XRDC

Porting Your VR Game to Oculus Quest

Lessons Learned from Porting Creed: Rise to Glory

ALEX SILKIN
Co-Founder / Chief Technology Officer
SURVIOS

Key Takeaways

1. Overcoming common pitfalls

2. Recommended workflows

3. Examples of “performant” tricks

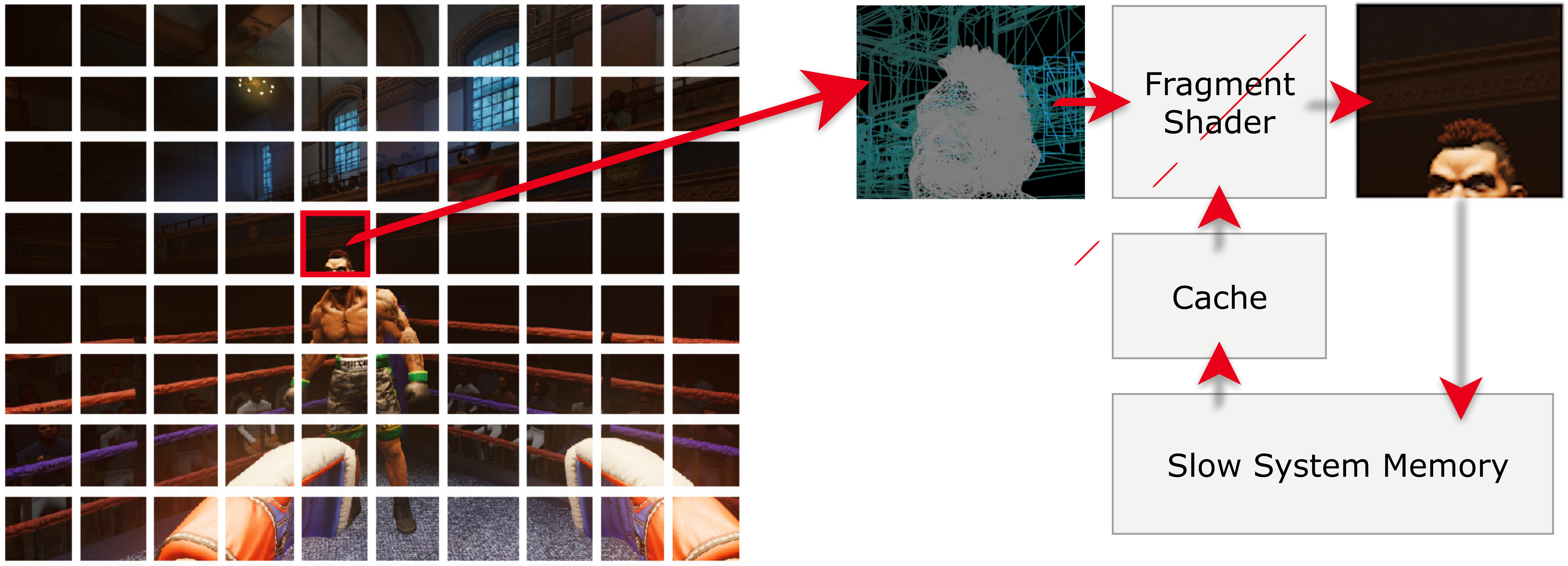
About The Game

CREED RISE TO GLORY™

Understanding the Target Hardware

Capabilities & Limitations

Tiled Based Renderer



Rough Target

- 300K Verts (*shipped with ~200K*)
- 100 - 150 Draw Calls (*shipped with ~70*)
- 13.8 ms / 72fps

Tools and Debug Environment

- OVR Metrics Tool
- RenderDoc
- SnapDragon Profiler
- Android Studio
- Set UE4 to -OpenGL -FeatureLevelES31

First Time Boot Up

Crash!!!!!????

Out of Memory?

- 2.75GB of RAM for Quest applications
- Console command “memreport -full”
- Textures were blowing us way past the budget
  - Temporarily globally clamp textures to a small size
  - Use ASTC compression

Break-up those messy reference chains!

