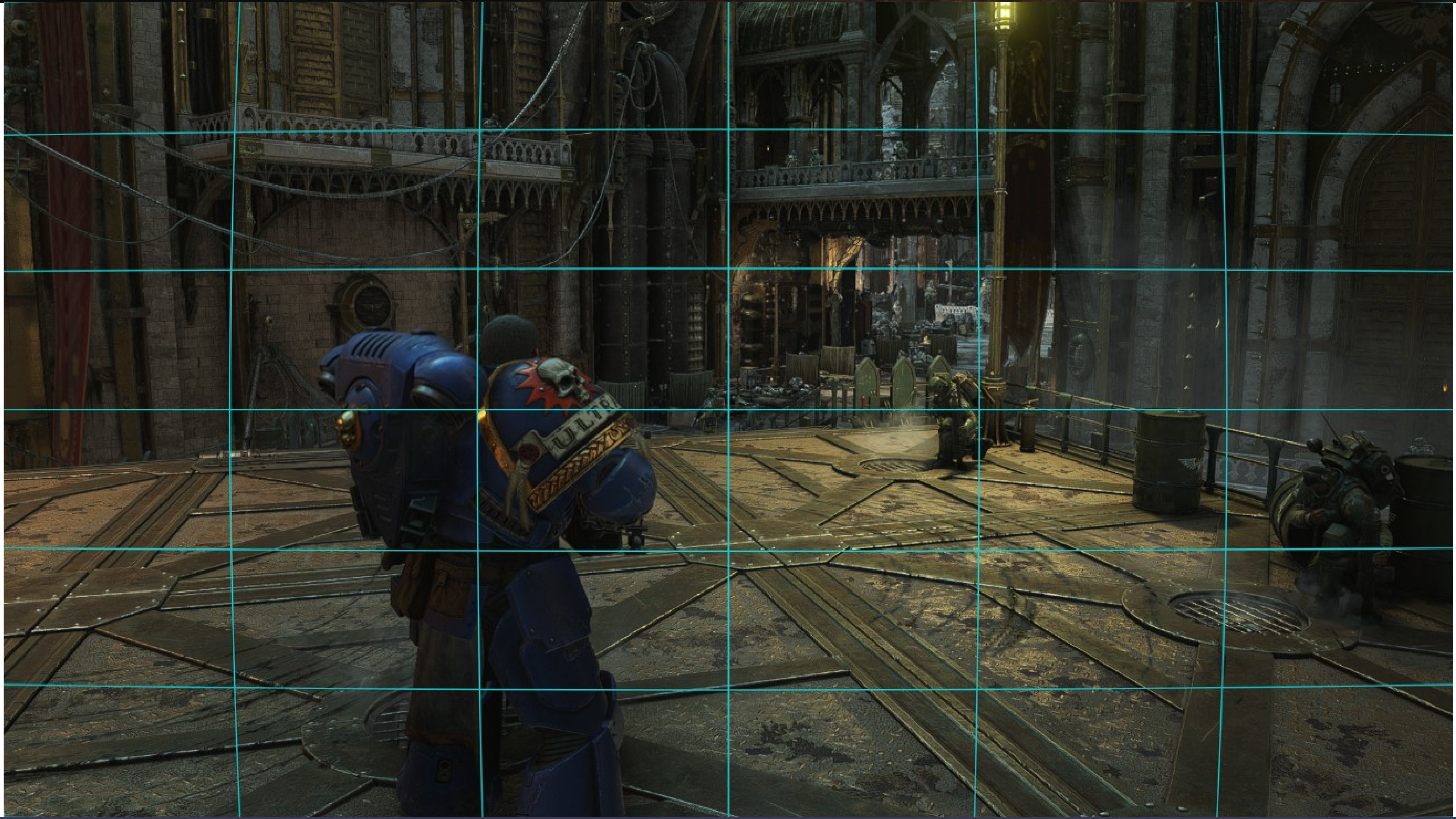
Integration is standard except for:

- We use render-rate UI, since our UI is static and we can save some performance
- We use distortion mask to avoid make up for mismatch between final color and depth & velocity introduced by some post-effects

Barrel distortion effect: before



Barrel distortion effect: after

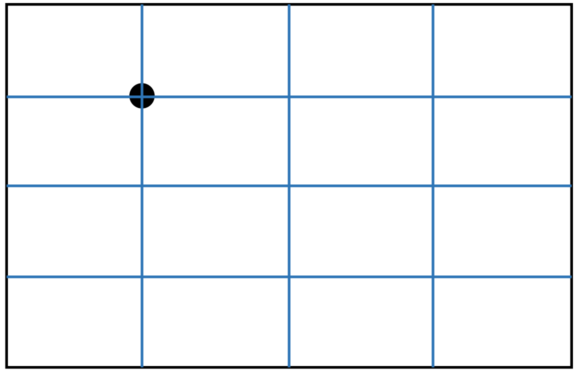


Distortion mask

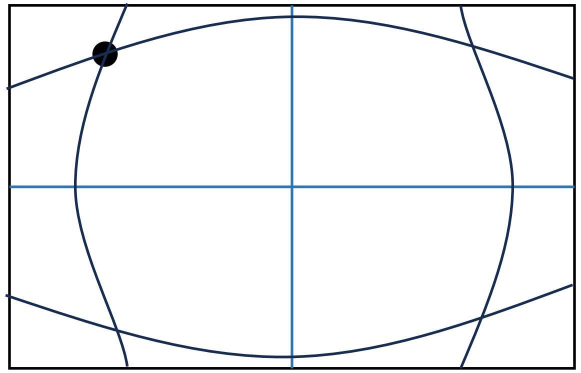
We need to reflect distortion effect in distortion mask, which stores:

- Backward distortion vector – to restore original pixel pos
- Forward distortion vector – to apply distortion again

