USER INTERFACE IN CYBERPUNK 2077 CHALLENGES AND OPTIMIZATIONS

Arkadiusz Antonik | Engineering Director CD Projekt RED

USER ID_98877 LENS DRVR 3.8.03
STATUS: ONLINE SERIAL>34 73.32

ARKADIUSZ ANTONIK

- 10 years in the game industry
- Engine, Tools, Gameplay and User Interface programmer
- UI Code Lead, Cyberpunk 2077
- Engineering Director, Polaris (next AAA Witcher game)

AGENDA

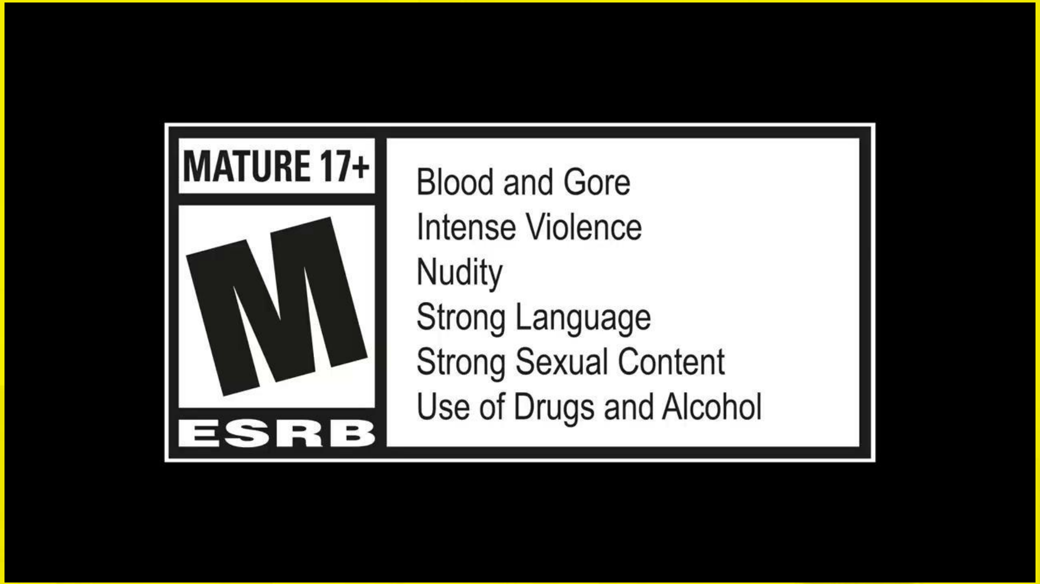
1. Introduction
2. UI framework evolution
3. Challenges
4. Optimizations
5. Summary

USER ID_99877 LENS DRVR 3.0.03
STATUS: ONLINE SERIAL>34 73.32
KIROSHI

NEURAL CONNECTION STATUS: ACTIVE

8234299 -AXRCG 0001

LET'S PLAY A GAME



MATURE 17+

ESRB

Blood and Gore
Intense Violence
Nudity
Strong Language
Strong Sexual Content
Use of Drugs and Alcohol

UI FRAMEWORK DEVELOPMENT TIMELINE

MAY 2016

WITCHER 3
BLOOD&WINE

Release date

WHY CUSTOM USER INTERFACE FRAMEWORK?

EXPECTATIONS TOWARDS THE UI FRAMEWORK

Technical side

- In-engine creation
- Shorter iteration time
- Using internal engine subsystems
- Higher density and complexity

Creative side

- More dynamic elements
- Support for multiple languages
- Possibility to display & interact in 3D
- Contains embedded videos
- Possibility to use custom materials & effects

RESEARCH AND LACK OF EXISTING SOLUTIONS

Evaluation of existing solutions

- Scaleform was dropped and not supported
- No single complete solution
- Partial solutions/middleware:
  - not efficient
  - not scalable enough
  - challenging integration with RedEngine

UI FRAMEWORK DEVELOPMENT TIMELINE

MAY 2016

WITCHER 3
BLOOD&WINE

Release date

APRIL 2017

Decision about creating custom UI framework

Ink framework was born

UI FRAMEWORK DEVELOPMENT – STEP 1/3

- In-engine MVP (a few weeks)
  - simple widget types
  - 2d input propagation
  - layout building
  - integration with RedEngine systems
    - renderer
    - input system
    - file system