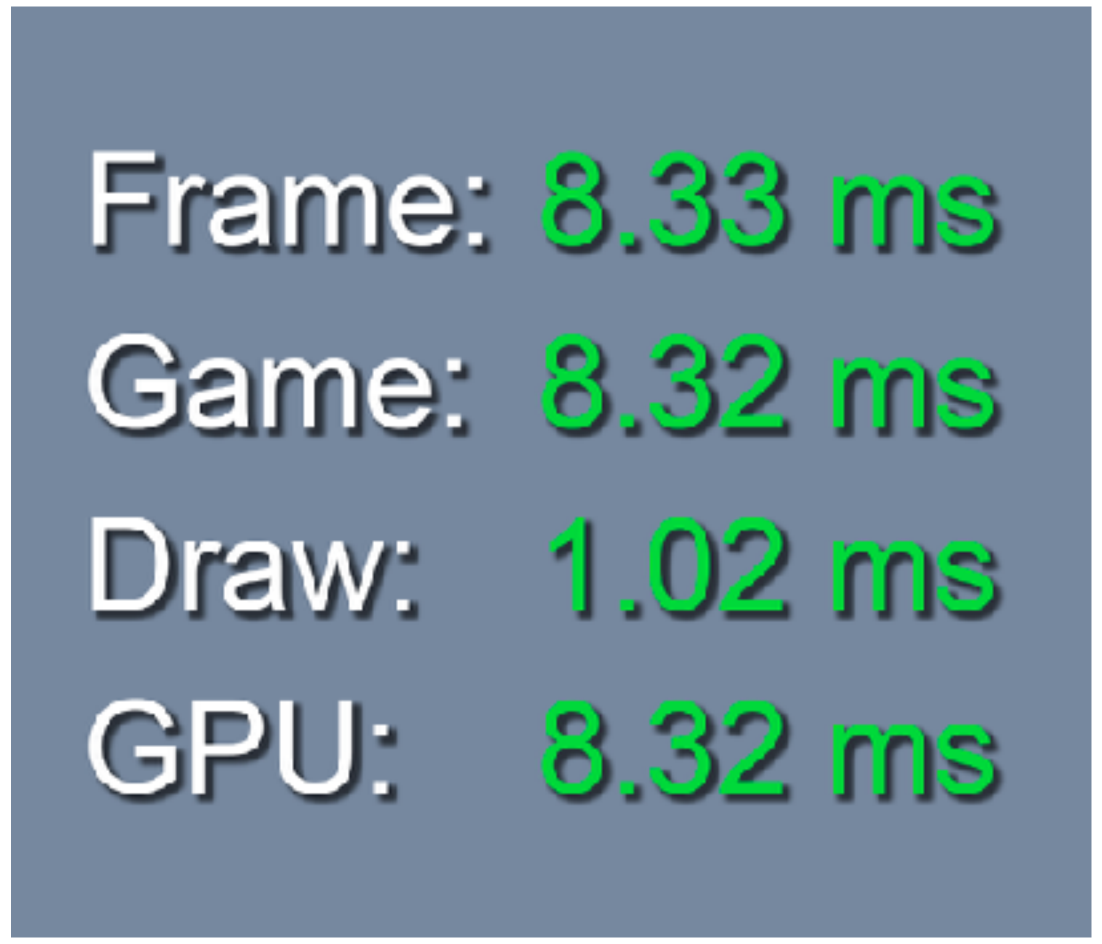
- Console command “obj refs name=[object_name]”
- UMG / UI is usual culprit
  - UI textures do not stream!
- Unnecessary hard references to assets
- Avoid blueprint dependencies
  - Casting to a BP class is hard reference - use an interface or move referenced function/variable to C++

Optimization Time

Let the fun begin!