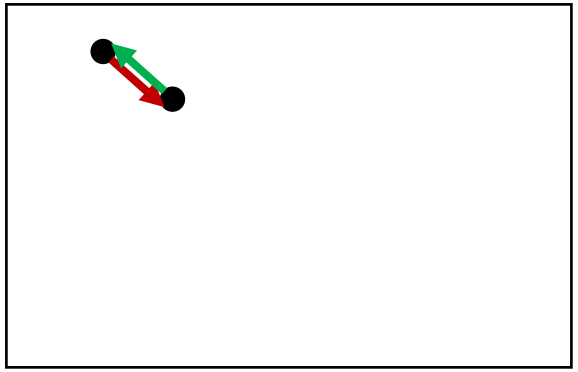
Before distortion



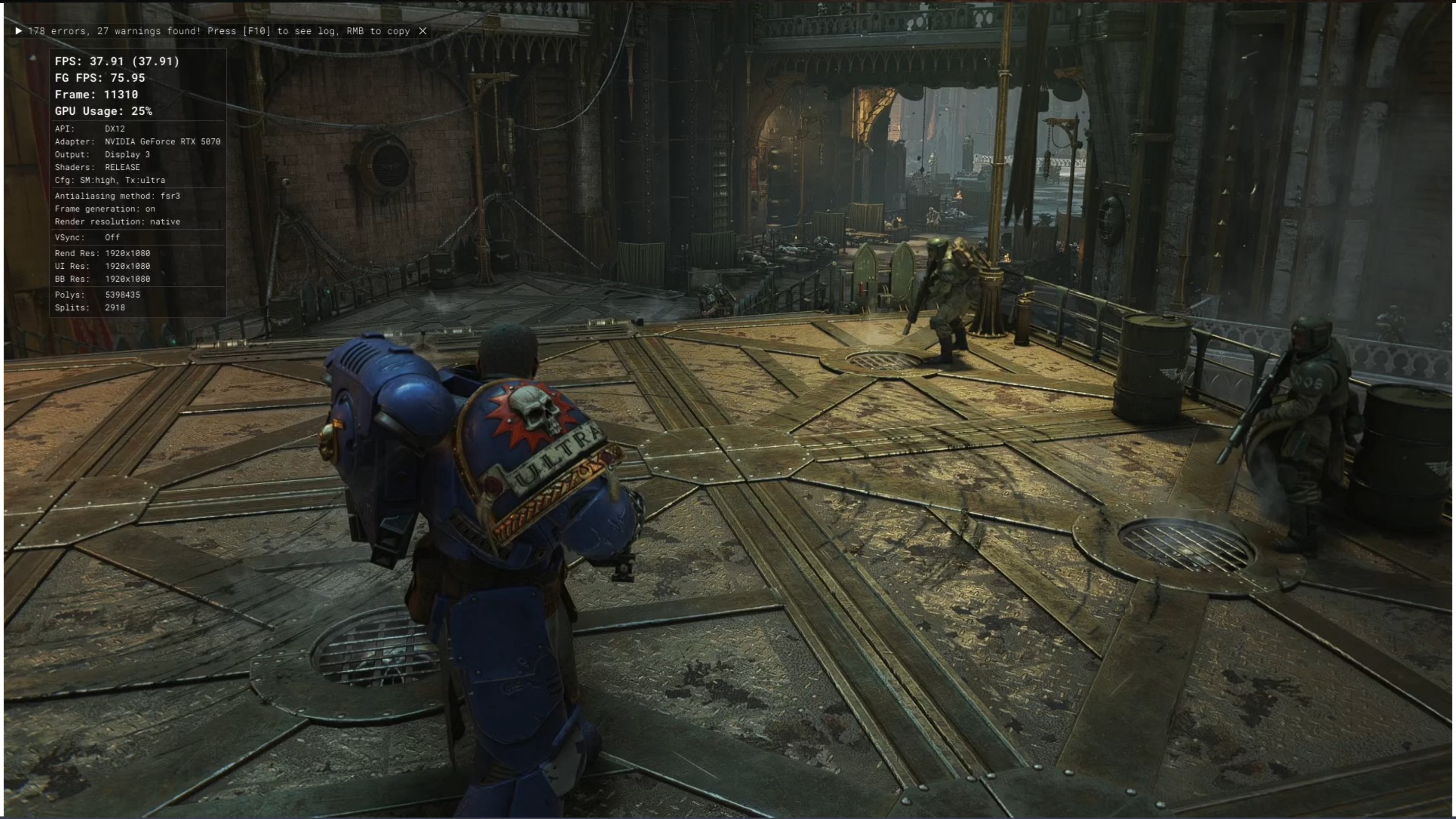
After distortion



Distortion mask

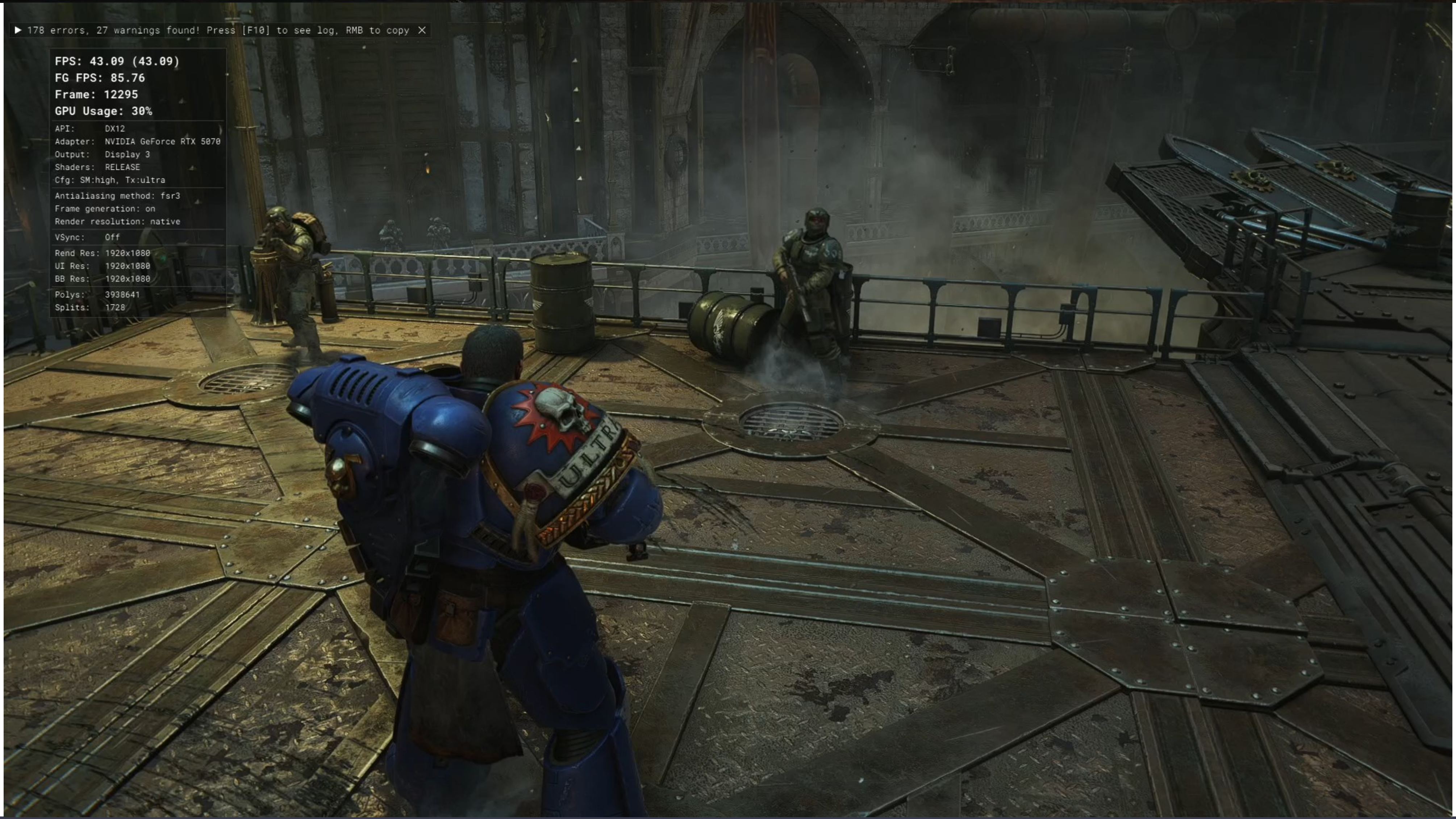


Distortion mask off



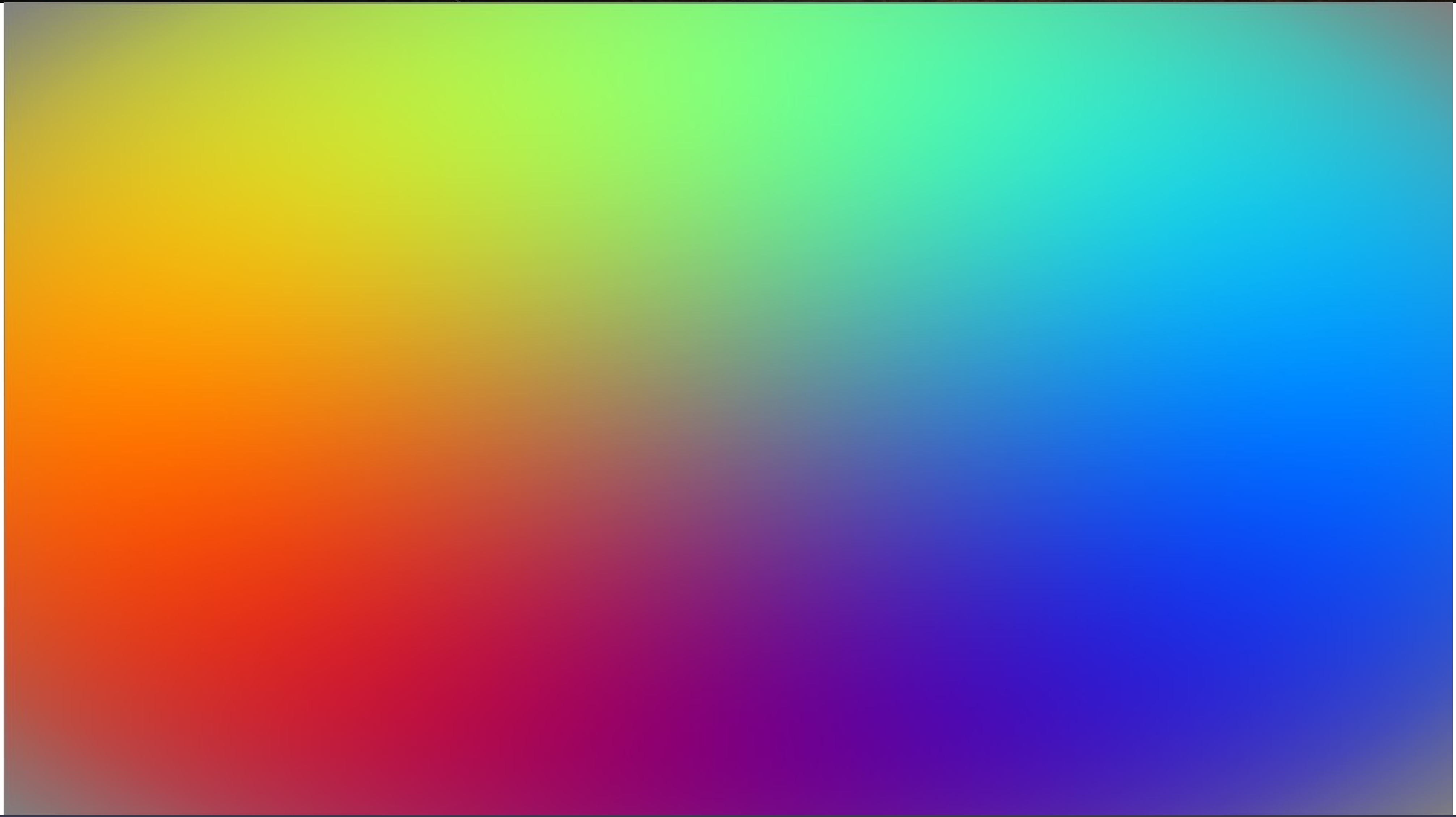
```
▶ 178 errors, 27 warnings found! Press [F10] to see log, RMB to copy ✕

FPS: 37.91 (37.91)
FG FPS: 75.95
Frame: 11310
GPU Usage: 25%
API:       DX12
Adapter:   NVIDIA GeForce RTX 5070
Output:    Display 3
Shaders:   RELEASE
Cfg: SM:high, Tx:ultra
Antialiasing method: fsr3
Frame generation: on
Render resolution: native
VSync:     Off
Rend Res:  1920x1080
UI Res:    1920x1080
BB Res:    1920x1080
Polys:     5398435
Splits:    2918
```