UI FRAMEWORK DEVELOPMENT TIMELINE

MAY 2016

WITCHER 3 BLOOD&WINE

Release date

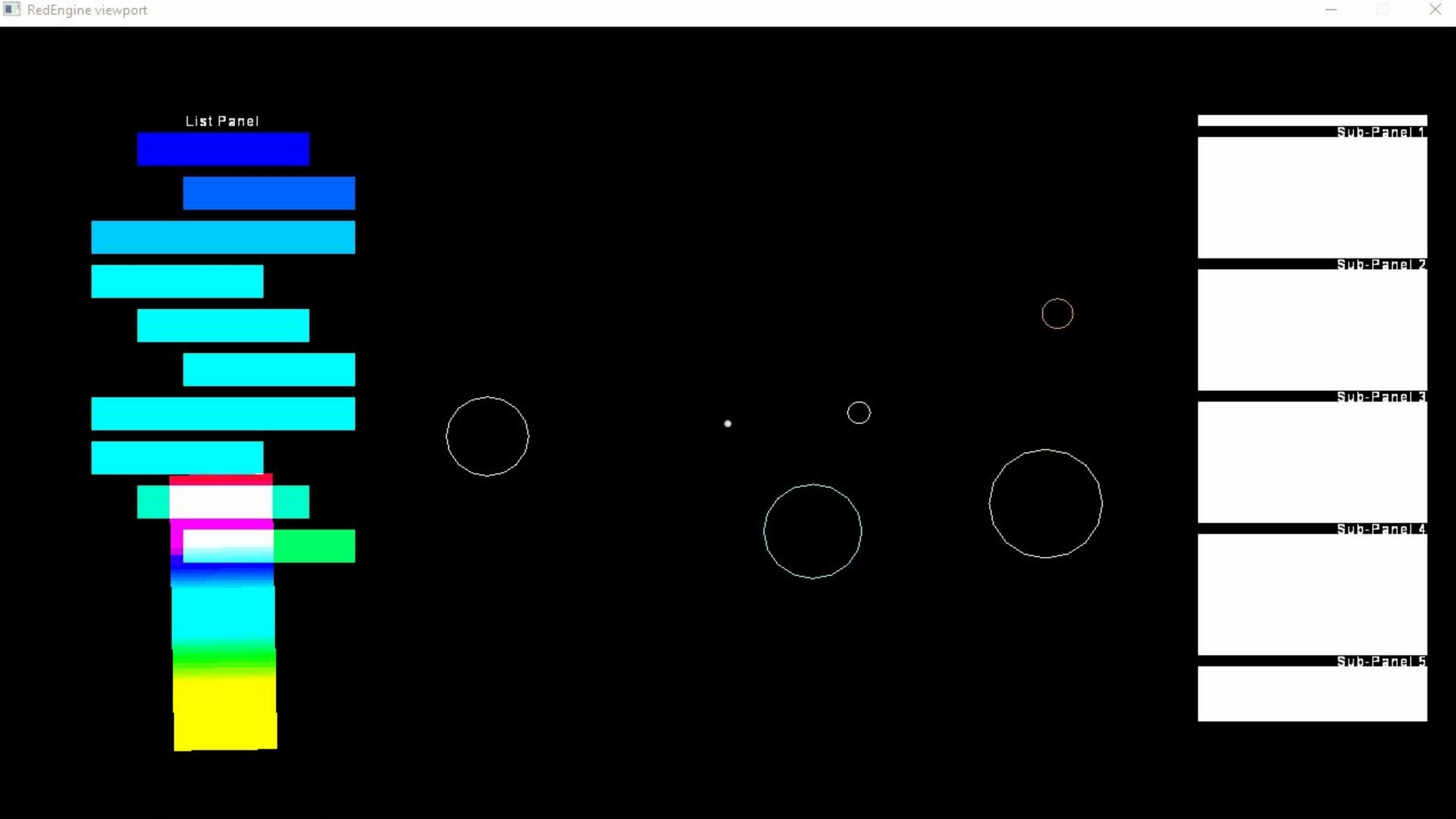
APRIL 2017

Decision about creating custom UI framework

Ink framework was born

JUNE 2017

MVP of UI framework


RedEngine viewport
List Panel
Sub-Panel 1
Sub-Panel 2
Sub-Panel 3
Sub-Panel 4
Sub-Panel 5

UI FRAMEWORK DEVELOPMENT – STEP 2/3

- In-game UI (around half a year)
  - fully functional UI elements
  - simple 3d implementation
  - no editors
    - all layouts hardcoded in c++
  - support for embedded videos

UI FRAMEWORK DEVELOPMENT TIMELINE

MAY 2016

WITCHER 3 BLOOD&WINE

Release date

APRIL 2017

Decision about creating custom UI framework

Ink framework was born

JUNE 2017

MVP of UI framework

DECEMBER 2017

The first in-game UI

Internal demo

QUEST ACCEPTED

A FAVOR FOR A FRIEND

Fix Cobb's problem:

Talk to Cobb:

UI FRAMEWORK DEVELOPMENT – STEP 3/3

- Proper implementation
  - proper connection engine <- UI -> gameplay
- Proper pipelines
  - editors and asset importers
- Proper UI in game
  - not hardcoded in c++
- and more ...

SOME OFFLINE STATISTICS

	Number of files	Weight of files
Embedded videos	584	10.4 GB
UI textures	1933	4.3 GB
Widget libraries	1584	116.8 MB
Widget atlases	1079	37.3 MB
Fonts	49	31.4 MB
Widget animations	714	21.4 MB
Widget styles	253	4.7 MB
HUD resources	16	0.9 MB
Widget menus	7	0.5 MB