Let's Find the Bottleneck

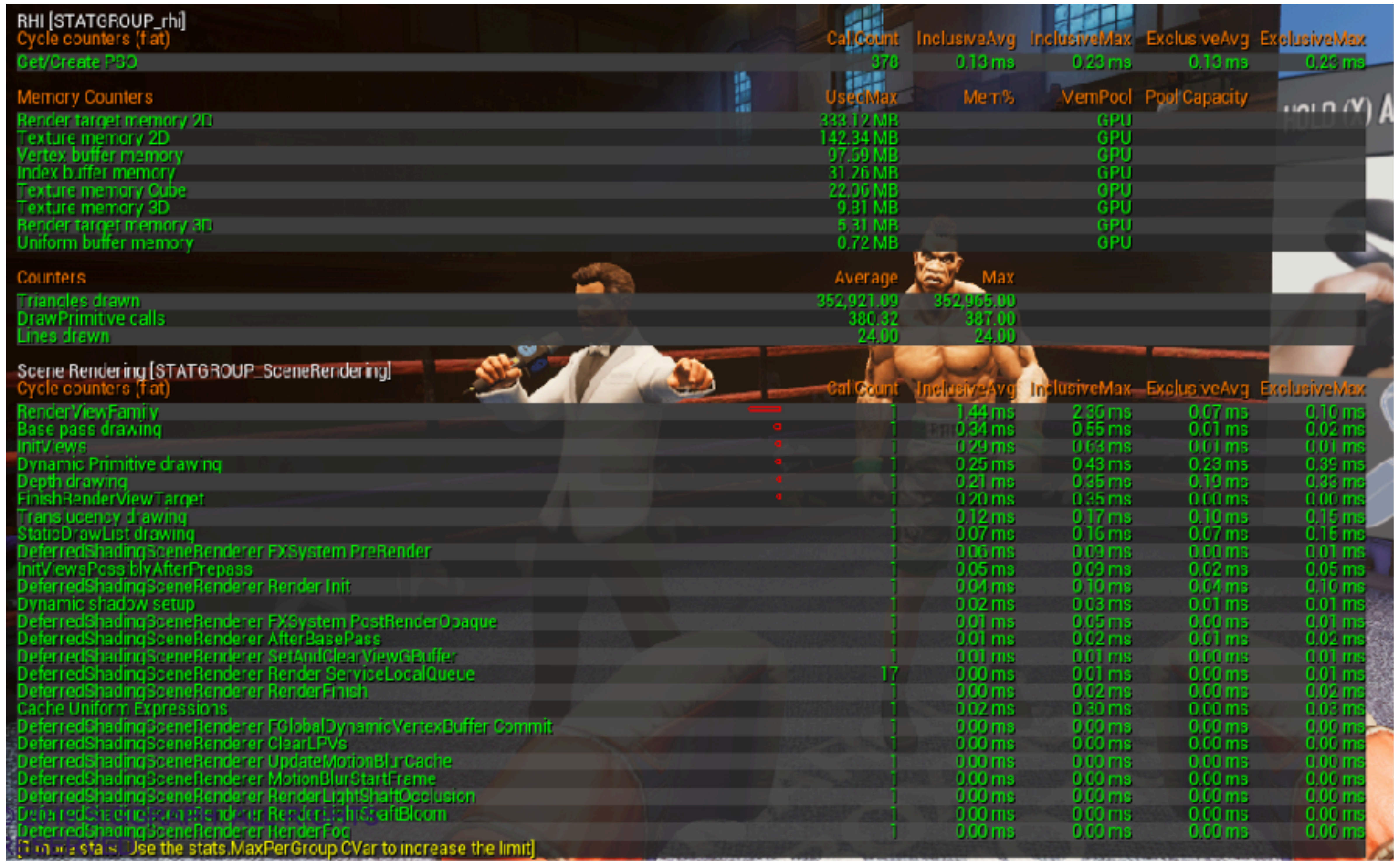
- Console command "Stat unit"
- CPU?
  - Pause the game using "pause" or keybinding
- Fill rate?
  - Decrease resolution with "r.screenpercentage .1"
- Shaders?
  - Disable material rendering with "show materials"
- Draw Calls?
  - Hide objects using "show X" like "show staticmeshes" and show "skeltalmeshes"



```
Frame: 8.33 ms
Game:  8.32 ms
Draw:  1.02 ms
GPU:   8.32 ms
```