Distortion mask on



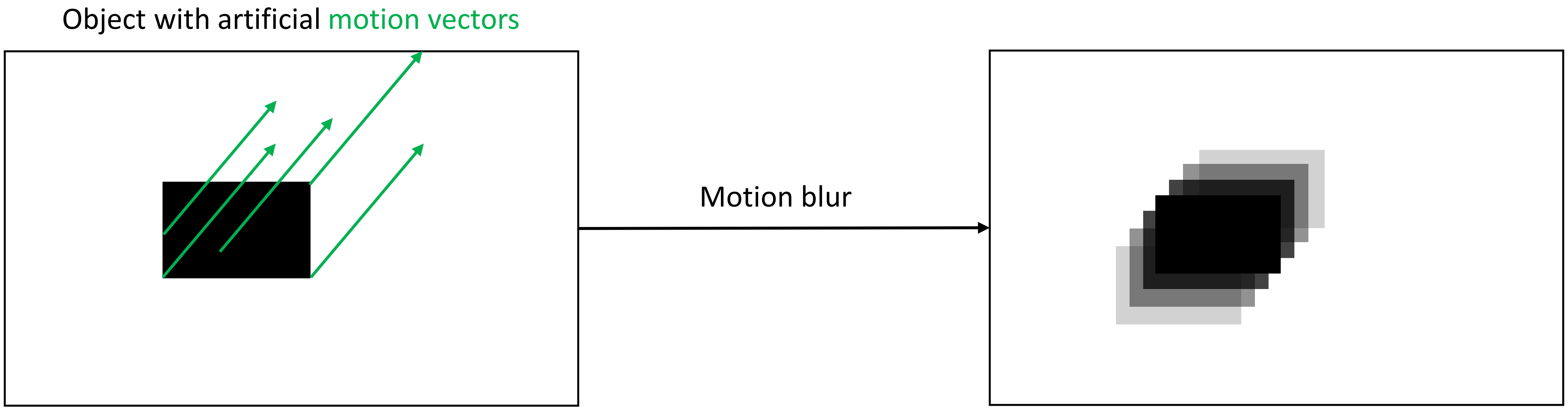
```
▶ 178 errors, 27 warnings found! Press [F10] to see log, RMB to copy ✕

FPS: 43.09 (43.09)
FG FPS: 85.76
Frame: 12295
GPU Usage: 30%
API:       DX12
Adapter:   NVIDIA GeForce RTX 5070
Output:    Display 3
Shaders:   RELEASE
Cfg: SM:high, Tx:ultra
Antialiasing method: fsr3
Frame generation: on
Render resolution: native
VSync:     Off
Rend Res:  1920x1080
UI Res:    1920x1080
BB Res:    1920x1080
Polys:     3938641
Splits:    1728
```

Distortion mask



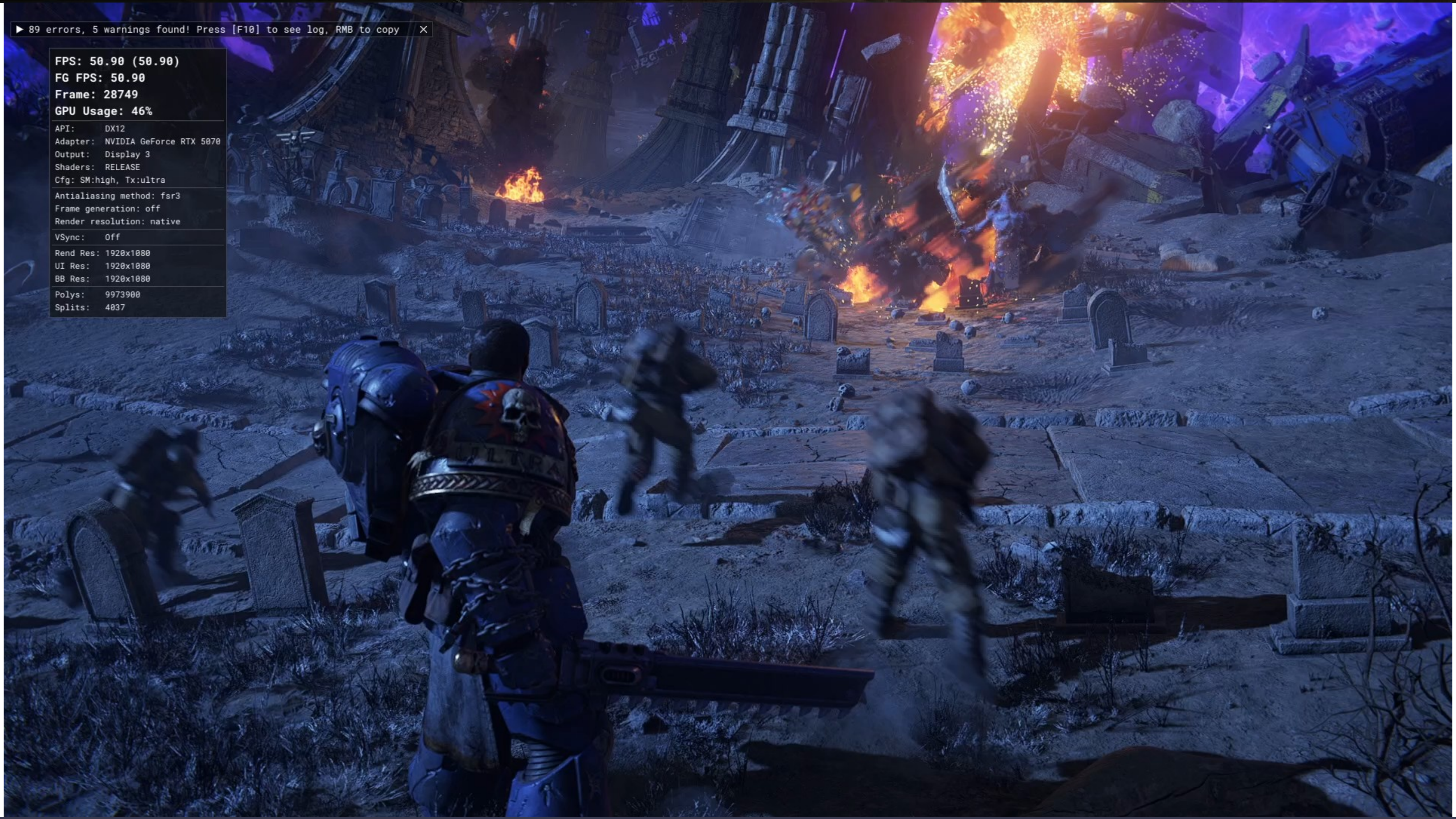
Special VFXs

For some VFXs we explicitly change motion vectors for artistic purposes, so that motion blur later would directionally blur it



