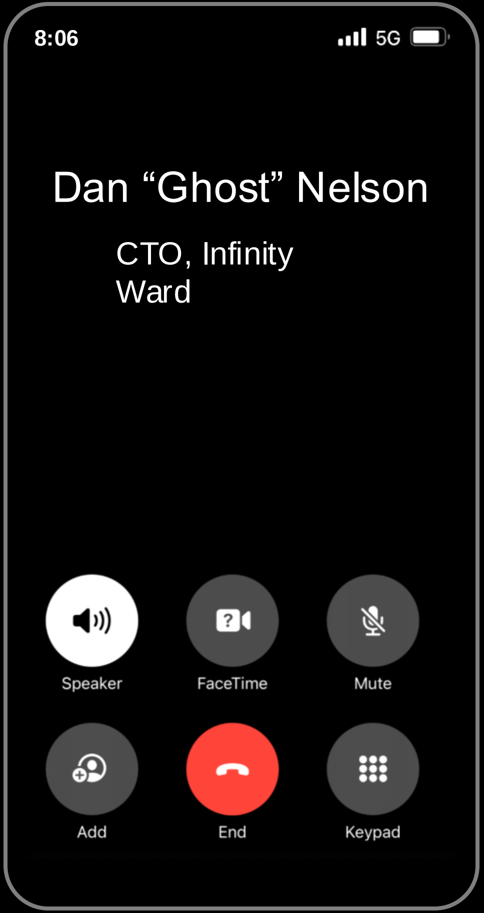
OPTIMIZING THE UI's CPU BOUND FRAMERATE IN CALL OF DUTY

OPERATION UI CODE

SIMON ESCHBACH | SLEDGEHAMMER GAMES

THE CALL TO ARMS

OPERATION UI CODE


8:06

Dan "Ghost" Nelson

CTO, Infinity Ward

HELLO?

HEY IT'S DAN. WE'VE UH..
WE'VE GOT A PROBLEM.

WHAT IS IT?

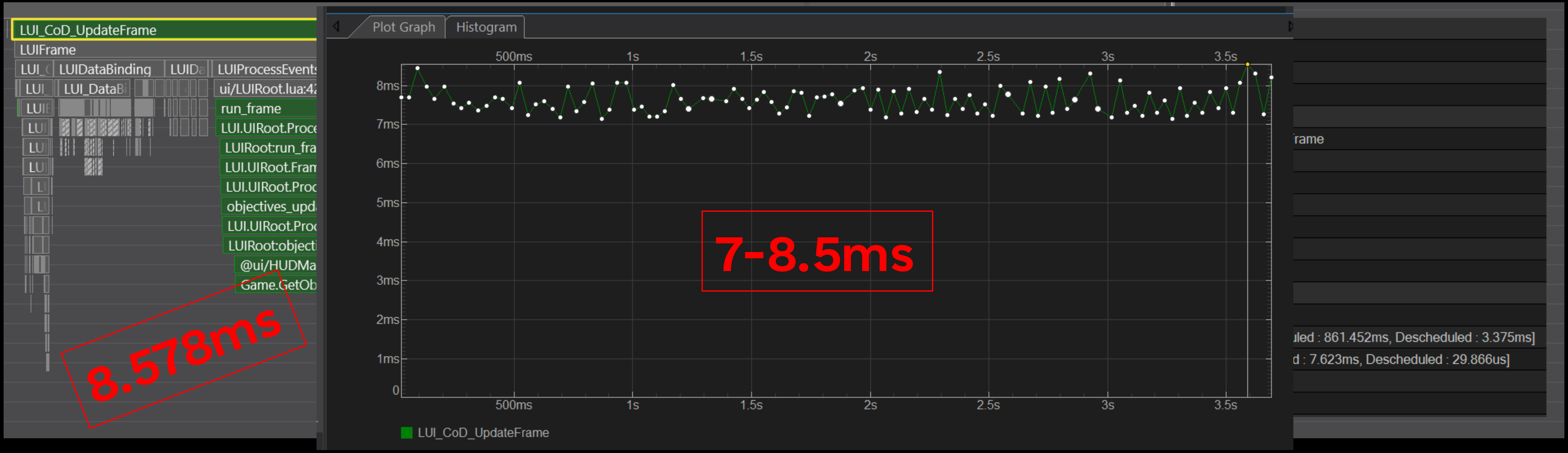
IT'S THE UI. IT'S
INFILTRATED OUR BORDERS.

ALRIGHT. I CAN HELP. BUT I'M
GOING TO NEED A TEAM.

ASSEMBLE IT.

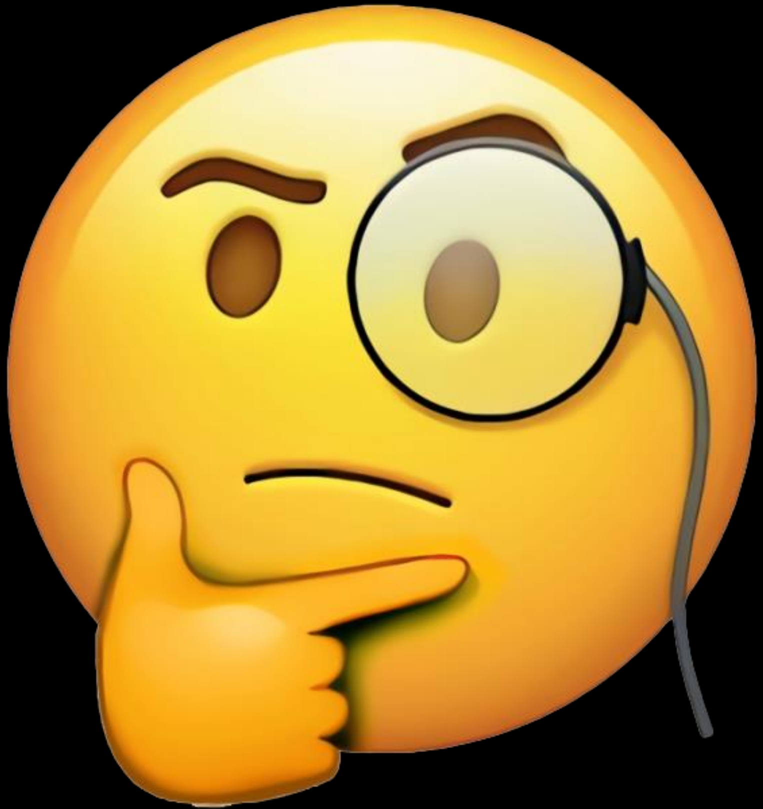
PART 1: THE STATE OF AFFAIRS

SUSTAINED HUD FRAME TIME

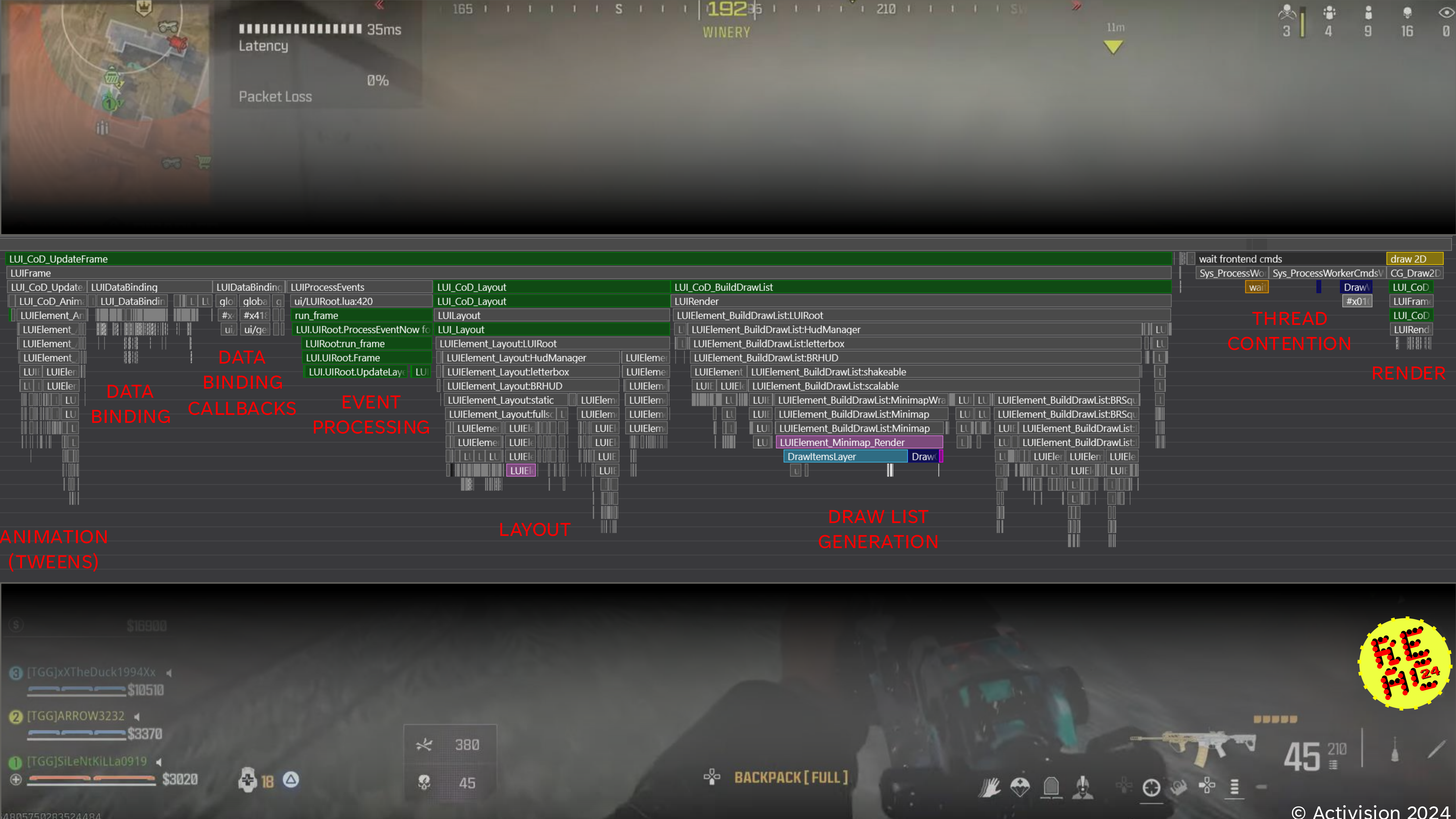


MAIN THREAD.
GROUND WAR, LOCAL CAPTURE, 25 BOTS (PS4 BASE).