Drilling Into the Stats

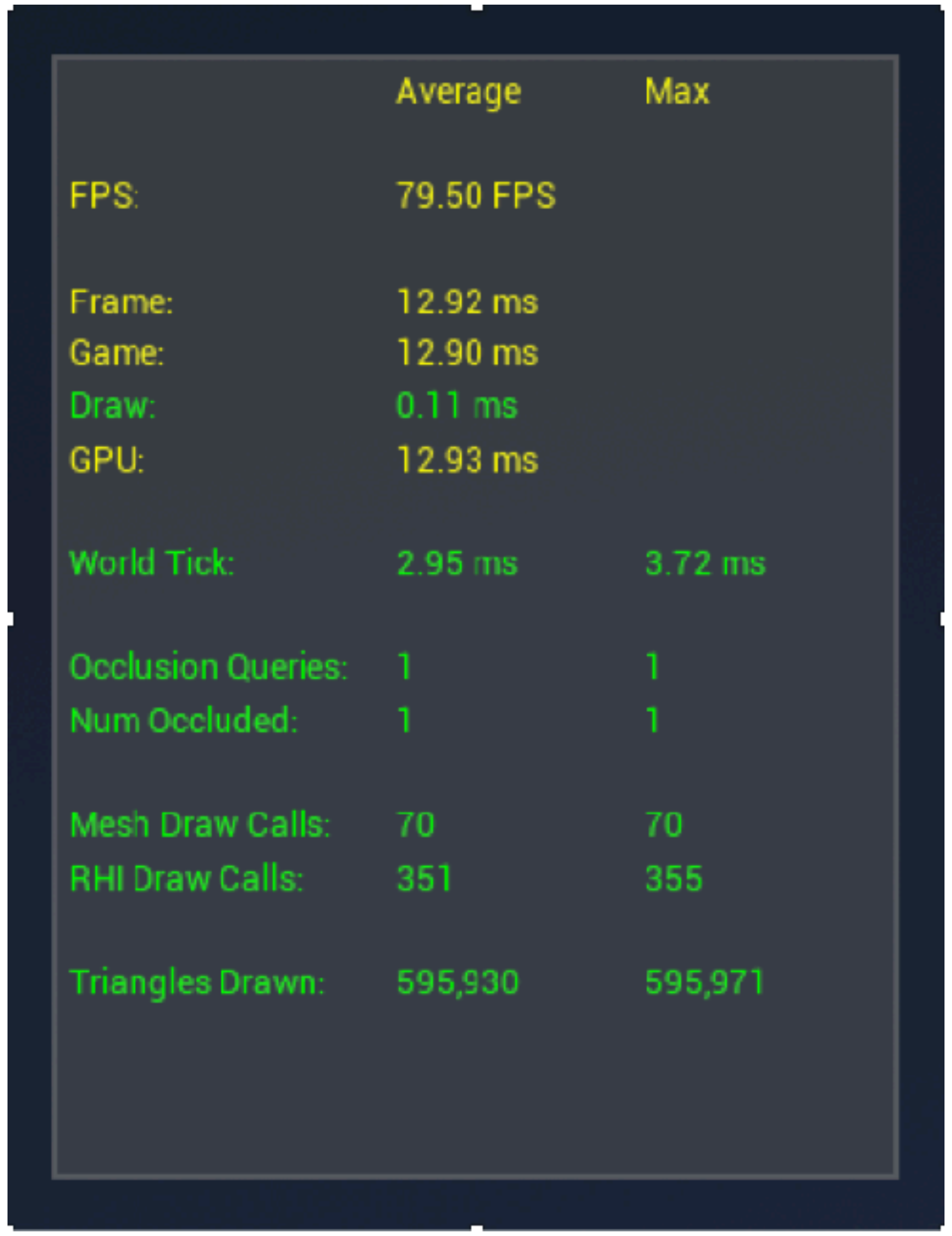
- Stat RHI
- Stat scenerendering
- Stat system overhead!
    - Stat system affects perf, distorts the numbers it reports
    - CPU overhead can push game thread beyond 13ms
    - Stat rendering increases draw calls



```
RHI [STATGROUP_rhi]
Cycle counters (flat)                              CallCount  InclusiveAvg  InclusiveMax  ExclusiveAvg  ExclusiveMax
Get/Create PSO                                           378     0.13 ms       0.23 ms       0.13 ms       0.23 ms

Memory Counters                                      UsedMax        Mem%       MemPool  Pool Capacity
Render target memory 2D                            333.12 MB                       GPU
Texture memory 2D                                  142.34 MB                       GPU
Vertex buffer memory                                97.59 MB                       GPU
Index buffer memory                                 31.26 MB                       GPU
Texture memory Cube                                 22.06 MB                       GPU
Texture memory 3D                                    9.31 MB                       GPU
Render target memory 3D                              5.31 MB                       GPU
Uniform buffer memory                                0.72 MB                       GPU

Counters                                             Average         Max
Triangles drawn                                   352,921.09  352,965.00
DrawPrimitive calls                                   380.32      387.00
Lines drawn                                            24.00       24.00

Scene Rendering [STATGROUP_SceneRendering]
Cycle counters (flat)                              CallCount  InclusiveAvg  InclusiveMax  ExclusiveAvg  ExclusiveMax
RenderViewFamily                                           1     1.44 ms       2.36 ms       0.07 ms       0.10 ms
Base pass drawing                                          1     0.34 ms       0.55 ms       0.01 ms       0.02 ms
InitViews                                                  1     0.29 ms       0.63 ms       0.01 ms       0.01 ms
Dynamic Primitive drawing                                  1     0.25 ms       0.43 ms       0.23 ms       0.39 ms
Depth drawing                                              1     0.21 ms       0.35 ms       0.19 ms       0.33 ms