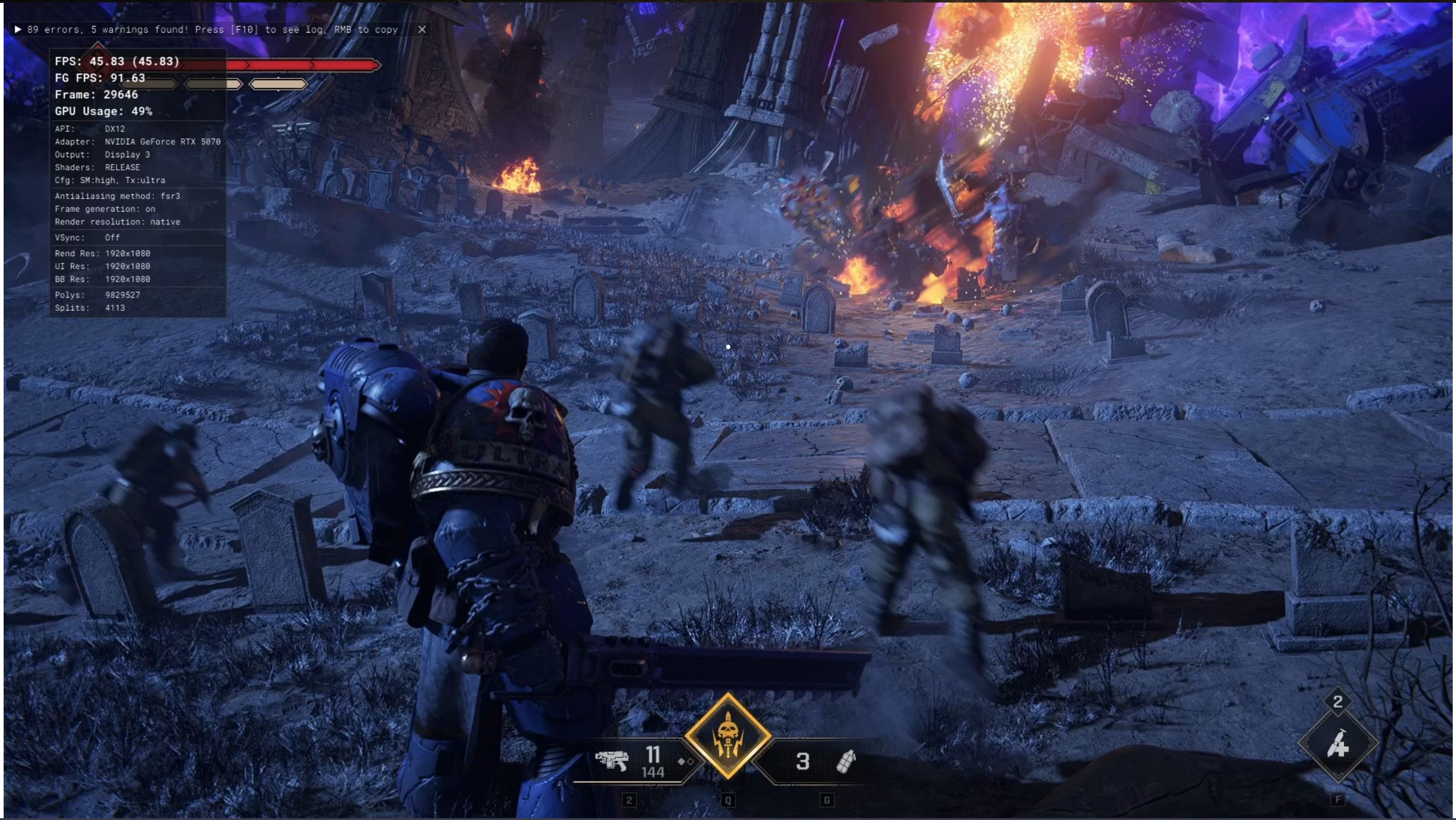
Solution: decrease amplitude of these effects

Reference (without FG)



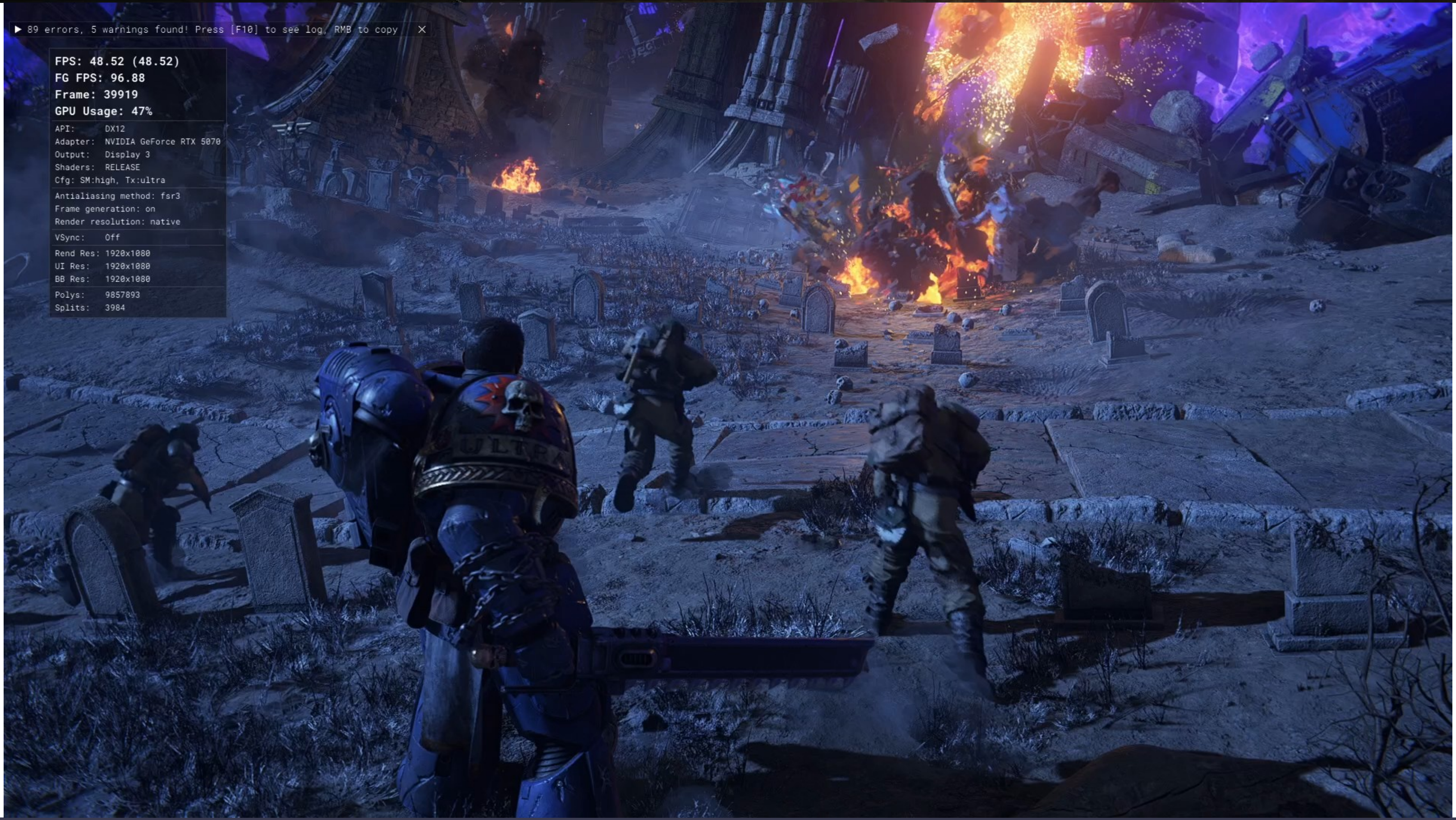
```
▶ 89 errors, 5 warnings found! Press [F10] to see log, RMB to copy  ✕

FPS: 50.90 (50.90)
FG FPS: 50.90
Frame: 28749
GPU Usage: 46%
API:       DX12
Adapter:   NVIDIA GeForce RTX 5070
Output:    Display 3
Shaders:   RELEASE
Cfg: SM:high, Tx:ultra
Antialiasing method: fsr3
Frame generation: off
Render resolution: native
VSync:     Off
Rend Res:  1920x1080
UI Res:    1920x1080
BB Res:    1920x1080
Polys:     9973900
Splits:    4037
```