USER INTERFACE BUDGETS IN CYBERPUNK 2077

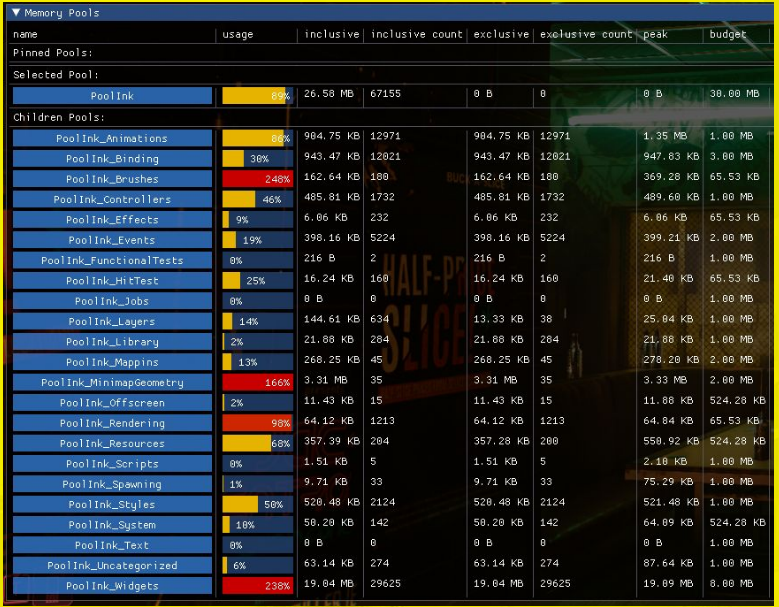
UI MEMORY BUDGET

Budget

- 40 MB on CPU
  - finally 55 MB (because of fonts)
- 250 MB on GPU

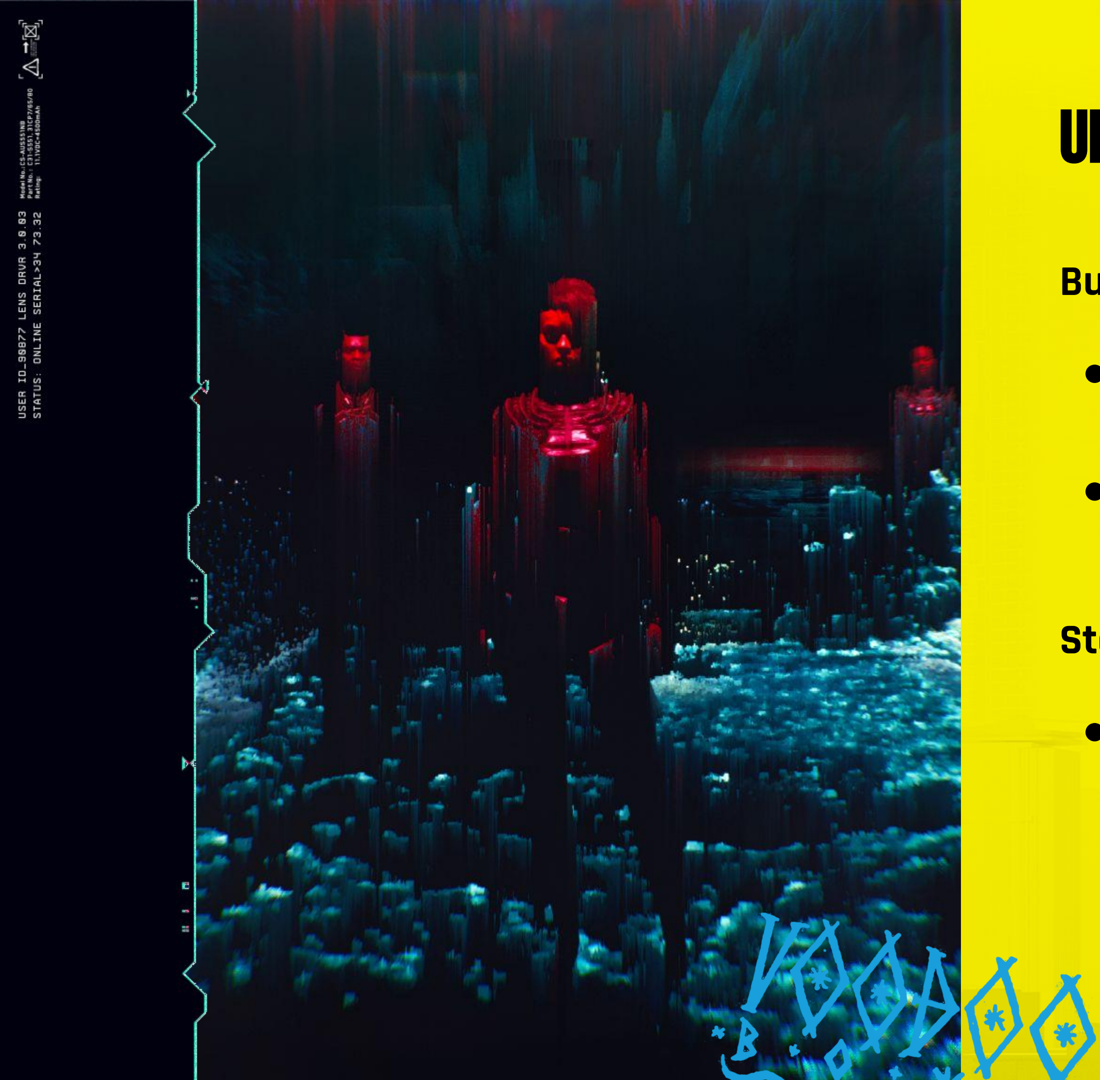
Starting point

- CPU memory: ~150 MB
- GPU memory: ~700 MB



▼ Memory Pools

name	usage	inclusive	inclusive count	exclusive	exclusive count	peak	budget
Pinned Pools:							
Selected Pool:							
PoolInk	89%	26.58 MB	67155	0 B	0	0 B	30.00 MB
Children Pools:							
PoolInk_Animations	86%	904.75 KB	12971	904.75 KB	12971	1.35 MB	1.00 MB
PoolInk_Binding	30%	943.47 KB	12021	943.47 KB	12021	947.83 KB	3.00 MB
PoolInk_Brushes	248%	162.64 KB	180	162.64 KB	180	369.28 KB	65.53 KB
PoolInk_Controllers	46%	485.81 KB	1732	485.81 KB	1732	489.60 KB	1.00 MB
PoolInk_Effects	9%	6.06 KB	232	6.06 KB	232	6.06 KB	65.53 KB
PoolInk_Events	19%	398.16 KB	5224	398.16 KB	5224	399.21 KB	2.00 MB
PoolInk_FunctionalTests	0%	216 B	2	216 B	2	216 B	1.00 MB
PoolInk_HitTest	25%	16.24 KB	160	16.24 KB	160	21.40 KB	65.53 KB
PoolInk_Jobs	0%	0 B	0	0 B	0	0 B	1.00 MB
PoolInk_Layers	14%	144.61 KB	634	13.33 KB	38	25.04 KB	1.00 MB
PoolInk_Library	2%	21.88 KB	284	21.88 KB	284	21.88 KB	1.00 MB
PoolInk_Mappins	13%	268.25 KB	45	268.25 KB	45	278.20 KB	2.00 MB
PoolInk_MinimapGeometry	166%	3.31 MB	35	3.31 MB	35	3.33 MB	2.00 MB
PoolInk_Offscreen	2%	11.43 KB	15	11.43 KB	15	11.88 KB	524.28 KB
PoolInk_Rendering	98%	64.12 KB	1213	64.12 KB	1213	64.84 KB	65.53 KB
PoolInk_Resources	68%	357.39 KB	204	357.28 KB	200	550.92 KB	524.28 KB
PoolInk_Scripts	0%	1.51 KB	5	1.51 KB	5	2.10 KB	1.00 MB
PoolInk_Spawning	1%	9.71 KB	33	9.71 KB	33	75.29 KB	1.00 MB
PoolInk_Styles	50%	520.48 KB	2124	520.48 KB	2124	521.48 KB	1.00 MB
PoolInk_System	10%	50.20 KB	142	50.20 KB	142	64.09 KB	524.28 KB
PoolInk_Text	0%	0 B	0	0 B	0	0 B	1.00 MB
PoolInk_Uncategorized	6%	63.14 KB	274	63.14 KB	274	87.64 KB	1.00 MB
PoolInk_Widgets	238%	19.04 MB	29625	19.04 MB	29625	19.09 MB	8.00 MB



UI TIME BUDGET

Budget

- 3-5ms on the heaviest synchronization thread
- Multithreaded execution

Starting point

- 10-15 ms on a single thread

UI FRAMEWORK DEVELOPMENT TIMELINE

APRIL 2017

Decision about creating custom UI framework

Ink framework was born

JUNE 2017

MVP of UI framework

DECEMBER 2017

The first in-game UI

Internal demo

AUGUST 2018

Public Gameplay Reveal

Gamescom demo

SEPTEMBER 2018

Budgets are introduced into the game.

Big optimization has been started!

USER ID_98877 LENS DRVR 3.0.03
STATUS: ONLINE SERIAL>34 73.32

WELCOME BUDGETS!

WELCOME CHALLENGES!

WHY UI IS THAT HEAVY?

ANSWER 1:

UI framework is not optimized [code]

ANSWER 2:

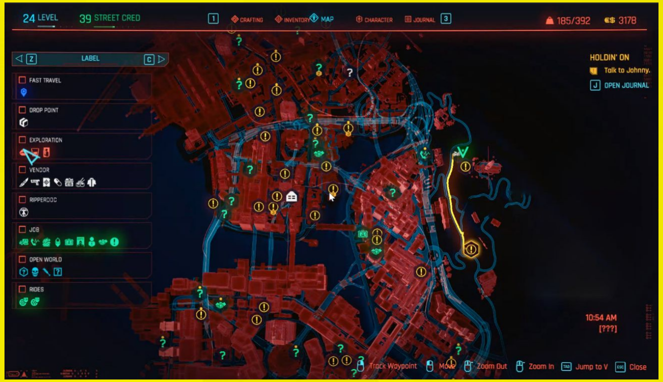
Too many UI instances [content]

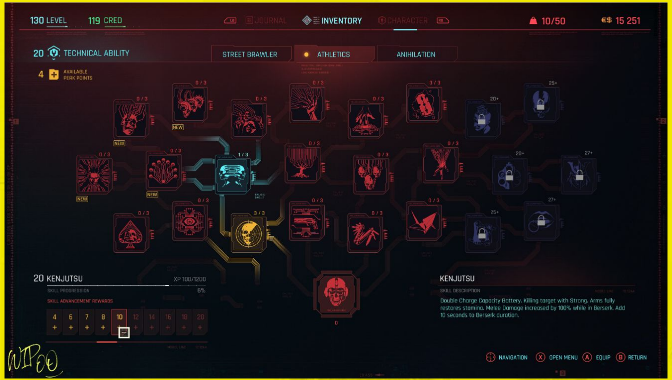
SOLUTION

Optimize UI framework to keep in memory, update and draw ONLY UI instances that are really necessary at the specific moment in the game.

WHERE IS THAT COMPLEXITY?