FinishRenderViewTarget                                     1     0.20 ms       0.35 ms       0.00 ms       0.00 ms
Translucency drawing                                       1     0.12 ms       0.17 ms       0.10 ms       0.15 ms
StaticDrawList drawing                                     1     0.07 ms       0.16 ms       0.07 ms       0.16 ms
DeferredShadingSceneRenderer FXSystem PreRender            1     0.06 ms       0.09 ms       0.00 ms       0.01 ms
InitViewsPossiblyAfterPrepass                              1     0.05 ms       0.09 ms       0.02 ms       0.05 ms
DeferredShadingSceneRenderer Render Init                   1     0.04 ms       0.10 ms       0.04 ms       0.10 ms
Dynamic shadow setup                                       1     0.02 ms       0.03 ms       0.01 ms       0.01 ms
DeferredShadingSceneRenderer FXSystem PostRenderOpaque     1     0.01 ms       0.05 ms       0.00 ms       0.01 ms
DeferredShadingSceneRenderer AfterBasePass                 1     0.01 ms       0.02 ms       0.01 ms       0.02 ms
DeferredShadingSceneRenderer SetAndClearViewGBuffer        1     0.01 ms       0.01 ms       0.00 ms       0.01 ms
DeferredShadingSceneRenderer Render ServiceLocalQueue     17     0.00 ms       0.01 ms       0.00 ms       0.01 ms
DeferredShadingSceneRenderer RenderFinish                  1     0.00 ms       0.02 ms       0.00 ms       0.02 ms
Cache Uniform Expressions                                  1     0.02 ms       0.30 ms       0.00 ms       0.03 ms
DeferredShadingSceneRenderer FGlobalDynamicVertexBuffer Commit 1 0.00 ms      0.00 ms       0.00 ms       0.00 ms
DeferredShadingSceneRenderer ClearLPVs                     1     0.00 ms       0.00 ms       0.00 ms       0.00 ms
DeferredShadingSceneRenderer UpdateMotionBlurCache         1     0.00 ms       0.00 ms       0.00 ms       0.00 ms
DeferredShadingSceneRenderer MotionBlurStartFrame          1     0.00 ms       0.00 ms       0.00 ms       0.00 ms
DeferredShadingSceneRenderer RenderLightShaftOcclusion     1     0.00 ms       0.00 ms       0.00 ms       0.00 ms
DeferredShadingSceneRenderer RenderLightShaftBloom         1     0.00 ms       0.00 ms       0.00 ms       0.00 ms
DeferredShadingSceneRenderer RenderFog                     1     0.00 ms       0.00 ms       0.00 ms       0.00 ms
[[illegible] stats. Use the stats.MaxPerGroup CVar to increase the limit]
```