FG without fixes



```
▶ 89 errors, 5 warnings found! Press [F10] to see log, RMB to copy  ×

FPS: 45.83 (45.83)
FG FPS: 91.63
Frame: 29646
GPU Usage: 49%
API:      DX12
Adapter:  NVIDIA GeForce RTX 5070
Output:   Display 3
Shaders:  RELEASE
Cfg: SM:high, Tx:ultra
Antialiasing method: fsr3
Frame generation: on
Render resolution: native
VSync:    Off
Rend Res: 1920x1080
UI Res:   1920x1080
BB Res:   1920x1080
Polys:    9829527
Splits:   4113
```

Lower amplitude



```
▶ 89 errors, 5 warnings found! Press [F10] to see log, RMB to copy  ✕

FPS: 48.52 (48.52)
FG FPS: 96.88
Frame: 39919
GPU Usage: 47%
API:      DX12
Adapter:  NVIDIA GeForce RTX 5070
Output:   Display 3
Shaders:  RELEASE
Cfg: SM:high, Tx:ultra
Antialiasing method: fsr3
Frame generation: on
Render resolution: native
VSync:    Off
Rend Res: 1920x1080
UI Res:   1920x1080
BB Res:   1920x1080
Polys:    9857893
Splits:   3984
```

Third-party tweaks

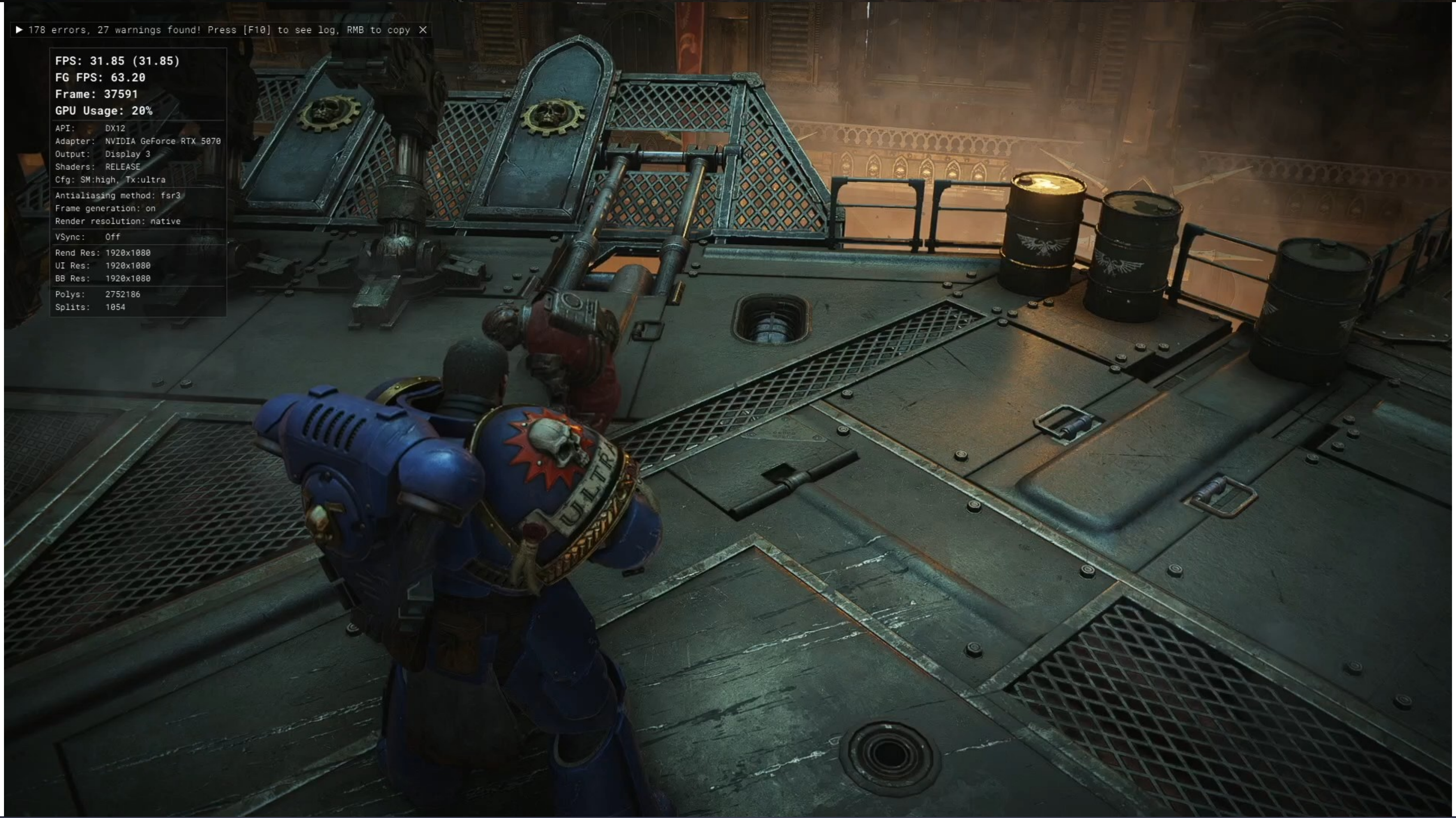
With DLSS-FG Transparent UI caused strobing and flickering

- We tweak UI alpha channel
- We turn on autodetect UI option when transparent UI covers almost all screen