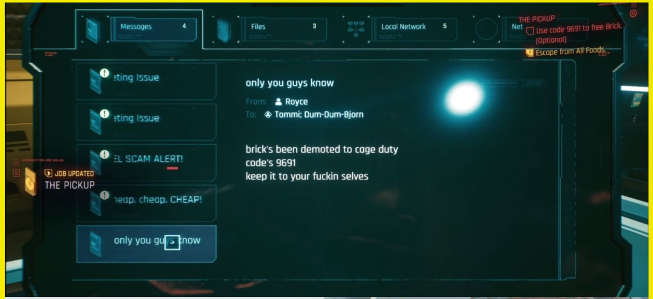
- On-screen UI
  - Minimap
  - Mappins
  - 3D Map
  - Inventory
  - Perks
  - HUD elements





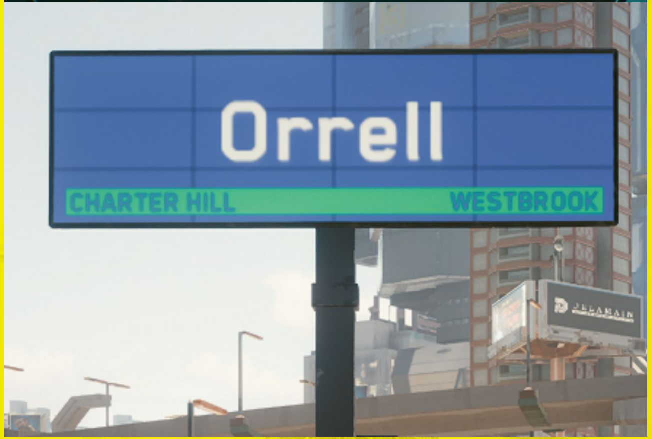
WHERE IS THAT DENSITY?

- In-world UI
    - UI on weapons
    - UI in cars
    - Device system
    - Billboards
    - Localized street signs
    - Global TV system
    - Localized texts in random places



only you guys know
From: Royce
To: Tommi; Dum-Dum-Bjorn

brick's been demoted to cage duty
code's 9691
keep it to your fuckin selves



SOME RUNTIME STATISTICS

Active animations	~330
Passive animations	~60
Spawned widgets	~16000
Game controllers	~160
Logic controllers	~1650
Advertisements	~330
Independent layers	~12
HUD entries	~65
3D UI spawned	~150
3D UI in view	~70
Languages	Up to 4 simultaneously

UI TERMS DEFINITION - PART 1/3 : HIERARCHY

UI Instance

Independent UI hierarchy, instantiated in memory with it's all depended assets and objects, like: textures, animations, render targets. Spawned from Widget Library Item.

Widget Library Item

Single UI hierarchy (template) that can be spawned (instantiated) in runtime and became a UI instance. Exists in Widget Library asset.

Widget Library asset

Asset/file that contains a list (a library) of Widget Library items and references to depended assets.

OPTIMIZATIONS

S/N::: 1 / 052423
TASK STATUS: ACTIVE
TASK NAME:
GROUP UI INSTANCES

S/N::: 2 / 052423
TASK STATUS: TO-DO
TASK NAME:
REDUCE UI UPDATES

S/N::: 3 / 052423
TASK STATUS: TO-DO
TASK NAME:
REDUCE UI ANIMATION UPDATES

S/N::: 4 / 052423
TASK STATUS: TO-DO
TASK NAME:
UI INSTANCE CULLING

S/N::: 5 / 052423
TASK STATUS: TO-DO
TASK NAME:
DEFERRED UPDATE & DRAW

S/N::: 6 / 052423
TASK STATUS: TO-DO
TASK NAME:
HLOD FOR UI IN 3D-WORLD

LAYER CONCEPT (1/3): DEFINITION

UI layer

Independent logical object containing many UI hierarchies (dependent or independent from each other).

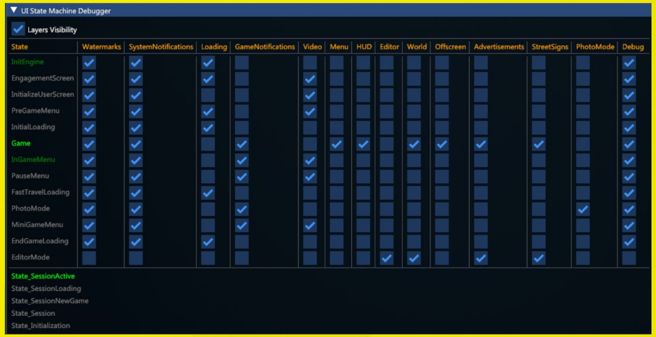
A single layer contains

- four main objects:
  - event broker
  - spawning processor
  - controller processor
  - animation processor
- separate resource management
- custom separate drawing logic

LAYER CONCEPT (2/3): TYPES

Layers type

- Fullscreen (*Watermarks, System Notifications, Loading, Game Notifications, Menu, Video, HUD, Photo Mode, Editor*)
- In-world (*World, Advertisements, Street Signs*)
- Misc (*Offscreen, Debug*)

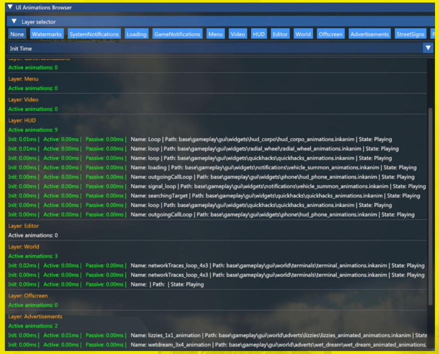


UI State Machine Debugger

✓ Layers Visibility

State	Watermarks	SystemNotifications	Loading	GameNotifications	Video	Menu	HUD	Editor	World	Offscreen	Advertisements	StreetSigns	PhotoMode	Debug
InitEngine	✓	✓	✓											✓
EngagementScreen	✓	✓	✓		✓									✓
InitializeUserScreen	✓	✓			✓									✓
PreGameMenu	✓	✓	✓		✓									✓
InitialLoading	✓	✓	✓											✓
Game	✓	✓		✓		✓	✓		✓	✓	✓	✓		✓
InGameMenu	✓	✓		✓	✓									✓
PauseMenu	✓	✓		✓	✓									✓
FastTravelLoading	✓	✓	✓											✓
PhotoMode	✓	✓		✓									✓	✓
MiniGameMenu	✓	✓		✓	✓									✓
EndGameLoading	✓	✓	✓											✓
EditorMode								✓	✓		✓	✓		

State_SessionActive
State_SessionLoading
State_SessionNewGame
State_Session
State_Initialization