Custom Stats and Budgeting

- Lightweight slate widget
    - Rendered to a rendertexture and composed on screen using UDebugDrawService
- Average and max values
- INI defines budget for text color coding
- Existing cached engine values
    - GOcclusionQueryCount
    - GNumDrawCallsRHI
    - GNumPrimitivesDrawnRHI
- Modified engine for additional values
    - World Tick Time

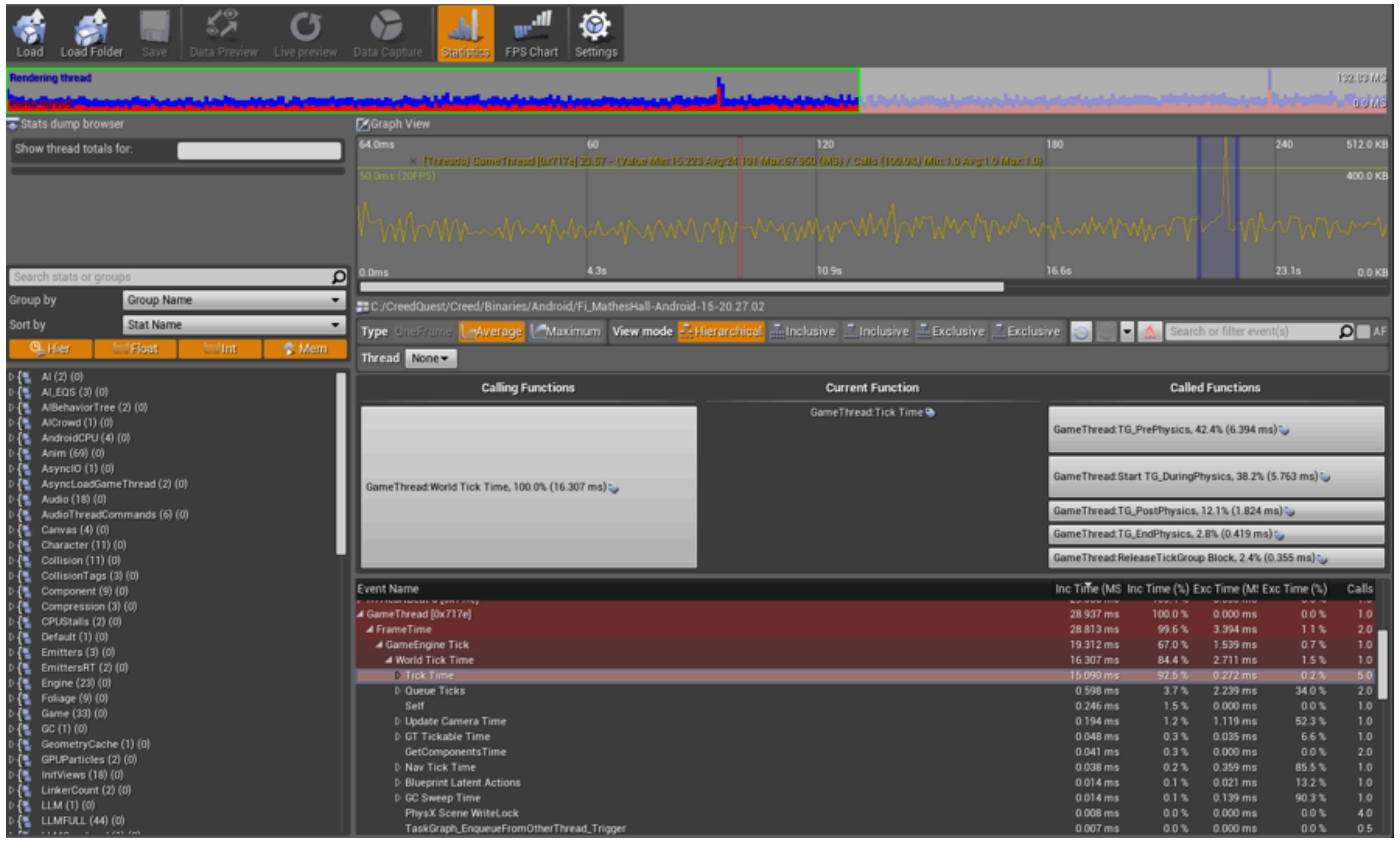


	Average	Max
FPS:	79.50 FPS	
Frame:	12.92 ms	
Game:	12.90 ms	
Draw:	0.11 ms	
GPU:	12.93 ms	
World Tick:	2.95 ms	3.72 ms
Occlusion Queries:	1	1
Num Occluded:	1	1
Mesh Draw Calls:	70	70
RHI Draw Calls:	351	355
Triangles Drawn:	595,930	595,971

CPU

- Session Fronted - “stat startfile / stat stopfile”

- Unreal Insights in 4.23+



AI (2) (0)
AI_EQS (3) (0)
AIBehaviorTree (2) (0)
AICrowd (1) (0)
AndroidCPU (4) (0)
Anim (69) (0)
AsyncIO (1) (0)
AsyncLoadGameThread (2) (0)
Audio (18) (0)
AudioThreadCommands (6) (0)
Canvas (4) (0)
Character (11) (0)
Collision (11) (0)
CollisionTags (3) (0)
Component (9) (0)
Compression (3) (0)
CPUStalls (2) (0)
Default (1) (0)
Emitters (3) (0)
EmittersRT (2) (0)
Engine (23) (0)
Foliage (9) (0)
Game (33) (0)
GC (1) (0)
GeometryCache (1) (0)
GPUParticles (2) (0)
InitViews (18) (0)
LinkerCount (2) (0)
LLM (1) (0)
LLMFULL (44) (0)

C:/CreedQuest/Creed/Binaries/Android/Fi_MathesHall-Android-15-20.27.02

Type: OneFrame, Average, Maximum — View mode: Hierarchical, Inclusive, Inclusive, Exclusive, Exclusive — Thread: None

Calling Functions	Current Function	Called Functions
GameThread:World Tick Time, 100.0% (16.307 ms)	GameThread:Tick Time	GameThread:TG_PrePhysics, 42.4% (6.394 ms)
		GameThread:Start TG_DuringPhysics, 38.2% (5.763 ms)
		GameThread:TG_PostPhysics, 12.1% (1.824 ms)
		GameThread:TG_EndPhysics, 2.8% (0.419 ms)
		GameThread:ReleaseTickGroup Block, 2.4% (0.355 ms)

Event Name	Inc Time (MS	Inc Time (%)	Exc Time (M!	Exc Time (%)	Calls
GameThread [0x717e]	28.937 ms	100.0 %	0.000 ms	0.0 %	1.0
FrameTime	28.813 ms	99.6 %	3.394 ms	1.1 %	2.0
GameEngine Tick	19.312 ms	67.0 %	1.539 ms	0.7 %	1.0
World Tick Time	16.307 ms	84.4 %	2.711 ms	1.5 %	1.0
Tick Time	15.090 ms	92.5 %	0.272 ms	0.2 %	5.0
Queue Ticks	0.598 ms	3.7 %	2.239 ms	34.0 %	2.0
Self	0.246 ms	1.5 %	0.000 ms	0.0 %	1.0
Update Camera Time	0.194 ms	1.2 %	1.119 ms	52.3 %	1.0
GT Tickable Time	0.048 ms	0.3 %	0.035 ms	6.6 %	1.0
GetComponentsTime	0.041 ms	0.3 %	0.000 ms	0.0 %	2.0
Nav Tick Time	0.038 ms	0.2 %	0.359 ms	85.5 %	1.0
Blueprint Latent Actions	0.014 ms	0.1 %	0.021 ms	13.2 %	1.0
GC Sweep Time	0.014 ms	0.1 %	0.139 ms	90.3 %	1.0
PhysX Scene WriteLock	0.008 ms	0.0 %	0.000 ms	0.0 %	4.0
TaskGraph_EnqueueFromOtherThread_Trigger	0.007 ms	0.0 %	0.000 ms	0.0 %	0.5