HOW COULD WE SPEND SO MUCH TIME

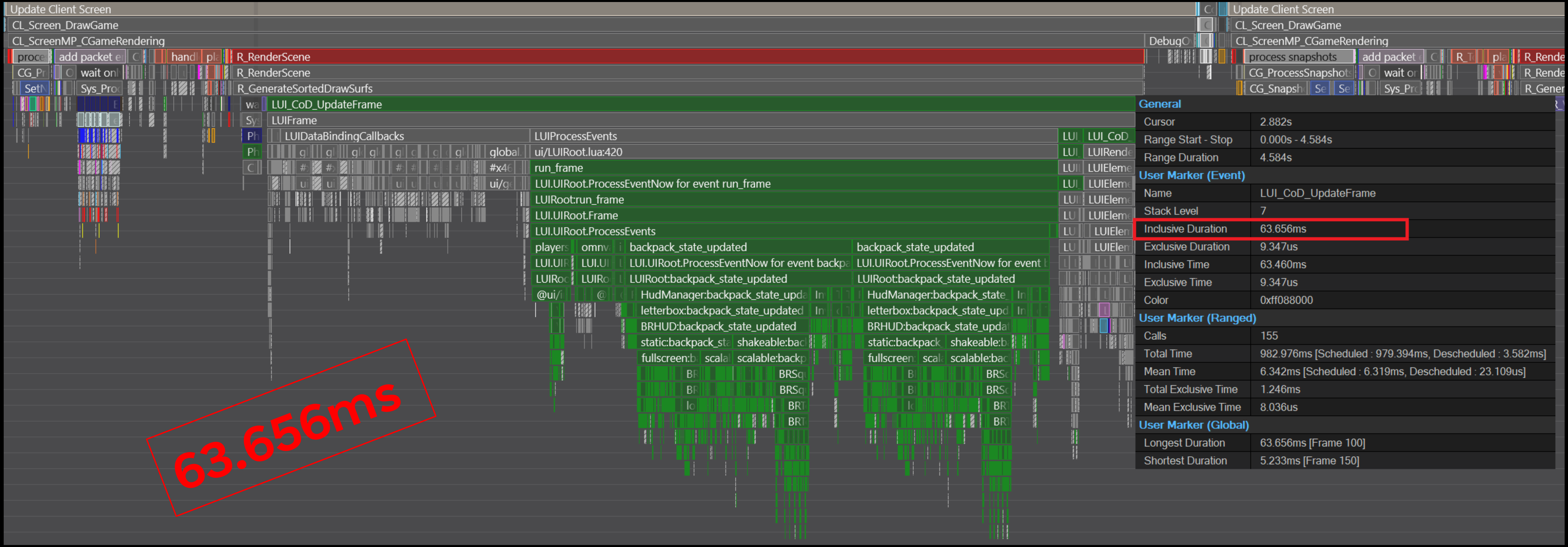


RENDERING QUADS?



LUI_CoD_UpdateFrame
LUIFrame
LUI_CoD_Update
LUIDataBinding
LUIProcessEvents
LUI_CoD_Layout
LUI_CoD_BuildDrawList
LUIRender
LUIElement_BuildDrawList:LUIRoot
LUIElement_BuildDrawList:HudManager
LUIElement_BuildDrawList:letterbox
LUIElement_BuildDrawList:BRHUD
LUIElement_BuildDrawList:shakeable
LUIElement_BuildDrawList:scalable
LUIElement_BuildDrawList:Minimap
LUIElement_Minimap_Render
DrawItemsLayer
LUILayout
LUI_Layout
LUIElement_Layout:LUIRoot
LUIElement_Layout:HudManager
LUIElement_Layout:letterbox
LUIElement_Layout:BRHUD
LUIElement_Layout:static
ui/LUIRoot.lua:420
run_frame
LUI.UIRoot.ProcessEventNow
LUIRoot:run_frame
LUI.UIRoot.Frame
LUI.UIRoot.UpdateLayout
wait frontend cmds
Sys_ProcessWorkerCmds
draw 2D
CG_Draw2D
ANIMATION (TWEENS)
DATA BINDING
DATA BINDING CALLBACKS
EVENT PROCESSING
LAYOUT
DRAW LIST GENERATION
THREAD CONTENTION
RENDER

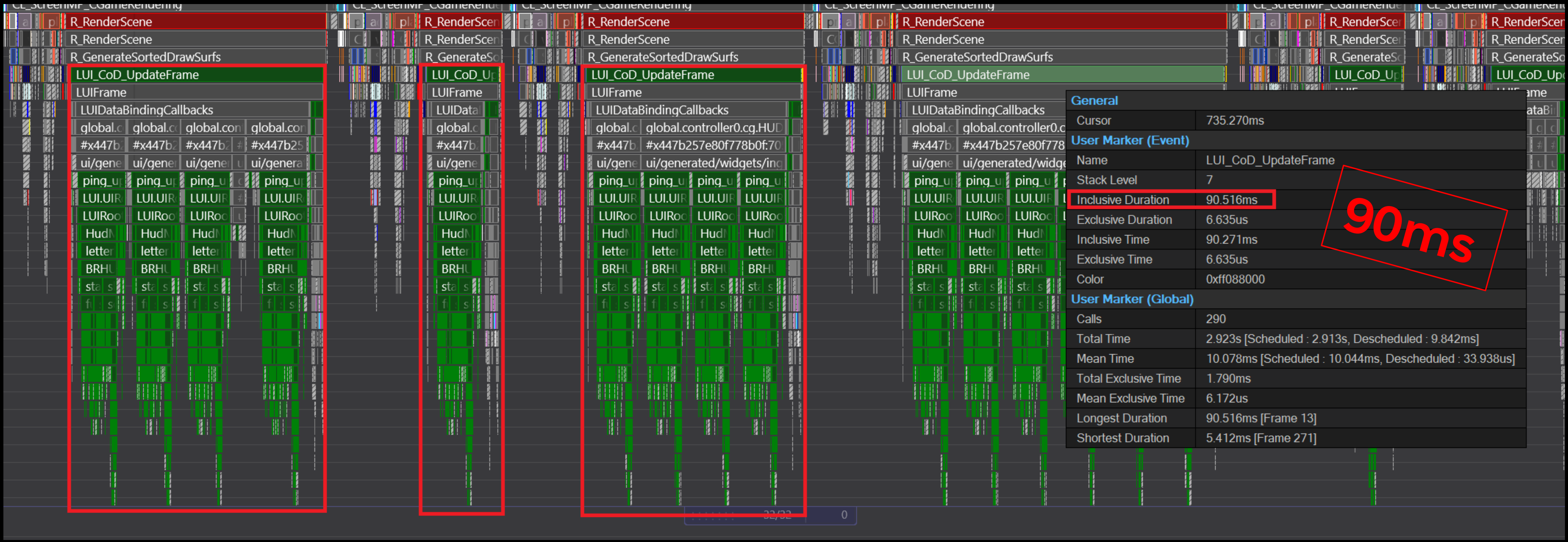
SPIKE FRAMES



General	
Cursor	2.882s
Range Start - Stop	0.000s - 4.584s
Range Duration	4.584s
User Marker (Event)	
Name	LUI_CoD_UpdateFrame
Stack Level	7
Inclusive Duration	63.656ms
Exclusive Duration	9.347us
Inclusive Time	63.460ms
Exclusive Time	9.347us
Color	0xff088000
User Marker (Ranged)	
Calls	155
Total Time	982.976ms [Scheduled : 979.394ms, Descheduled : 3.582ms]
Mean Time	6.342ms [Scheduled : 6.319ms, Descheduled : 23.109us]
Total Exclusive Time	1.246ms
Mean Exclusive Time	8.036us
User Marker (Global)	
Longest Duration	63.656ms [Frame 100]
Shortest Duration	5.233ms [Frame 150]

5 FRAMES DROPPED WHEN A PARTY MEMBER DIES.

SPIKE FRAMES IN SUCCESSION



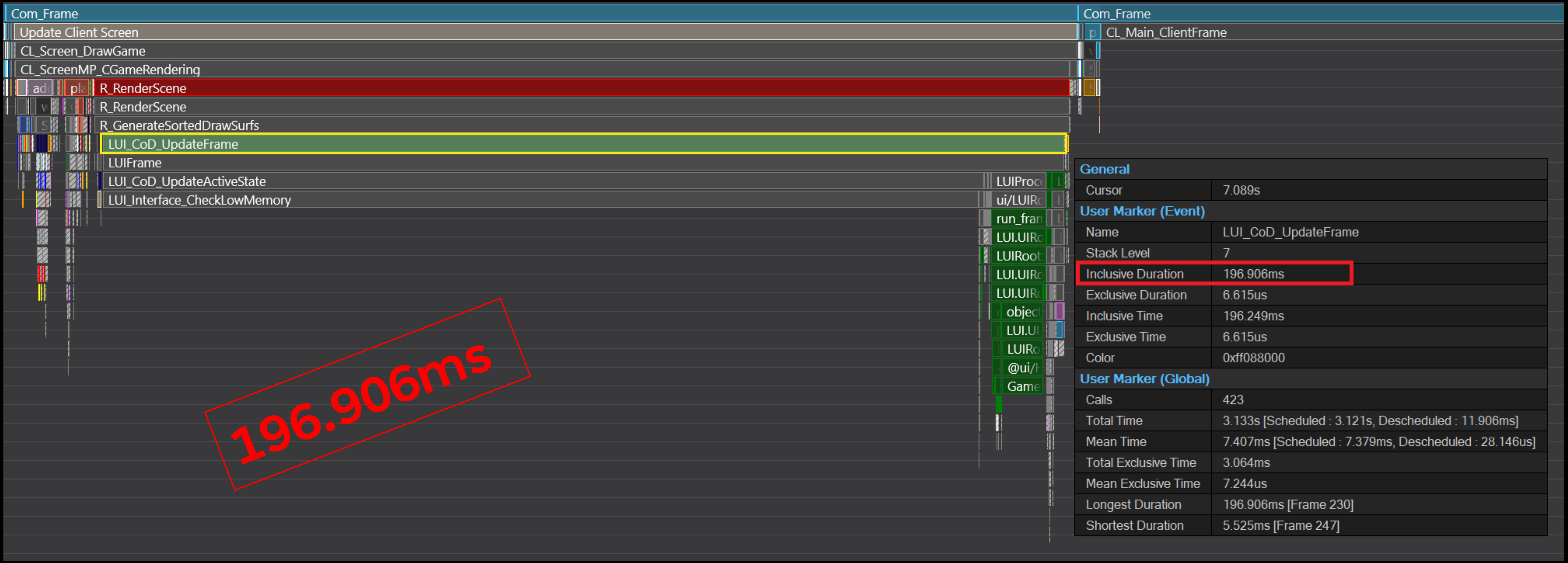
General	
Cursor	735.270ms
User Marker (Event)	
Name	LUI_CoD_UpdateFrame
Stack Level	7
Inclusive Duration	90.516ms
Exclusive Duration	6.635us
Inclusive Time	90.271ms
Exclusive Time	6.635us
Color	0xff088000
User Marker (Global)	
Calls	290
Total Time	2.923s [Scheduled : 2.913s, Descheduled : 9.842ms]
Mean Time	10.078ms [Scheduled : 10.044ms, Descheduled : 33.938us]
Total Exclusive Time	1.790ms
Mean Exclusive Time	6.172us
Longest Duration	90.516ms [Frame 13]
Shortest Duration	5.412ms [Frame 271]