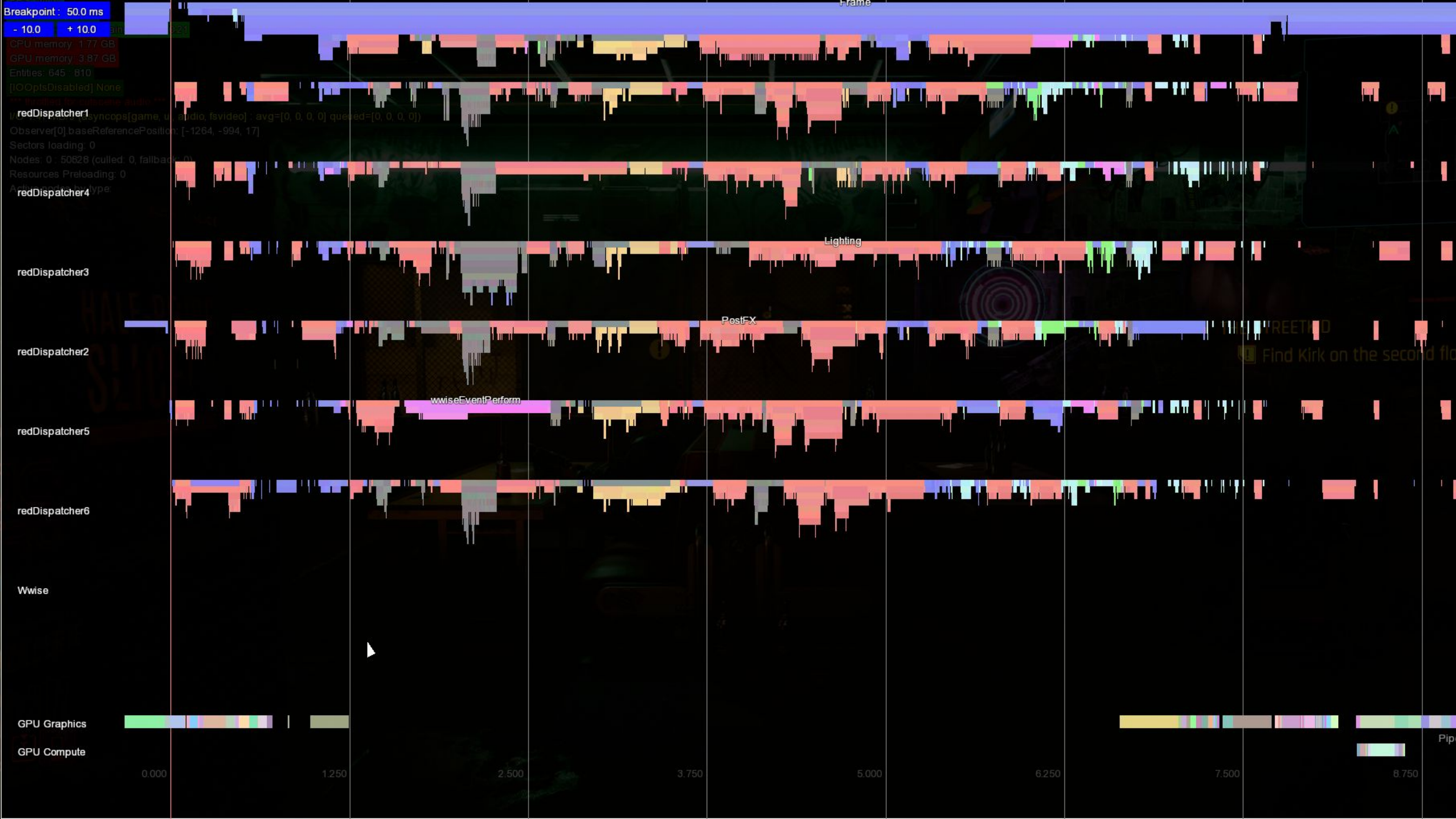
UI Animations Browser

Layer selector

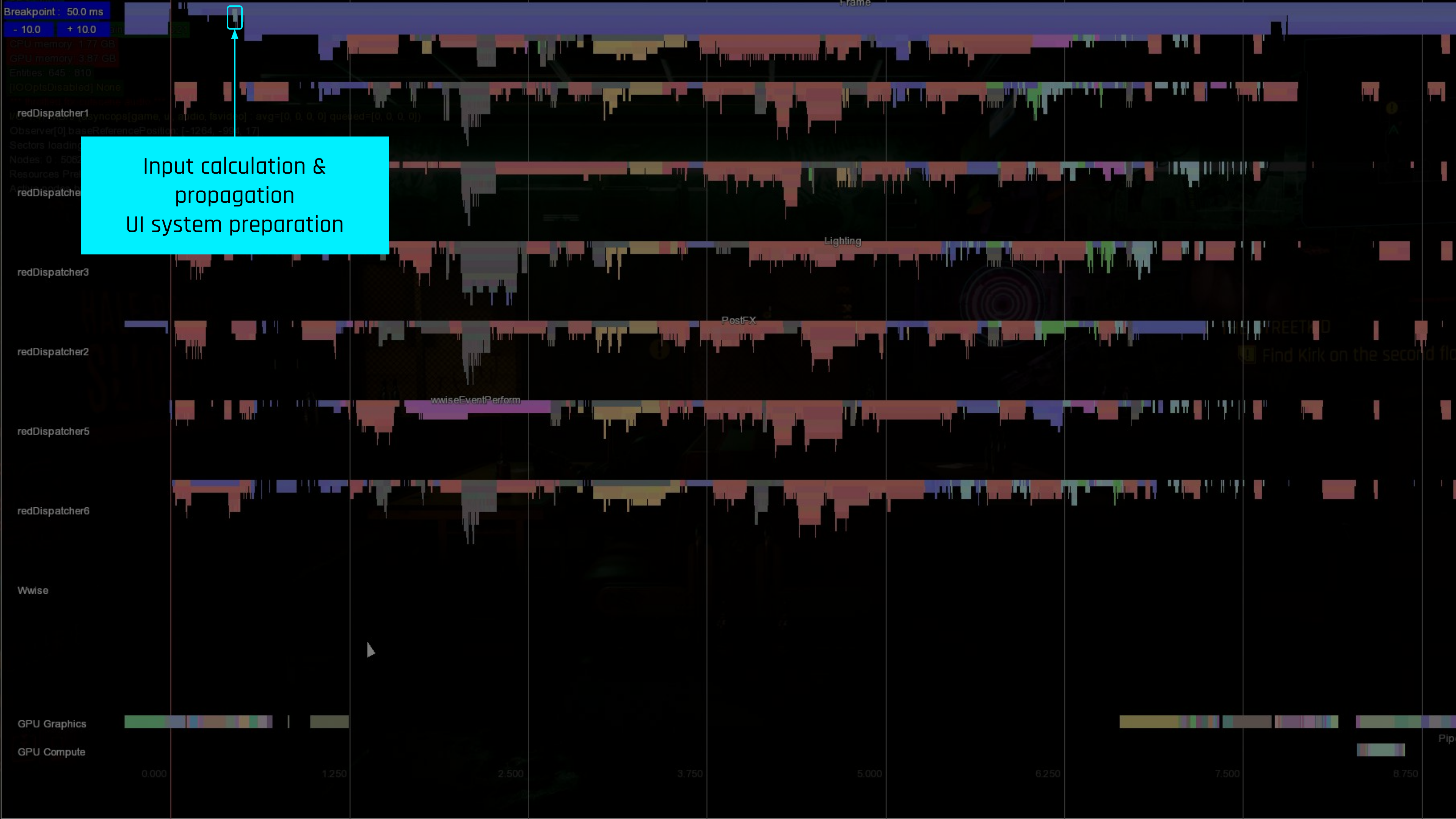
```
Active animations: 0
Layer: Menu
Active animations: 0
Layer: Video
Active animations: 0
Layer: HUD
Active animations: 9
Init: 0.01ms |  Active: 0.00ms |  Passive: 0.00ms |  Name: Loop | Path: base\gameplay\gui\widgets\hud_corpo\hud_corpo_animations.inkanim | State: Playing
Init: 0.01ms |  Active: 0.00ms |  Passive: 0.00ms |  Name: loop | Path: base\gameplay\gui\widgets\radial_wheel\radial_wheel_animations.inkanim | State: Playing
Init: 0.00ms |  Active: 0.00ms |  Passive: 0.00ms |  Name: loop | Path: base\gameplay\gui\widgets\quickhacks\quickhacks_animations.inkanim | State: Playing
Init: 0.00ms |  Active: 0.00ms |  Passive: 0.00ms |  Name: loading | Path: base\gameplay\gui\widgets\notifications\vehicle_summon_animations.inkanim | State: Playing
Init: 0.00ms |  Active: 0.00ms |  Passive: 0.00ms |  Name: outgoingCallLoop | Path: base\gameplay\gui\widgets\phone\hud_phone_animations.inkanim | State: Playing
Init: 0.00ms |  Active: 0.00ms |  Passive: 0.00ms |  Name: signal_loop | Path: base\gameplay\gui\widgets\notifications\vehicle_summon_animations.inkanim | State: Playing
Init: 0.00ms |  Active: 0.00ms |  Passive: 0.00ms |  Name: searchingTarget | Path: base\gameplay\gui\widgets\quickhacks\quickhacks_animations.inkanim | State: Playing
Init: 0.00ms |  Active: 0.00ms |  Passive: 0.00ms |  Name: loop | Path: base\gameplay\gui\widgets\quickhacks\quickhacks_animations.inkanim | State: Playing
Init: 0.00ms |  Active: 0.00ms |  Passive: 0.00ms |  Name: outgoingCallLoop | Path: base\gameplay\gui\widgets\phone\hud_phone_animations.inkanim | State: Playing
Layer: Editor
Active animations: 0
Layer: World
Active animations: 3
Init: 0.02ms |  Active: 0.00ms |  Passive: 0.00ms |  Name: networkTraces_loop_4x3 | Path: base\gameplay\gui\world\terminals\terminal_animations.inkanim | State: Playing
Init: 0.00ms |  Active: 0.00ms |  Passive: 0.00ms |  Name: networkTraces_loop_4x3 | Path: base\gameplay\gui\world\terminals\terminal_animations.inkanim | State: Playing
Init: 0.00ms |  Active: 0.00ms |  Passive: 0.00ms |  Name:  | Path:  | State: Playing
Layer: Offscreen
Active animations: 0
Layer: Advertisements
Active animations: 2
Init: 0.00ms |  Active: 0.01ms |  Passive: 0.00ms |  Name: lizzies_1x1_animation | Path: base\gameplay\gui\world\adverts\lizzies\lizzies_animated_animations.inkanim | State
Init: 0.00ms |  Active: 0.00ms |  Passive: 0.00ms |  Name: wetdream_3x4_animation | Path: base\gameplay\gui\world\adverts\wet_dream\wet_dream_animated_animations.
```