CPU

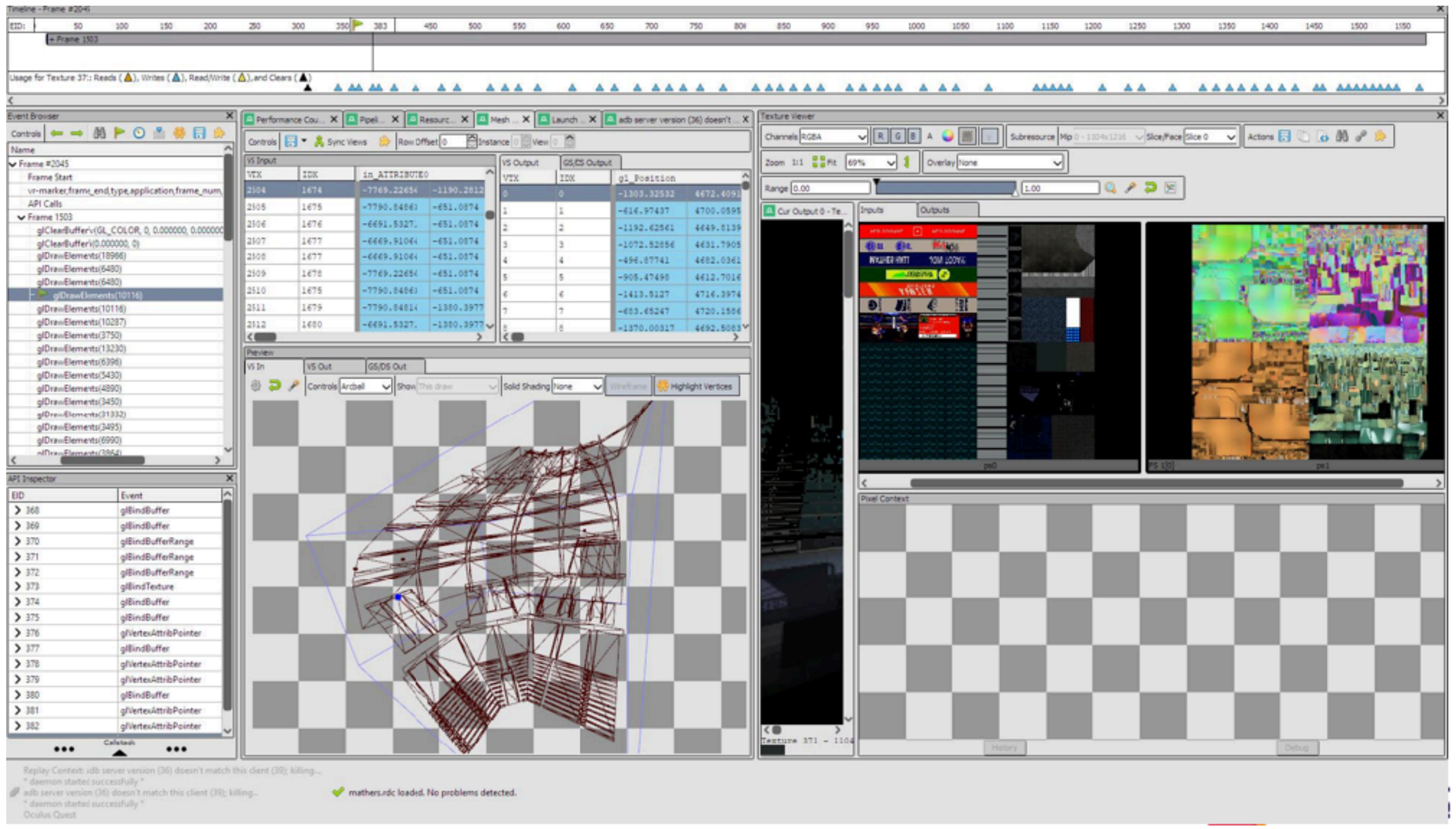
- Moving components
  - Detach hidden/disabled scene components
  - Avoid using built-in overlaps
- Ticking
  - Disable ticking when not needed
  - Tick interval
- Nativize as much BP logic as possible
- Actor pooling
- UMG Invalidation Box

Draw Calls

- Use Multi View
- Merge Geometry
    - Manually or using Unreal’s “Merge Actors”
    - Minimize material count
- Instance as much as you can
- Not all draw calls are equal!
    - Complexity of mesh affects draw call cost

Merging Level Geo

- Stadium divided in 5-7 pieces
- ~3 Shared texture atlases
- D2 Textures - Diffuse and Lightmap