6 FRAMES DROPPED OVER MULTIPLE FRAMES IN SUCCESSION WHEN PINGING

DROPPED FRAMES

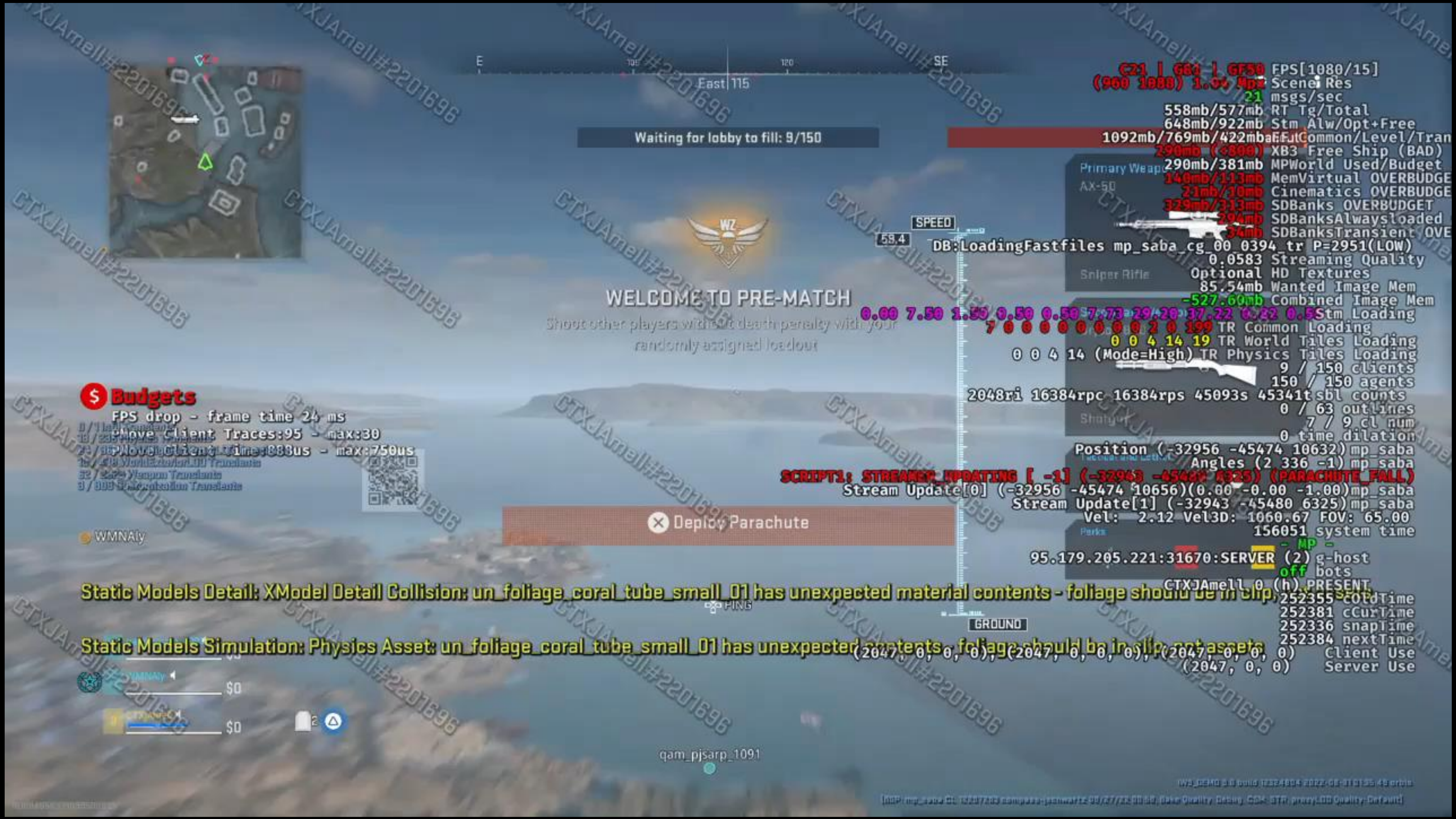
MAKE US SAD

STALL FRAMES



General	
Cursor	7.089s
User Marker (Event)	
Name	LUI_CoD_UpdateFrame
Stack Level	7
Inclusive Duration	196.906ms
Exclusive Duration	6.615us
Inclusive Time	196.249ms
Exclusive Time	6.615us
Color	0xff088000
User Marker (Global)	
Calls	423
Total Time	3.133s [Scheduled : 3.121s, Descheduled : 11.906ms]
Mean Time	7.407ms [Scheduled : 7.379ms, Descheduled : 28.146us]
Total Exclusive Time	3.064ms
Mean Exclusive Time	7.244us
Longest Duration	196.906ms [Frame 230]
Shortest Duration	5.525ms [Frame 247]

12 FRAMES DROPPED WHEN OBJECTIVES UPDATE IN WARZONE



```
Budgets
FPS drop - frame time 24 ms
PMove Client Traces:95 - max:30
PMove Client Time:888us - max:750us
```

```
Static Models Detail: XModel Detail Collision: un_foliage_coral_tube_small_01 has unexpected material contents - foliage should be in clip,
Static Models Simulation: Physics Asset: un_foliage_coral_tube_small_01 has unexpected contents, foliage should be in clip, not assets
```

```
                    C21 | G81 | GF50 FPS[1080/15]
           (960 1088) 1.04 Mpx Scene Res
                                21 msgs/sec
                     558mb/577mb RT Tg/Total
                     648mb/922mb Stm Alw/Opt+Free
          1092mb/769mb/422mb FF Common/Level/Tran
                  290mb (<800) XB3 Free Ship (BAD)
                     290mb/381mb MPWorld Used/Budget
                     140mb/113mb MemVirtual OVERBUDGE
                       21mb/10mb Cinematics OVERBUDGE
                     329mb/313mb SDBanks OVERBUDGET
                           294mb SDBanksAlwaysloaded
                            34mb SDBanksTransient OVE
DB:LoadingFastfiles mp_saba_cg_00_0394_tr P=2951(LOW)
                          0.0583 Streaming Quality
                        Optional HD Textures
                         85.54mb Wanted Image Mem
                       -527.60mb Combined Image Mem
0.00 7.50 1.50 0.50 0.50 7.73 29.20 37.22 0.22 0.5Stm Loading
      7 0 0 0 0 0 0 0 0 2 0 199 TR Common Loading
                0 0 4 14 19 TR World Tiles Loading
        0 0 4 14 (Mode=High) TR Physics Tiles Loading
                        9 / 150 clients
                      150 / 150 agents
2048ri 16384rpc 16384rps 45093s 45341t sbl counts
                          0 / 63 outlines
                           7 / 9 cl num
                               0 time dilation
           Position (-32956 -45474 10632) mp_saba
                        Angles (2 336 -1) mp_saba
SCRIPT1: STREAMER_UPDATING [ -1] (-32943 -45480 6325) (PARACHUTE_FALL)
Stream Update[0] (-32956 -45474 10656)(0.00 -0.00 -1.00)mp_saba
           Stream Update[1] (-32943 -45480 6325)mp_saba
               Vel:  2.12 Vel3D: 1060.67 FOV: 65.00
                          156051 system time
                               - MP -
      95.179.205.221:31670:SERVER (2) g-host
                              off bots
              CTXJAmell 0 (h) PRESENT
                          252355 coldTime
                          252381 cCurTime
                          252336 snapTime
                          252384 nextTime
(2047, 0, 0, 0),(2047, 0, 0, 0),(2047, 0, 0, 0)   Client Use
                          (2047, 0, 0)   Server Use
```

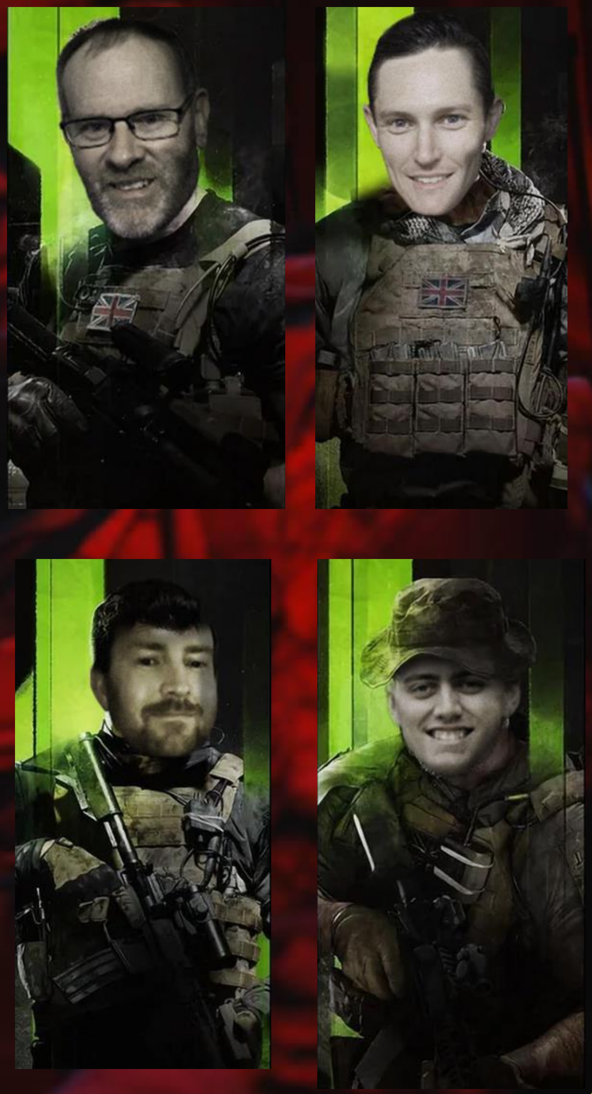
HARD STALLS

DANGEROUS FOR TELEVISIONS

PART 2: THE OFFENSIVE

CPU PERFORMANCE DEFINITIONS

SUSTAINED FRAMES:	< 10ms
SPIKE FRAMES:	10-100ms
STALL FRAMES:	> 100ms



OPTIMIZING IN A PINCH

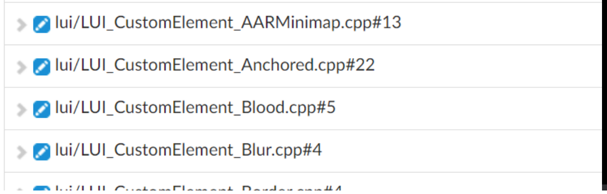
BETTER SUITED TO SUSTAINED FRAMES (<10ms)