LAYER CONCEPT (3/3): ASSUMPTIONS

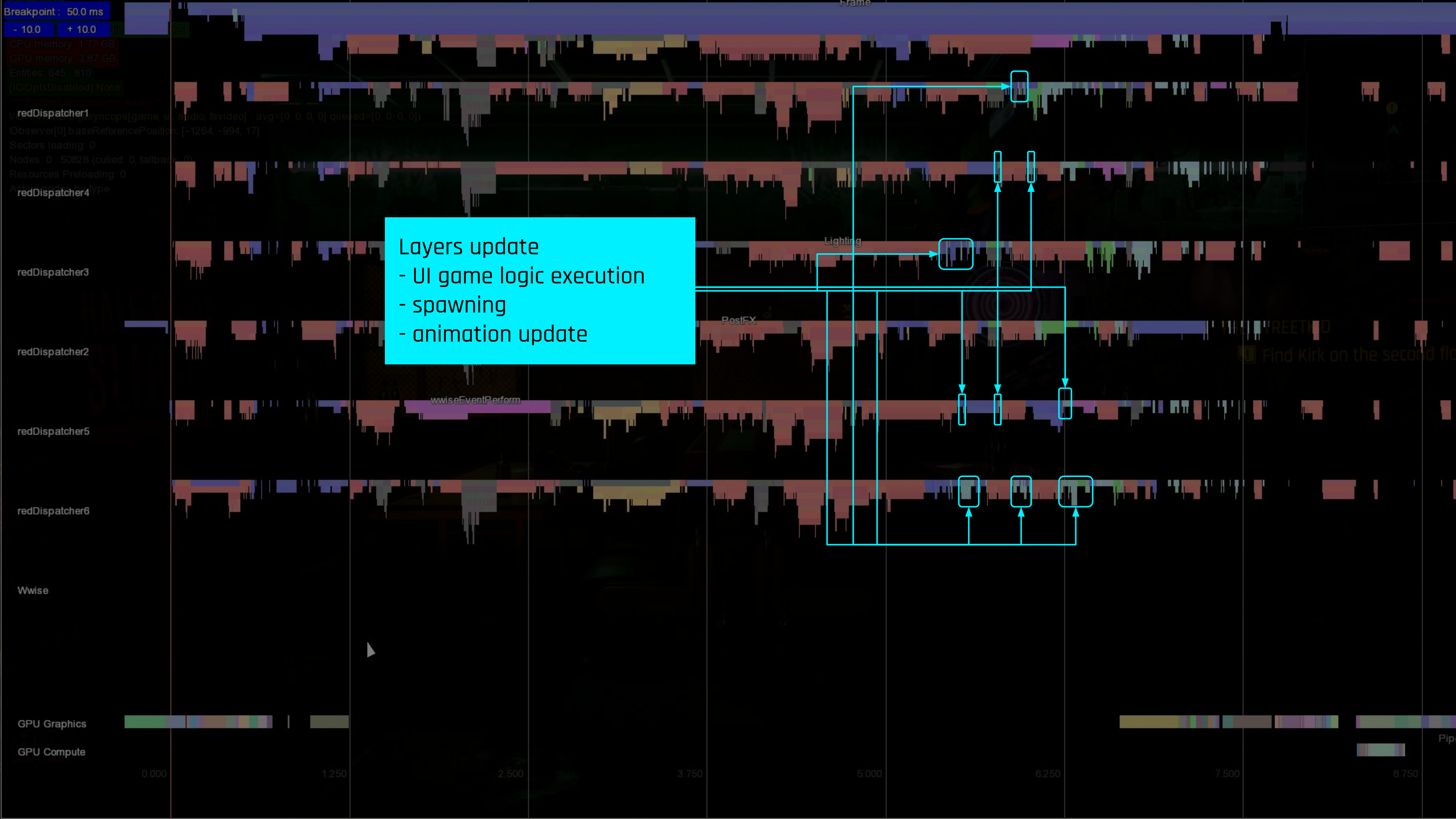
- Asynchronous layer update
    - Asynchronous game controller update
    - Asynchronous spawn request execution
        - Synchronized attaching process
    - Asynchronous animation update
        - Synchronized applying values process
- Asynchronous layer draw
    - Synchronized final composition process
- Independent job chain per layer