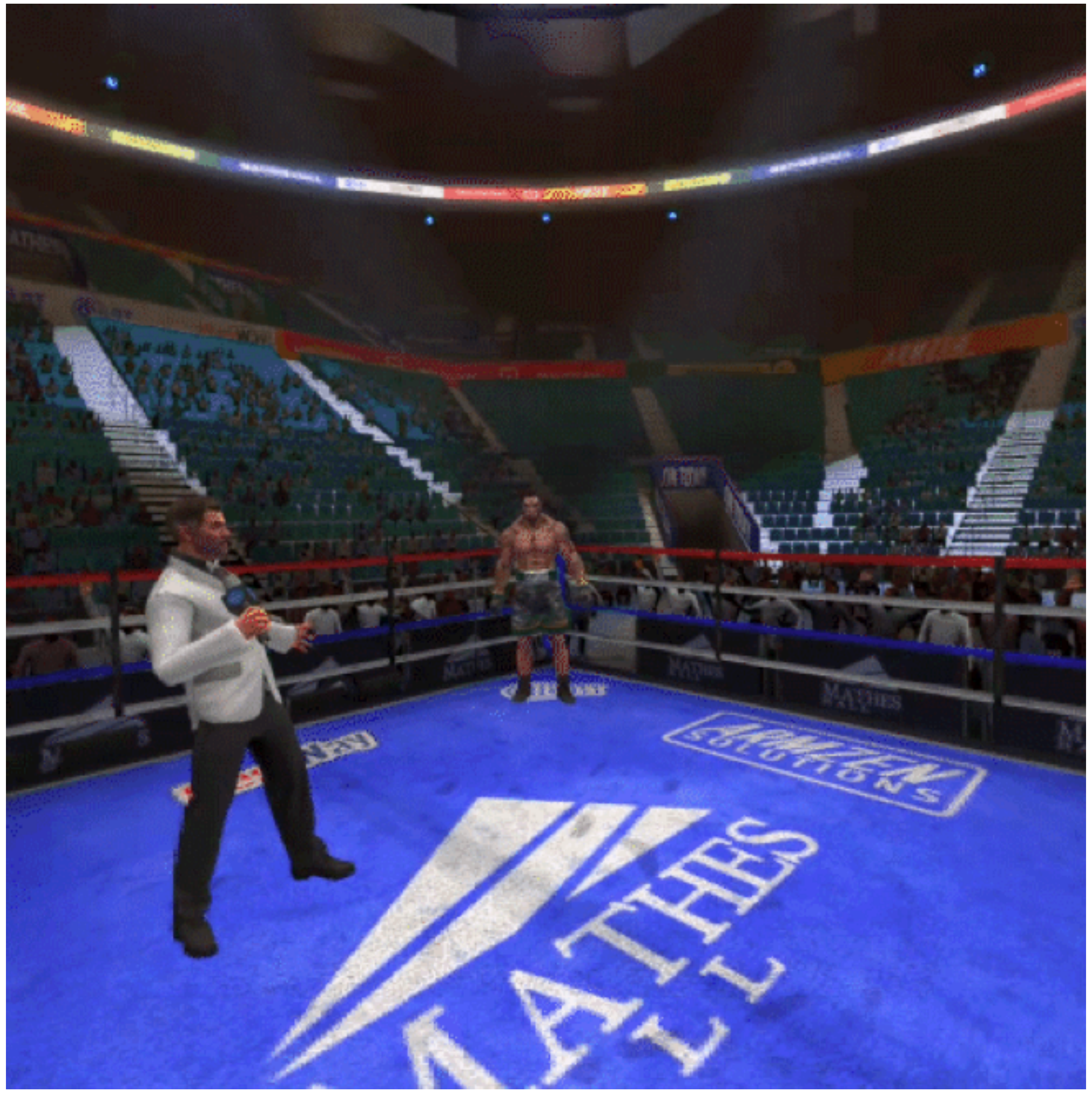

mathers.rdc loaded. No problems detected.

GPU

- Disable Early Z Pass
- Don’t use Alpha Test
- Get rid of specular
- Disable Post Processing
- Stay away from Dynamic Lights
    - Bake as much as you can

GPU

- Fake blooms and lights
    - Use transparent sprites as bloom
    - Project circular patterns in materials to stimulate spot lights



GPU

- Use texture atlases
- Minimize texture sampling (2-3 max for environment)
- ASTC Compression
- Dependent vs. Independent Texture Reads
- Texture dimensions power of 2 (mipmap + streaming)

GPU

- Use MSAA
  - Cheap(er) on Tiles Based Renderers
- Avoiding shader hitches
  - r.SaveShaderCache
  - PSO Caching on 4.21+
- Fixed Foveated Rendering
- Avoid long thin triangles

Memory

- Shared Bandwidth
    - Be careful with memory operations, they affect CPU and GPU

- Memory stomp tracking

```
adb shell setprop libc.debug.malloc 1
adb shell stop
adb shell start
adb shell setprop wrap.com.survios.Creed '"LIBC_DEBUG_MALLOC_OPTIONS=fill"'
adb shell "logcat | grep malloc"
```

Memory

```
E malloc_debug: +++ ALLOCATION 0x7ef7ba6fc0 SIZE 16 HAS A CORRUPTED
REAR GUARD
E malloc_debug:   allocation[16] = 0x04 (expected 0xbb)
E malloc_debug: Backtrace at time of failure:
E malloc_debug:           #00  pc 00000000000441b4 data/app/com.survios.Creed-1/
lib/arm64/libOVRPlugin.so (OVR::Util::CompositorVRAPI::State::Reset()+576)
```

General Tips

- Load up time
    - Load into an empty room first
    - Break reference chains
- Audio
    - AndroidAudio (Default) does not spatialize audio
    - Use AudioMixerAndroid or other third party plugins

Questions?