- UI SYSTEM ANALYSIS
  - ELEMENT INVALIDATION
  - QUAD CACHING
  - ELEMENT TRAVERSAL
- HOT CODE PATH OPTIMIZATION
- GARBAGE COLLECTION TUNING
- LAZY INITIALIZATION
- STAGGERED PROCESSING

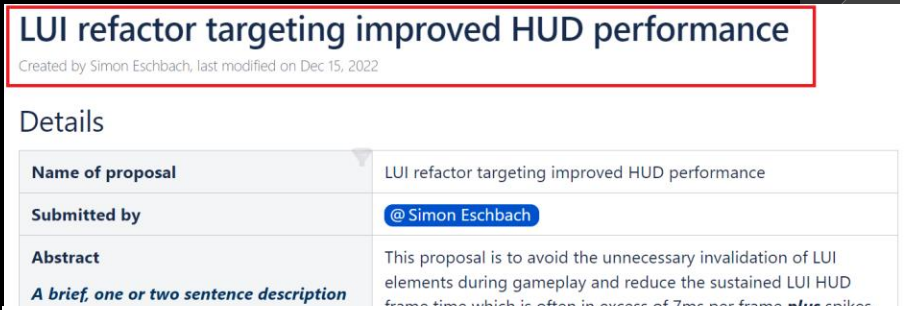
BETTER SUITED TO SPIKES AND STALLS (>10ms)

- ALGORITHMIC COMPLEXITY REDUCTION
- DATA CACHING (MEMORY TRADEOFF)
- DIRECT EVENT DISPATCHING
- FONT CACHE PRIMING
- HAND OPTIMIZATION

UI SYSTEM ANALYSIS



lui/LUI_CustomElement_AARMinimap.cpp#13
lui/LUI_CustomElement_Anchored.cpp#22
lui/LUI_CustomElement_Blood.cpp#5
lui/LUI_CustomElement_Blur.cpp#4

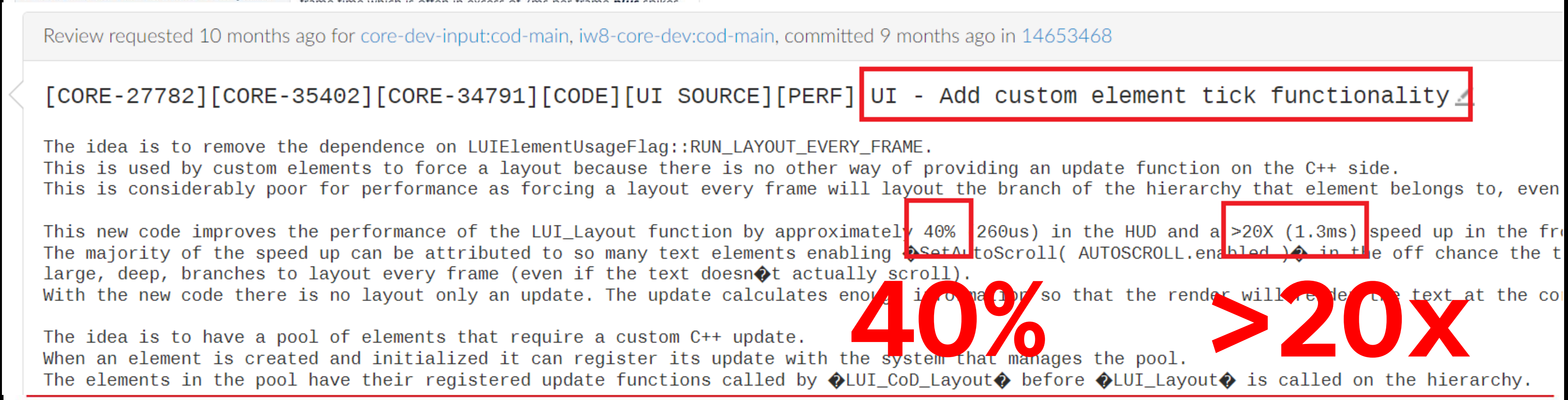


LUI refactor targeting improved HUD performance

Created by Simon Eschbach, last modified on Dec 15, 2022

Details

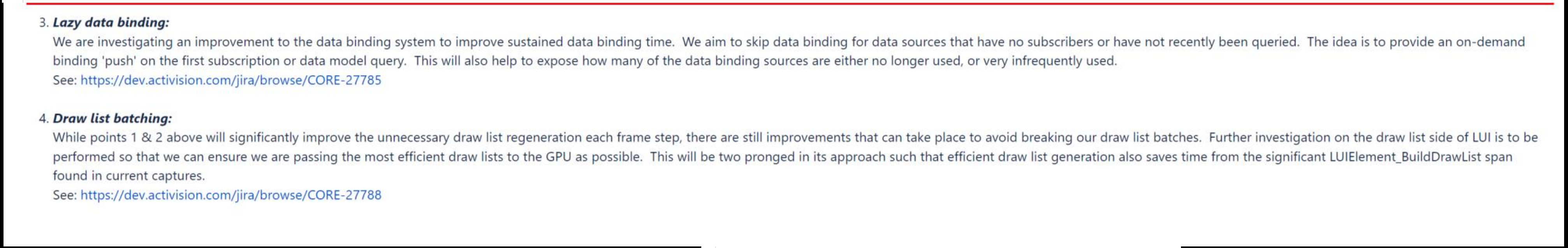
Name of proposal	LUI refactor targeting improved HUD performance
Submitted by	@Simon Eschbach
Abstract *A brief, one or two sentence description*	This proposal is to avoid the unnecessary invalidation of LUI elements during gameplay and reduce the sustained LUI HUD



Review requested 10 months ago for core-dev-input:cod-main, iw8-core-dev:cod-main, committed 9 months ago in 14653468

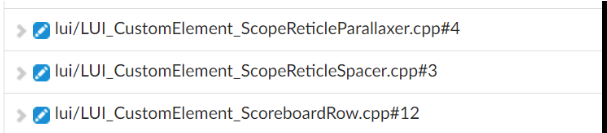
```
[CORE-27782][CORE-35402][CORE-34791][CODE][UI SOURCE][PERF] UI - Add custom element tick functionality

The idea is to remove the dependence on LUIElementUsageFlag::RUN_LAYOUT_EVERY_FRAME.
This is used by custom elements to force a layout because there is no other way of providing an update function on the C++ side.
This is considerably poor for performance as forcing a layout every frame will layout the branch of the hierarchy that element belongs to, even

This new code improves the performance of the LUI_Layout function by approximately 40% (260us) in the HUD and a >20X (1.3ms) speed up in the fr
The majority of the speed up can be attributed to so many text elements enabling �SetAutoScroll( AUTOSCROLL.enabled )� in the off chance the t
large, deep, branches to layout every frame (even if the text doesn�t actually scroll).
With the new code there is no layout only an update. The update calculates enough information so that the renderer will render the text at the co

The idea is to have a pool of elements that require a custom C++ update.
When an element is created and initialized it can register its update with the system that manages the pool.
The elements in the pool have their registered update functions called by �LUI_CoD_Layout� before �LUI_Layout� is called on the hierarchy.
```