With FSR-FG vignette caused artifacts near screen borders

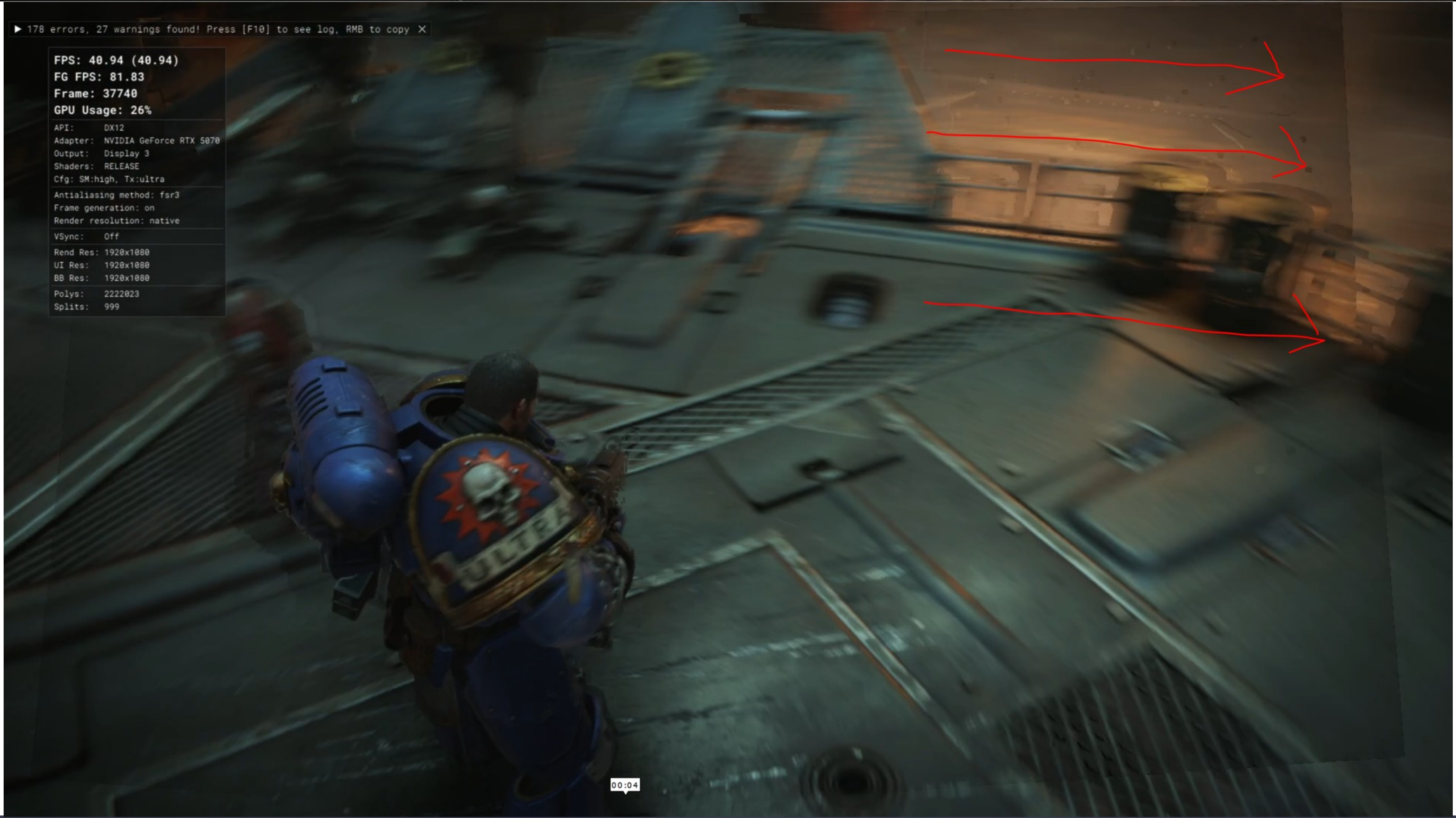
- We had to significantly lower vignette effect with FSR on
- Alternative: move vignette to FSR-FG post-process

FSR3 FG vignette bug



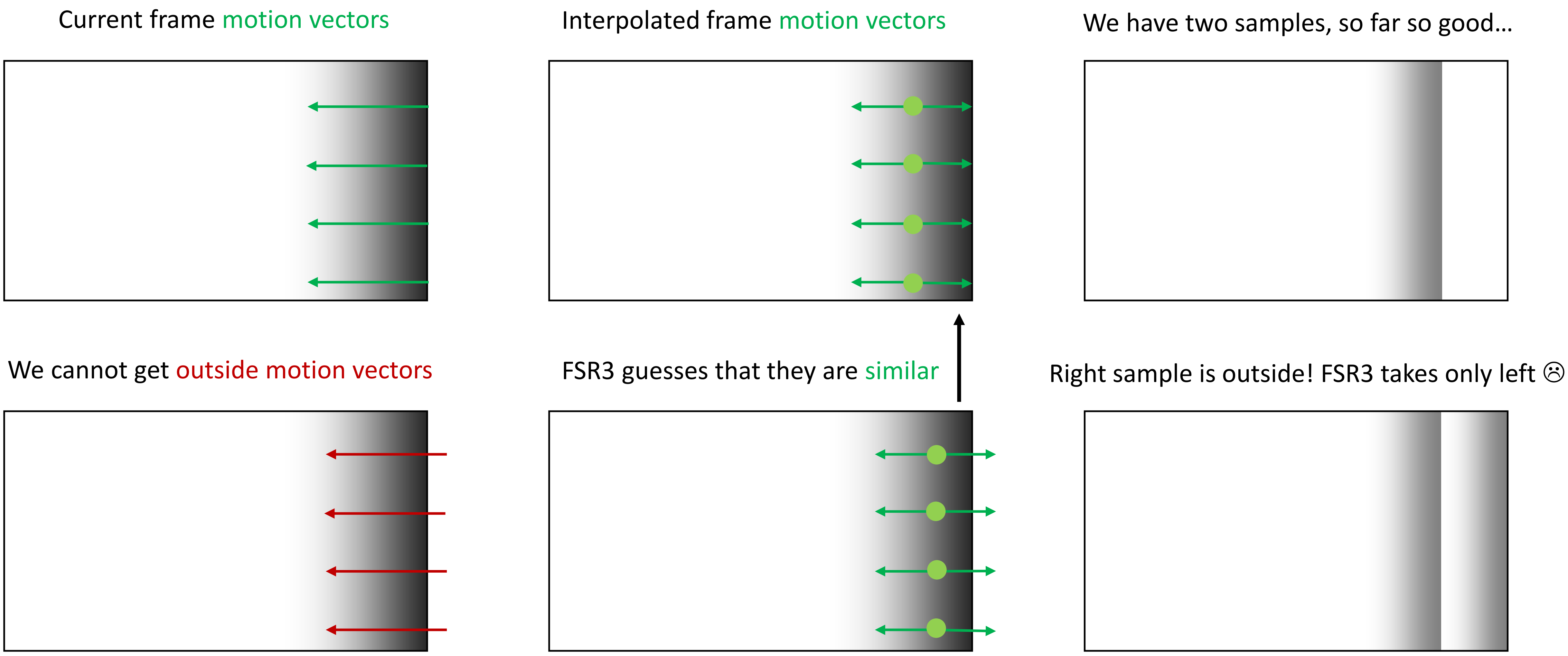
```
▶ 178 errors, 27 warnings found! Press [F10] to see log, RMB to copy ✕

FPS: 31.85 (31.85)
FG FPS: 63.20
Frame: 37591
GPU Usage: 20%
API:      DX12
Adapter:  NVIDIA GeForce RTX 5070
Output:   Display 3
Shaders:  RELEASE
Cfg: SM:high, Tx:ultra
Antialiasing method: fsr3
Frame generation: on
Render resolution: native
VSync:    Off
Rend Res: 1920x1080
UI Res:   1920x1080
BB Res:   1920x1080
Polys:    2752186
Splits:   1054
```