40% **>20x**



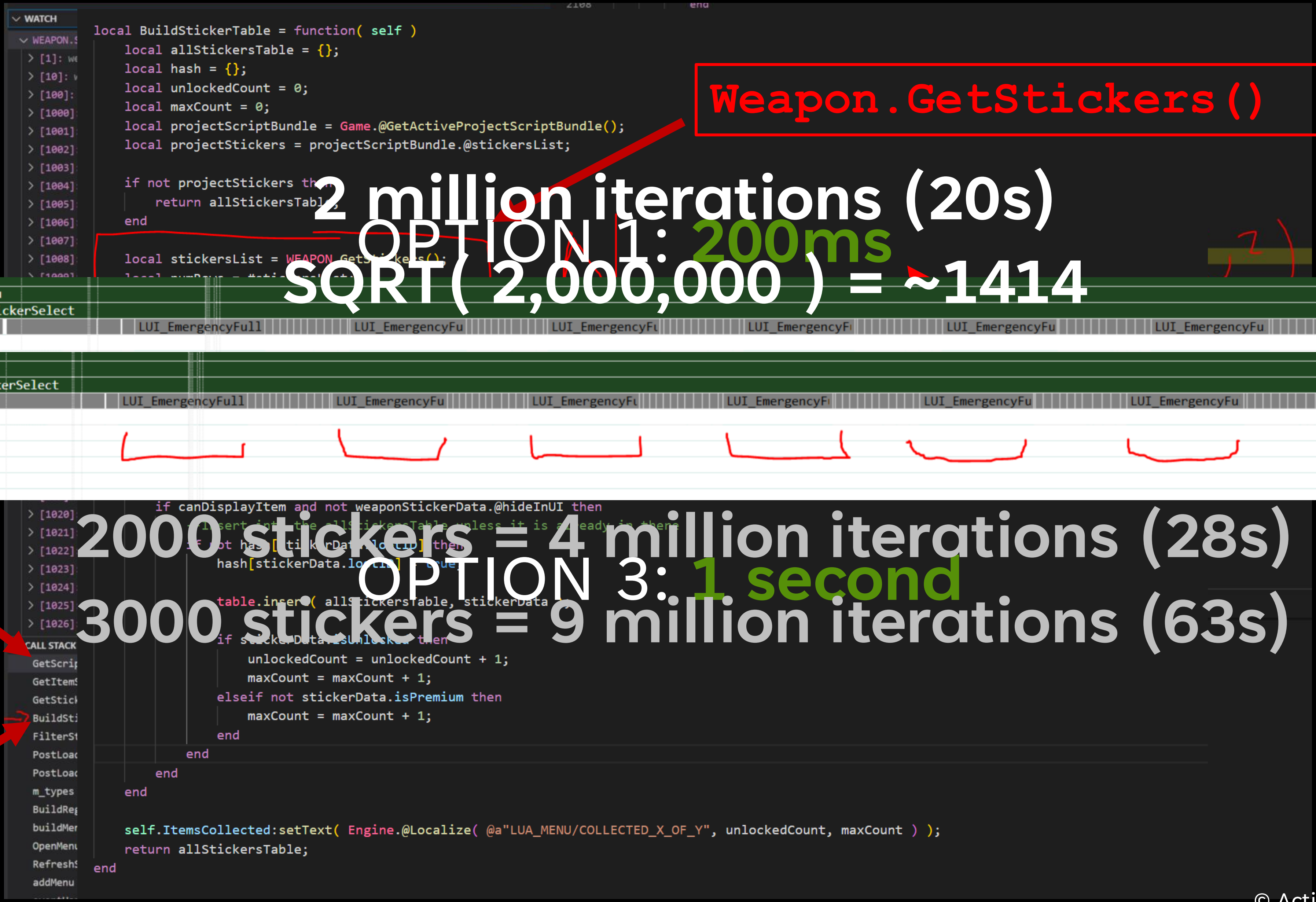
3. ***Lazy data binding:***
   We are investigating an improvement to the data binding system to improve sustained data binding time. We aim to skip data binding for data sources that have no subscribers or have not recently been queried. The idea is to provide an on-demand binding 'push' on the first subscription or data model query. This will also help to expose how many of the data binding sources are either no longer used, or very infrequently used.
   See: https://dev.activision.com/jira/browse/CORE-27785

4. ***Draw list batching:***
   While points 1 & 2 above will significantly improve the unnecessary draw list regeneration each frame step, there are still improvements that can take place to avoid breaking our draw list batches. Further investigation on the draw list side of LUI is to be performed so that we can ensure we are passing the most efficient draw lists to the GPU as possible. This will be two pronged in its approach such that efficient draw list generation also saves time from the significant LUIElement_BuildDrawList span found in current captures.
   See: https://dev.activision.com/jira/browse/CORE-27788



lui/LUI_CustomElement_ScopeReticleParallaxer.cpp#4
lui/LUI_CustomElement_ScopeReticleSpacer.cpp#3
lui/LUI_CustomElement_ScoreboardRow.cpp#12

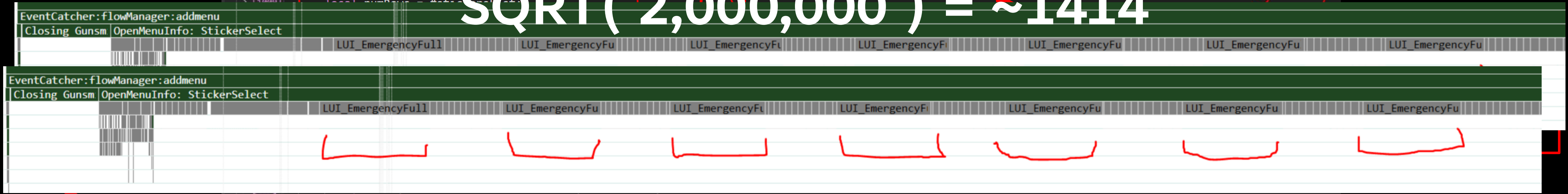
ALGORITHM COMPLEXITY REDUCTION



```
local BuildStickerTable = function( self )
    local allStickersTable = {};
    local hash = {};
    local unlockedCount = 0;
    local maxCount = 0;
    local projectScriptBundle = Game.@GetActiveProjectScriptBundle();
    local projectStickers = projectScriptBundle.@stickersList;

    if not projectStickers th
        return allStickersTabl
    end

    local stickersList = WEAPON.GetStickers();
```

```
        if canDisplayItem and not weaponStickerData.@hideInUI then
            -- Insert into the allStickersTable unless it is already in there
            if not hash[stickerData.lootID] then
                hash[stickerData.lootID] = true;

                table.insert( allStickersTable, stickerData );

                if stickerData.isUnlocked then
                    unlockedCount = unlockedCount + 1;
                    maxCount = maxCount + 1;
                elseif not stickerData.isPremium then
                    maxCount = maxCount + 1;
                end
            end
        end
    end

    self.ItemsCollected:setText( Engine.@Localize( @a"LUA_MENU/COLLECTED_X_OF_Y", unlockedCount, maxCount ) );
    return allStickersTable;
end
```

Weapon.GetStickers()



2 million iterations (20s)

OPTION 1: 200ms

SQRT(2,000,000) = ~1414

2000 stickers = 4 million iterations (28s)

OPTION 3: 1 second

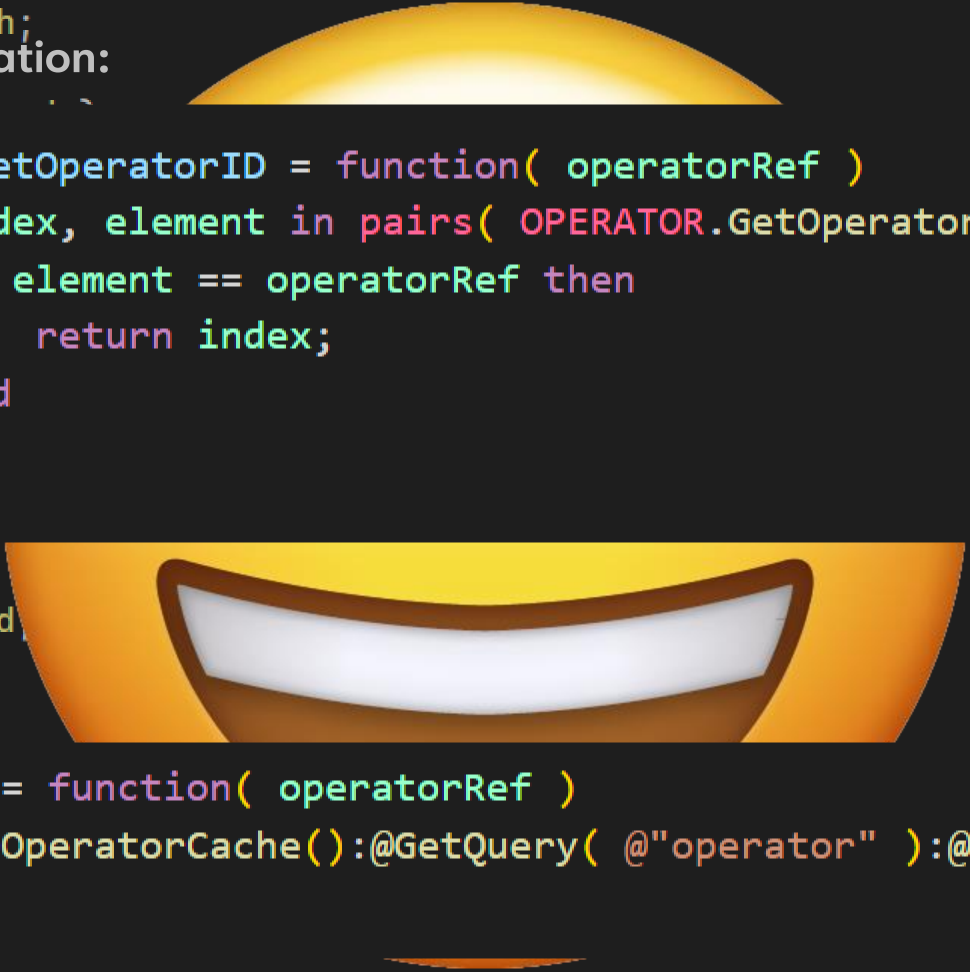
3000 stickers = 9 million iterations (63s)

EFFICIENT RUNTIME DATA QUERYING

```
uint begin = 0;
uint end = length;
whi
{
    {
        end = mid;
    }
    else

return std::nullopt;
```

Lua O(n) iteration:

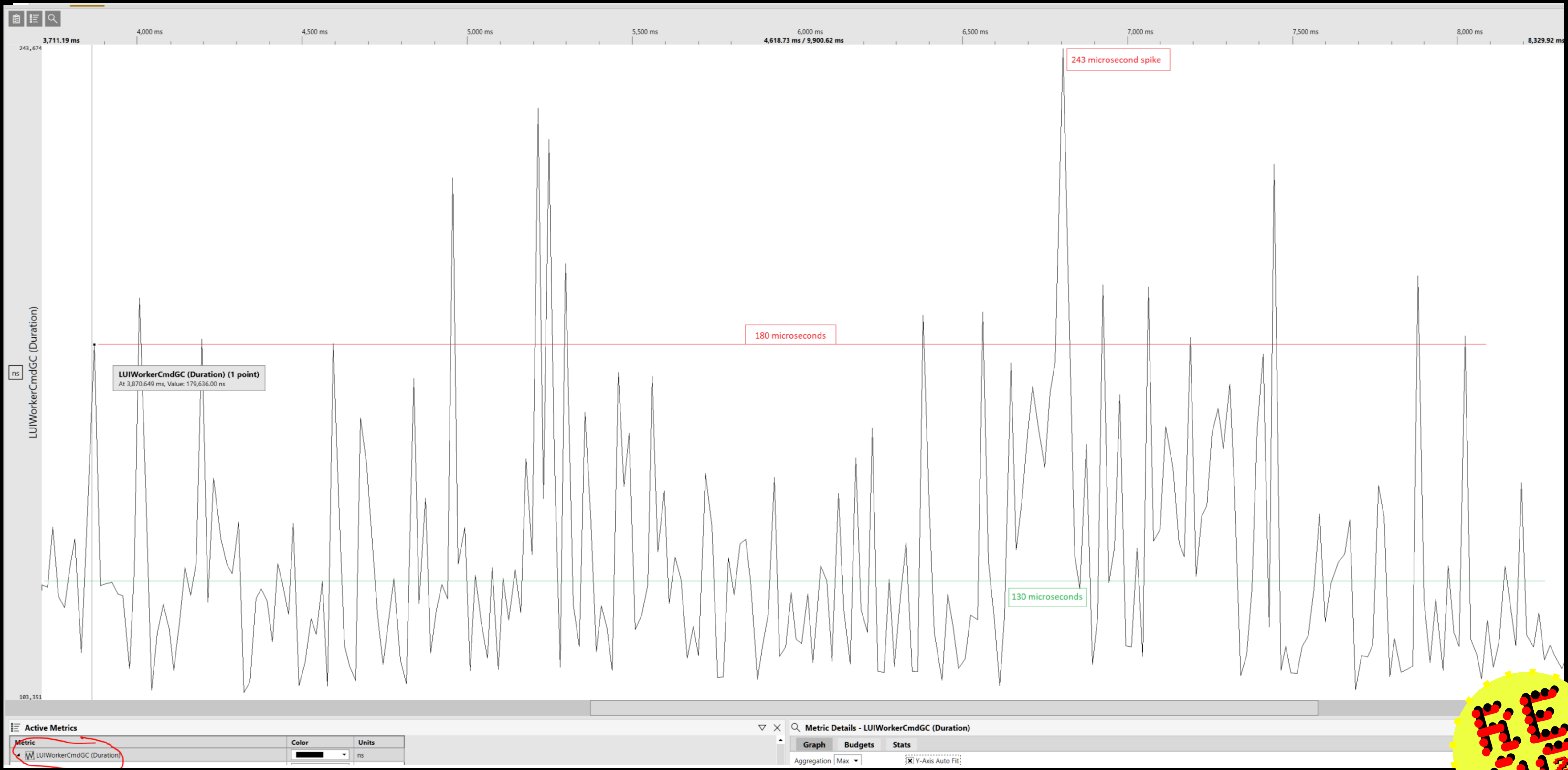
```
OPERATOR.GetOperatorID = function( operatorRef )
    for index, element in pairs( OPERATOR.GetOperatorCache() ) do
        if element == operatorRef then
            return index;
        end
    end
end
```



C++ O(log n) query:

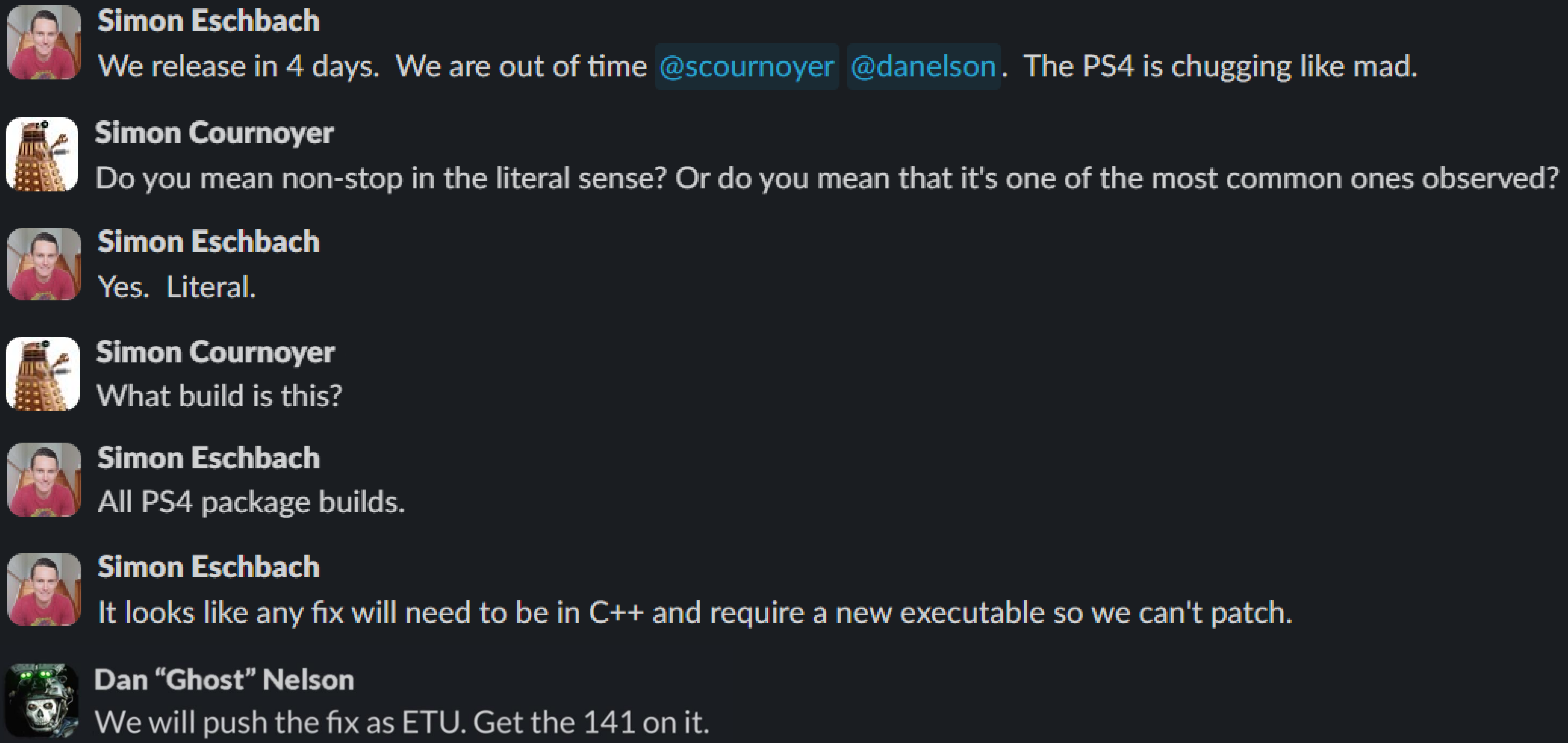
```
OPERATOR.GetOperatorID = function( operatorRef )
    return OPERATOR.GetOperatorCache():@GetQuery( @"operator" ):@FindIndex( operatorRef );
end
```

GARBAGE COLLECTION IN REAL TIME APPLICATIONS



PART 3: THE CLUTCH

4 DAYS TO LAUNCH



Simon Eschbach
We release in 4 days. We are out of time @scournoyer @danelson. The PS4 is chugging like mad.

Simon Cournoyer
Do you mean non-stop in the literal sense? Or do you mean that it's one of the most common ones observed?

Simon Eschbach
Yes. Literal.

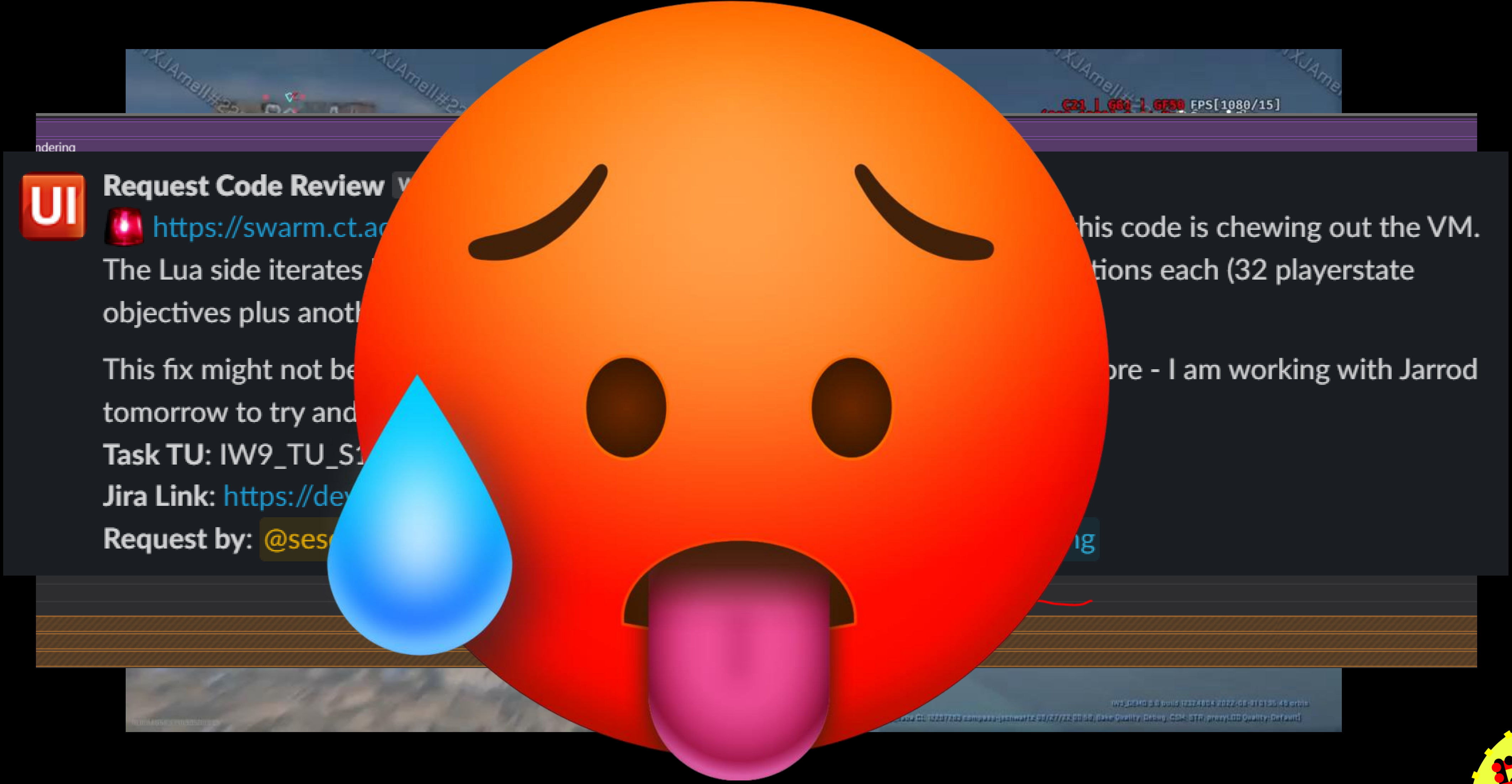
Simon Cournoyer
What build is this?