FSR3 FG vignette bug



```
▶ 178 errors, 27 warnings found! Press [F10] to see log, RMB to copy ✕

FPS: 40.94 (40.94)
FG FPS: 81.83
Frame: 37740
GPU Usage: 26%
API:      DX12
Adapter:  NVIDIA GeForce RTX 5070
Output:   Display 3
Shaders:  RELEASE
Cfg: SM:high, Tx:ultra
Antialiasing method: fsr3
Frame generation: on
Render resolution: native
VSync:    Off
Rend Res: 1920x1080
UI Res:   1920x1080
BB Res:   1920x1080
Polys:    2222023
Splits:   999
```

FSR3 FG vignette bug explanation



Current frame motion vectors

Interpolated frame motion vectors

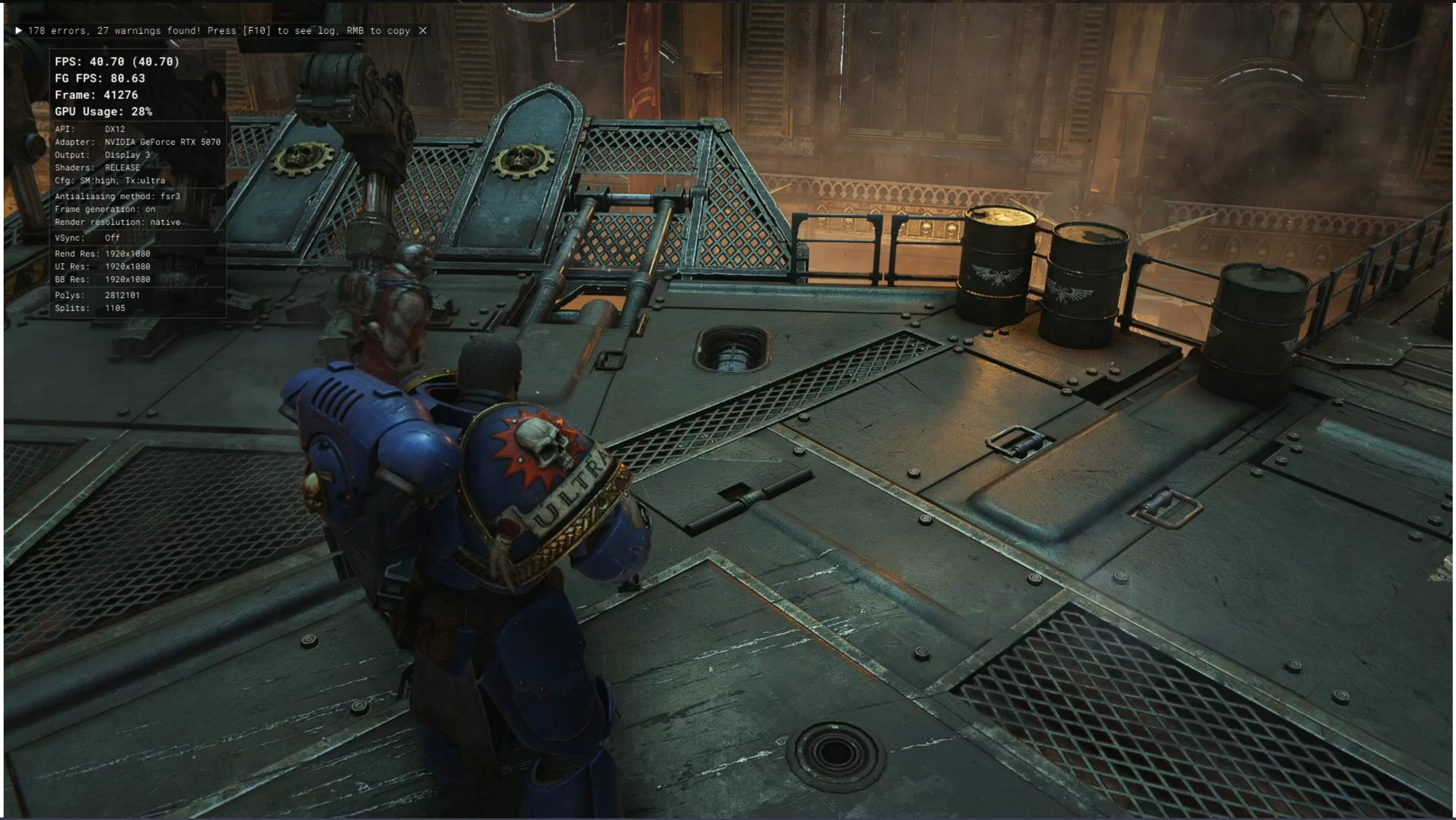
We have two samples, so far so good…

We cannot get outside motion vectors

FSR3 guesses that they are similar

Right sample is outside! FSR3 takes only left ☹

Lower intensity



```
▶ 178 errors, 27 warnings found! Press [F10] to see log, RMB to copy ✕

FPS: 40.70 (40.70)
FG FPS: 80.63
Frame: 41276
GPU Usage: 28%
API:       DX12
Adapter:   NVIDIA GeForce RTX 5070
Output:    Display 3
Shaders:   RELEASE
Cfg: SM:high, Tx:ultra
Antialiasing method: fsr3
Frame generation: on
Render resolution: native
VSync:     Off
Rend Res:  1920x1080
UI Res:    1920x1080
BB Res:    1920x1080
Polys:     2812101
Splits:    1105
```

Thank you!