Simon Eschbach
All PS4 package builds.

Simon Eschbach
It looks like any fix will need to be in C++ and require a new executable so we can't patch.

Dan "Ghost" Nelson
We will push the fix as ETU. Get the 141 on it.

IN THE NICK OF TIME


ndering

Request Code Review

https://swarm.ct.ac his code is chewing out the VM.
The Lua side iterates tions each (32 playerstate
objectives plus anoth

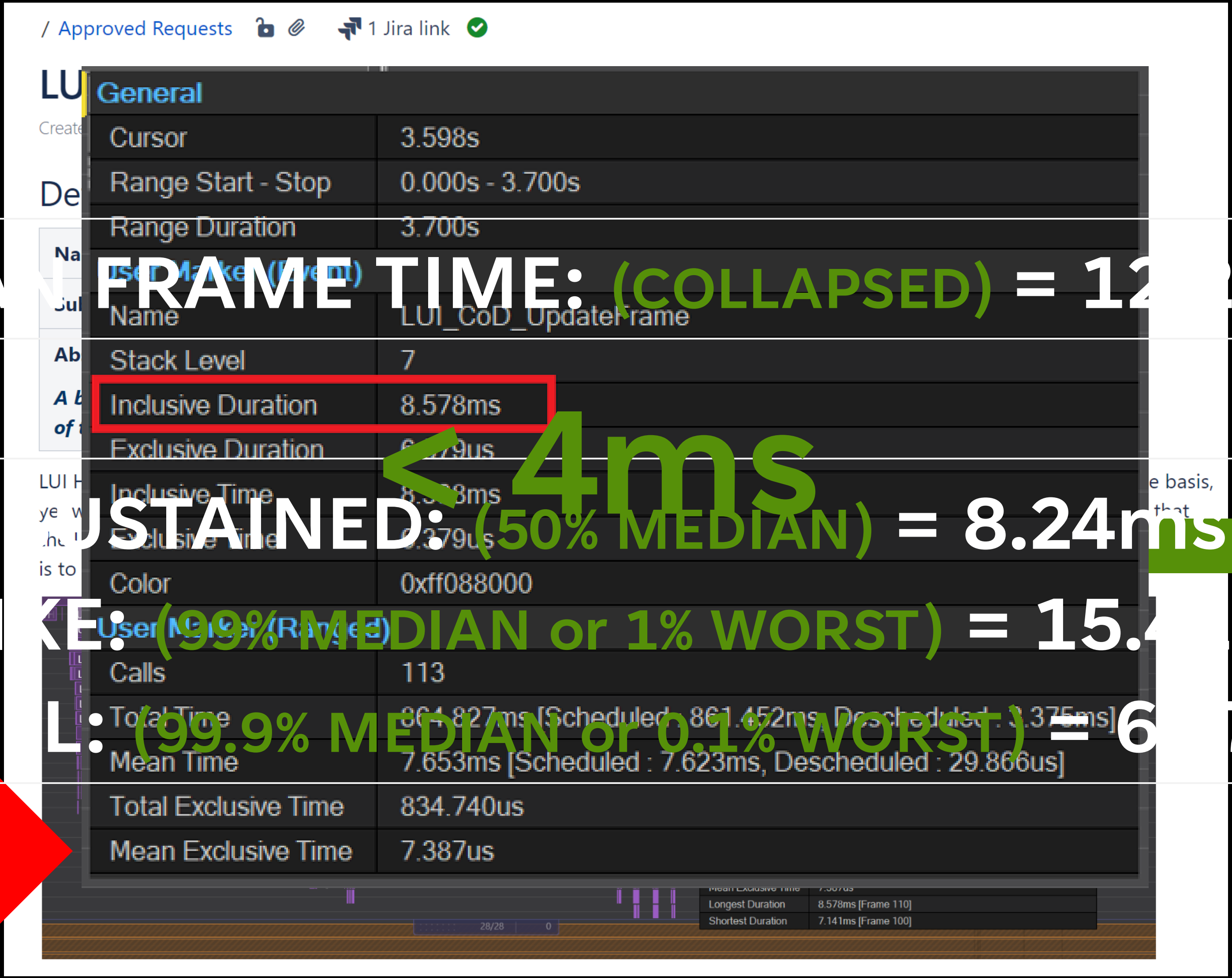
This fix might not be ore - I am working with Jarrod
tomorrow to try and
Task TU: IW9_TU_S1
Jira Link: https://de
Request by: @ses g

MISSION ACCOMPLISHED

OR WAS IT?

PART 4: BATTLE HARDENING

MEASUREMENT



/ Approved Requests 1 Jira link

General	
Cursor	3.598s
Range Start - Stop	0.000s - 3.700s
Range Duration	3.700s
User Marker (Event)	
Name	LUI_CoD_UpdateFrame
Stack Level	7
Inclusive Duration	8.578ms
Exclusive Duration	6.379us
Inclusive Time	8.578ms
Exclusive Time	6.379us
Color	0xff088000
User Marker (Ranged)	
Calls	113
Total Time	864.827ms [Scheduled : 861.452ms, Descheduled : 3.375ms]
Mean Time	7.653ms [Scheduled : 7.623ms, Descheduled : 29.866us]
Total Exclusive Time	834.740us
Mean Exclusive Time	7.387us
Longest Duration	8.578ms [Frame 110]
Shortest Duration	7.141ms [Frame 100]

MEAN FRAME TIME: (COLLAPSED) = 12[illegible]6ms

< 4ms

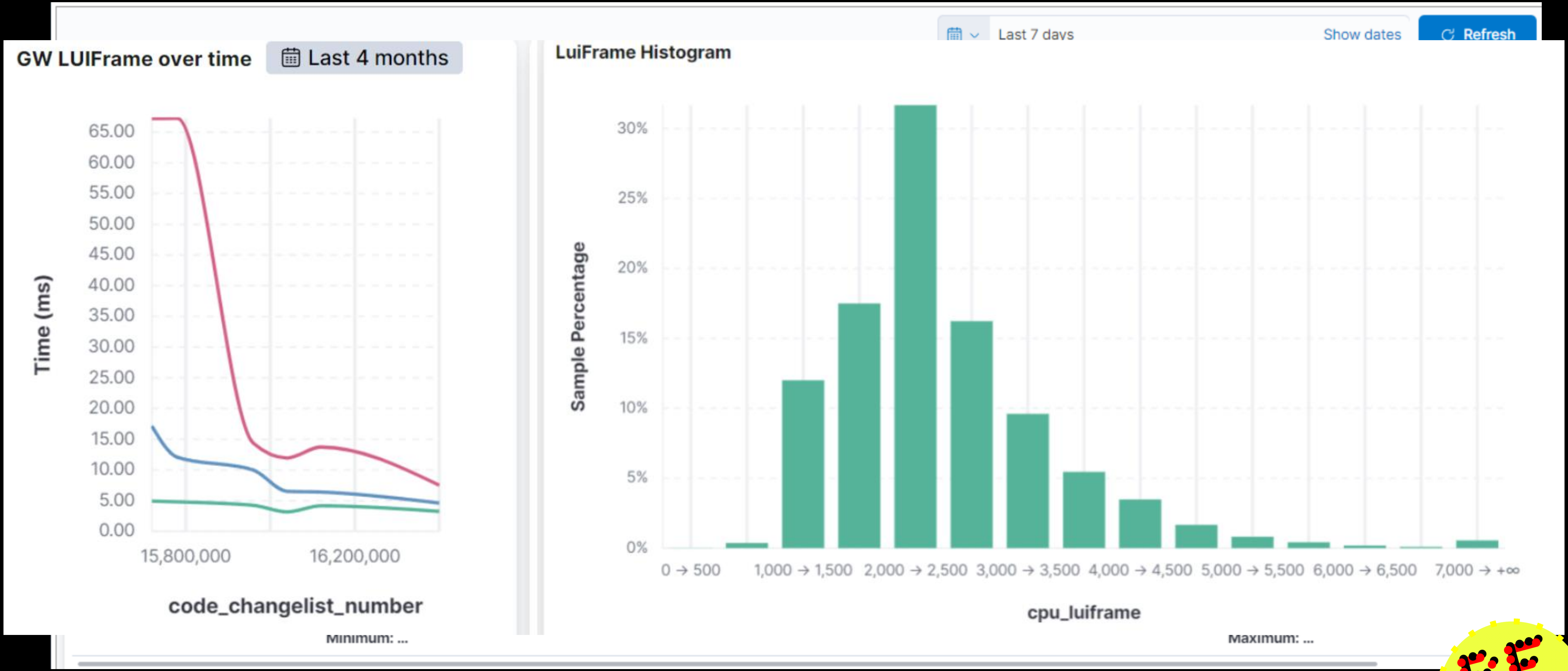
SUSTAINED: (50% MEDIAN) = 8.24ms

4ms

SPIKE: (99% MEDIAN or 1% WORST) = 15.[illegible]ms

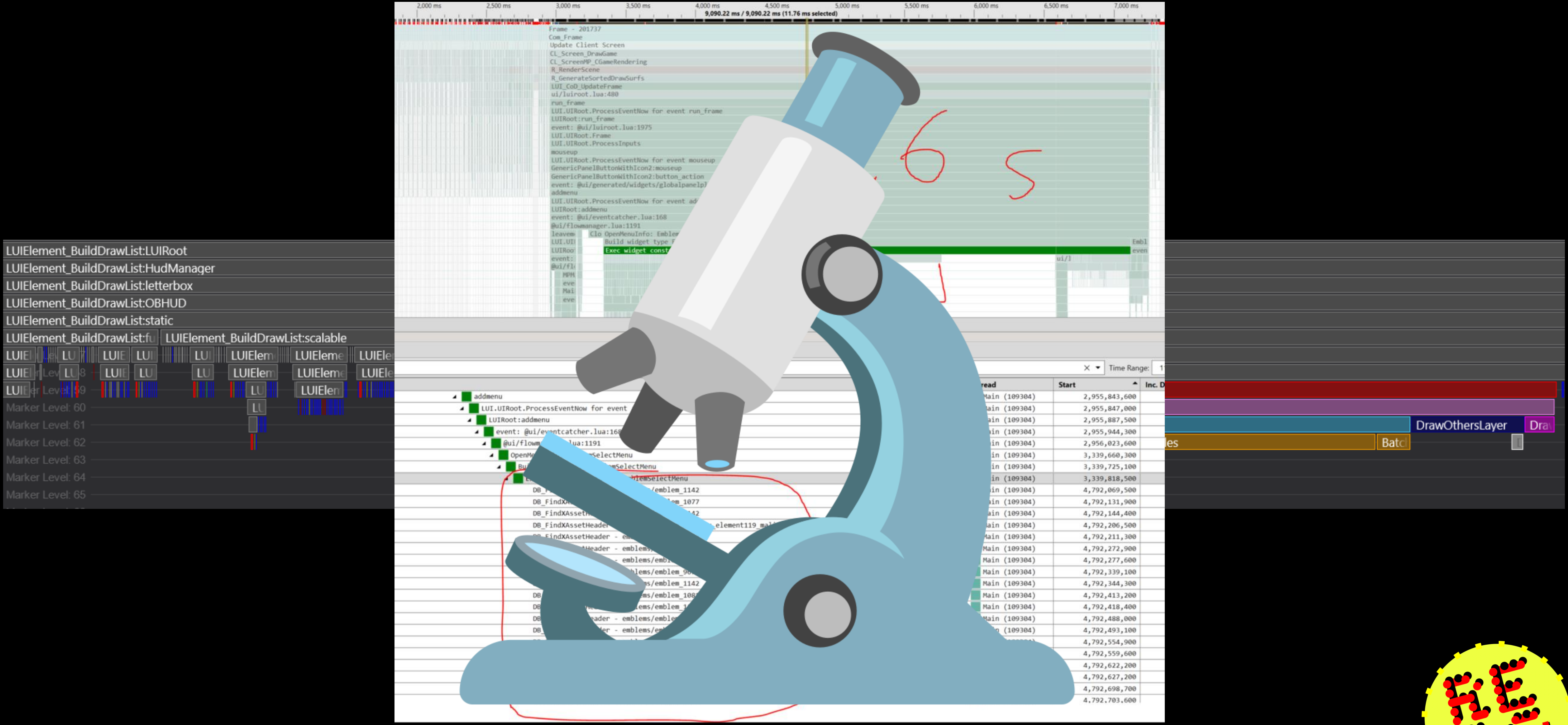
STALL: (99.9% MEDIAN or 0.1% WORST) = 6[illegible]7ms

DASHBOARDS



Last 7 days
Show dates
Refresh
GW LUIFrame over time
Last 4 months
Time (ms)
65.00
60.00
55.00
50.00
45.00
40.00
35.00
30.00
25.00
20.00
15.00
10.00
5.00
0.00
15,800,000
16,200,000
code_changelist_number
LuiFrame Histogram
Sample Percentage
30%
25%
20%
15%
10%
5%
0%
0 → 500
1,000 → 1,500
2,000 → 2,500
3,000 → 3,500
4,000 → 4,500
5,000 → 5,500
6,000 → 6,500
7,000 → +∞
cpu_luiframe
Minimum: ...
Maximum: ...

PROCESSESES AND DIAGNOSTICS

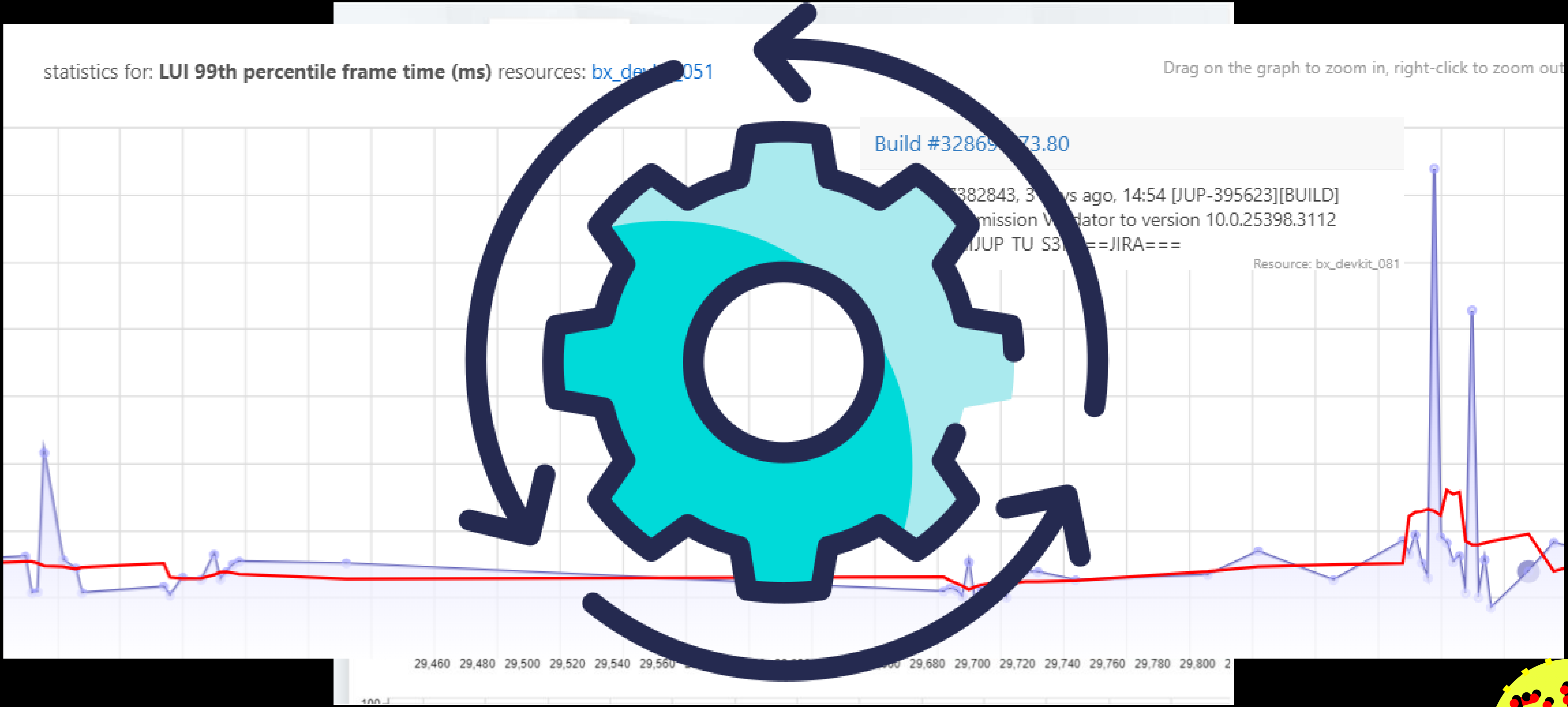


```
Frame - 201737
Com_Frame
Update Client Screen
CL_Screen_DrawGame
CL_ScreenMP_CGameRendering
R_RenderScene
R_GenerateSortedDrawSurfs
LUI_CoD_UpdateFrame
ui/luiroot.lua:480
run_frame
LUI.UIRoot.ProcessEventNow for event run_frame
LUIRoot:run_frame
event: @ui/luiroot.lua:1975
LUI.UIRoot.Frame
LUI.UIRoot.ProcessInputs
mouseup
LUI.UIRoot.ProcessEventNow for event mouseup
GenericPanelButtonWithIcon2:mouseup
GenericPanelButtonWithIcon2:button_action
event: @ui/generated/widgets/globalpanelpl
addmenu
LUI.UIRoot.ProcessEventNow for event ad
LUIRoot:addmenu
event: @ui/eventcatcher.lua:168
@ui/flowmanager.lua:1191
leavem   Clo OpenMenuInfo: Emblem
LUI.UI       Build widget type F
LUIRoo       Exec widget const
```

```
LUIElement_BuildDrawList:LUIRoot
LUIElement_BuildDrawList:HudManager
LUIElement_BuildDrawList:letterbox
LUIElement_BuildDrawList:OBHUD
LUIElement_BuildDrawList:static
LUIElement_BuildDrawList:fu  LUIElement_BuildDrawList:scalable
```

```
                                                            read            Start          Inc. D
addmenu                                                     Main (109304)   2,955,843,600
  LUI.UIRoot.ProcessEventNow for event                      Main (109304)   2,955,847,000
    LUIRoot:addmenu                                         Main (109304)   2,955,887,500
      event: @ui/eventcatcher.lua:168                       Main (109304)   2,955,944,300
        @ui/flowm      lua:1191                             Main (109304)   2,956,023,600
          OpenMe        SelectMenu                          Main (109304)   3,339,660,300
            Bu          mSelectMenu                         Main (109304)   3,339,725,100
              E          blemSelectMenu                     Main (109304)   3,339,818,500
                DB_F           /emblem_1142                 Main (109304)   4,792,069,500
                DB_FindX       em_1077                      Main (109304)   4,792,131,900
                DB_FindXAsset  142                          Main (109304)   4,792,144,400
                DB_FindXAssetHeader    element119_mal       Main (109304)   4,792,206,500
                DB_FindXAssetHeader - em                    Main (109304)   4,792,211,300
                   tHeader - emblems                        Main (109304)   4,792,272,900
                      - emblems/emb                         Main (109304)   4,792,277,600
                        blems/emblem_90                     Main (109304)   4,792,339,100
                        s/emblem_1142                       Main (109304)   4,792,344,300
                DB      ems/emblem_108                      Main (109304)   4,792,413,200
                DB      ems/emblem_1                        Main (109304)   4,792,418,400
                DB   eader - emblems/emble                  Main (109304)   4,792,488,000
                DB    er - emblems/em                       Main (109304)   4,792,493,100
                                                                            4,792,554,900
                                                                            4,792,559,600
                                                                            4,792,622,200
                                                                            4,792,627,200
                                                                            4,792,698,700
                                                                            4,792,703,600
```

AUTOMATED PERFORMANCE TESTING


statistics for: **LUI 99th percentile frame time (ms)** resources: bx_dev 051

Drag on the graph to zoom in, right-click to zoom out

Build #32869 73.80

382843, 3 s ago, 14:54 [JUP-395623][BUILD]
mission V dator to version 10.0.25398.3112
JUP TU S3 ==JIRA===

Resource: bx_devkit_081

RESULTS – PS4 BASE PLATFORM

SINGLEPLAYER:	1.2ms
MULTIPLAYER:	2.1ms
GROUND WAR:	3.1ms
WARZONE:	3.9ms

THANK-YOU!

Sound: Mike Tornabene

Voice Over: Dan Nelson

Photoshop: Carl Prescott & Kyle Turchik