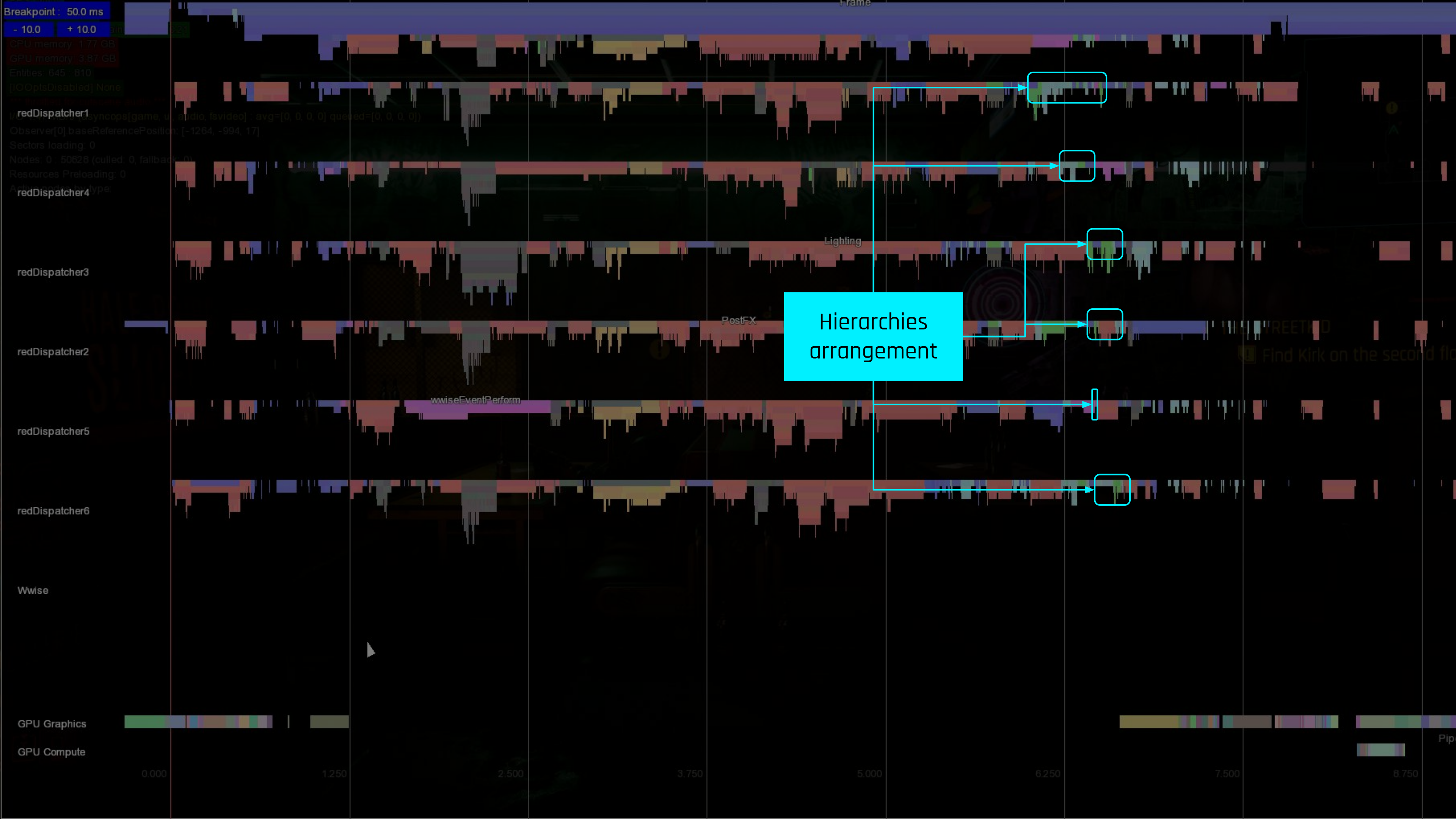
CPU memory 1.77 GB
GPU memory 3.87 GB
Entities: 645 : 810
[IOOptsDisabled] None
*** throttled for cutscene audio ***
I/O: [illegible] (asyncops[game, u[illegible], audio, fsvideo] : avg=[0, 0, 0, 0] queued=[0, 0, 0, 0])
Observer[0].baseReferencePosition: [-1264, -994, 17]
Sectors loading: 0
Nodes: 0 : 50828 (culled: 0, fallback: 0)
Resources Preloading: 0
Active nodes by type:

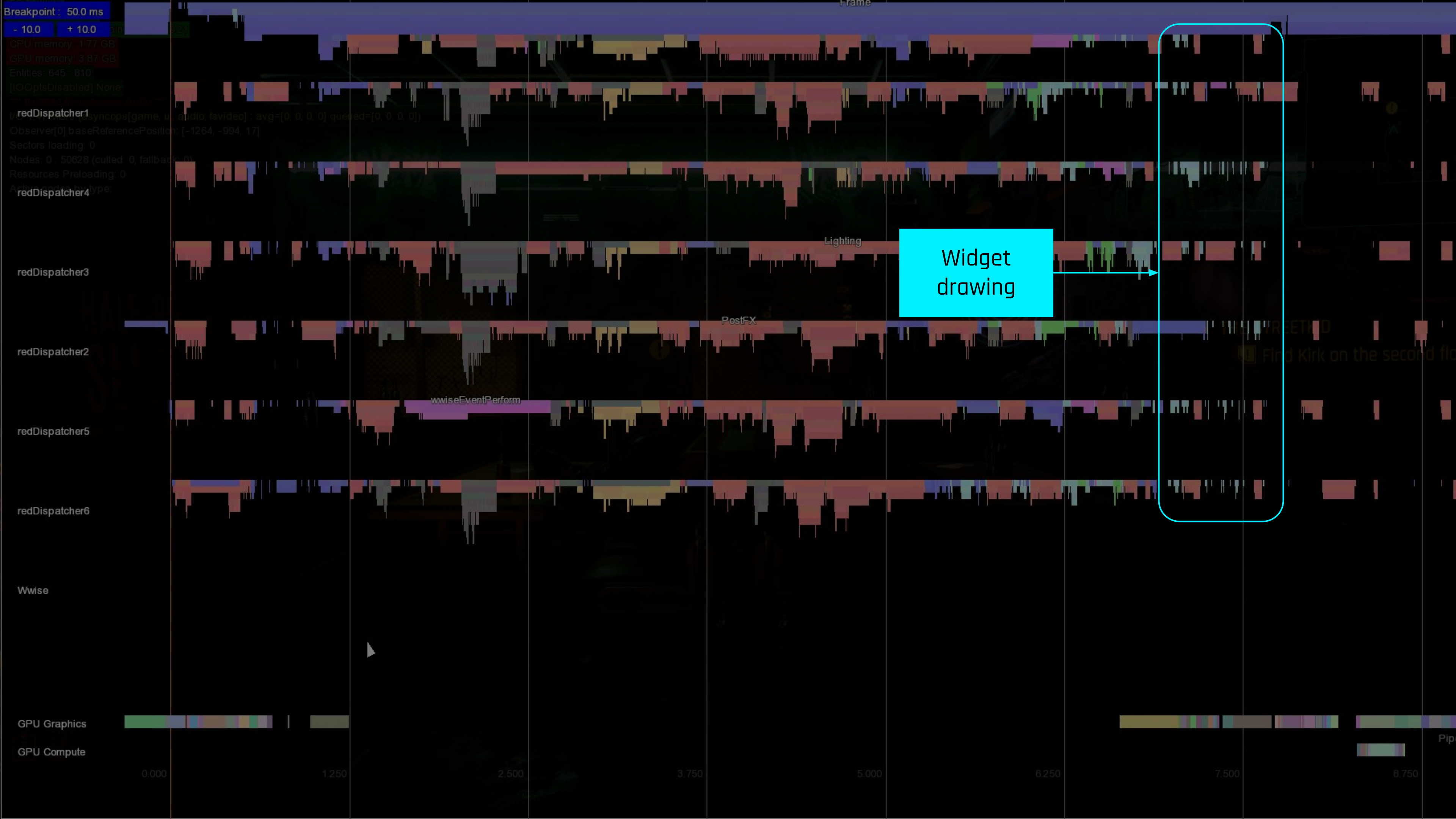


Input calculation & propagation
UI system preparation



Layers update
- UI game logic execution
- spawning
- animation update


Breakpoint : 50.0 ms
- 10.0
+ 10.0
Frame
redDispatcher1
redDispatcher4
redDispatcher3
redDispatcher2
redDispatcher5
redDispatcher6
Wwise
GPU Graphics
GPU Compute
Lighting
PostFX
wwiseEventPerform
Hierarchies arrangement
0.000
1.250
2.500
3.750
5.000
6.250
7.500
8.750


Breakpoint : 50.0 ms
- 10.0
+ 10.0
Frame
redDispatcher1
redDispatcher4
Lighting
redDispatcher3
Widget drawing
PostFX
redDispatcher2
wwiseEventPerform
redDispatcher5
redDispatcher6
Wwise
GPU Graphics
GPU Compute
0.000
1.250
2.500
3.750
5.000
6.250
7.500
8.750

UI FRAMEWORK DEVELOPMENT TIMELINE

JUNE 2017

MVP of UI framework

DECEMBER 2017

The first in-game UI

Internal demo

AUGUST 2018

Public Gameplay Reveal

Gamescom demo

SEPTEMBER 2018

Budgets are introduced into the game.

Big optimization has been started!

JANUARY 2019

Layer mechanism is introduced

The first set of layers decoupled UI correctly

TAKE AWAY

MULTITHREADING UI FRAMEWORK

- Extract all independent calculations and make them asynchronous
- Use separate render targets to draw UI asynchronously
- Cache everything what is possible
- Start UI update process in frame as soon as possible

OPTIMIZATIONS

S/N::: 1 / 052423
TASK STATUS: COMPLETE
TASK NAME:
GROUP UI INSTANCES

S/N::: 2 / 052423
TASK STATUS: ACTIVE
TASK NAME:
REDUCE UI UPDATES

S/N::: 3 / 052423
TASK STATUS: TO-DO
TASK NAME:
REDUCE UI ANIMATION UPDATES

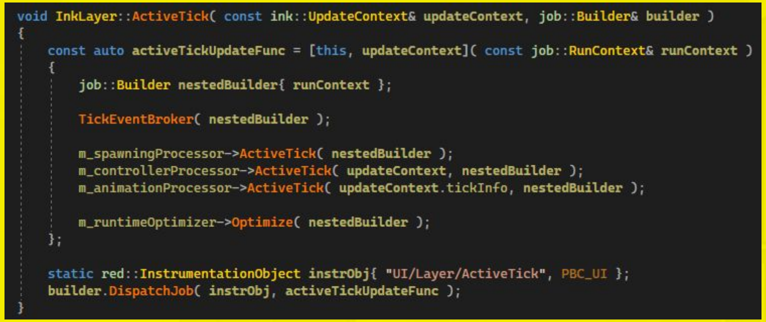
S/N::: 4 / 052423
TASK STATUS: TO-DO
TASK NAME:
UI INSTANCE CULLING

S/N::: 5 / 052423
TASK STATUS: TO-DO
TASK NAME:
DEFERRED UPDATE & DRAW

S/N::: 6 / 052423
TASK STATUS: TO-DO
TASK NAME:
HLOD FOR UI IN 3D-WORLD

ACTIVE & PASSIVE MODES

- Two types of logic execution
  - Active
    - UI is probably visible
    - All logic is executed
  - Passive
    - UI is not visible
    - Crucial part of logic is executed
- Custom definition per each layer
- Very flexible mechanism



```cpp
void InkLayer::ActiveTick( const ink::UpdateContext& updateContext, job::Builder& builder )
{
    const auto activeTickUpdateFunc = [this, updateContext]( const job::RunContext& runContext )
    {
        job::Builder nestedBuilder{ runContext };

        TickEventBroker( nestedBuilder );

        m_spawningProcessor->ActiveTick( nestedBuilder );
        m_controllerProcessor->ActiveTick( updateContext, nestedBuilder );
        m_animationProcessor->ActiveTick( updateContext.tickInfo, nestedBuilder );

        m_runtimeOptimizer->Optimize( nestedBuilder );
    };

    static red::InstrumentationObject instrObj{ "UI/Layer/ActiveTick", PBC_UI };
    builder.DispatchJob( instrObj, activeTickUpdateFunc );
}
```

UI TERMS DEFINITION - PART 2/3 : CONTROLLERS

Game controller

- Can be ticked (by default is off)
- Have access to all game system
- Is managed by controller processor
- Can be added only to UI instance
- Is controlled by a central system
- Examples:
  - CarGameController
  - ElevatorGameController
  - MinimapGameController
  - PaperdollGameController

Logic controller

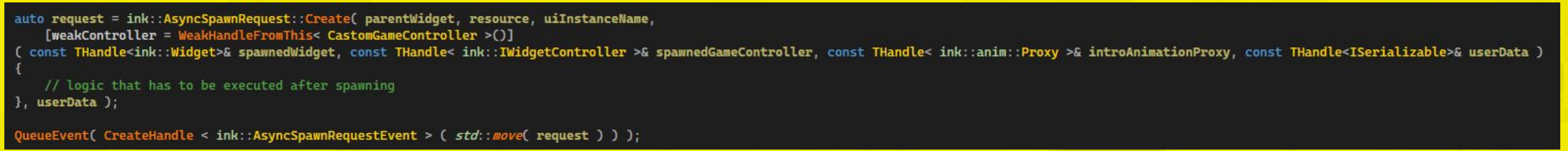
- Event based (no tick function)
- Contains UI logic only
- Has access only to owning widget and hierarchy below it
- Can be added into any widget
- Used to extend widget functionality
- Examples:
  - ButtonLogicController
  - CensorshipLogicController
  - SliderLogicController

CONTROLLER PROCESSOR

- contains all game controllers for a particular layer
- centralize tick execution
- decides about tick mode for each game controller
- pass a game systems context into game controllers
- spread execution on different threads
- decides about life time for all game controllers

SPAWNING PROCESSOR

- spawns UI instances asynchronously by default
- has a cap of active spawn processes per frame
- can deferred or cancel spawning process
- manages loading processes for dependent assets
- can use pools to reuse the same hierarchies
- attaching new instances sequentially



```
auto request = ink::AsyncSpawnRequest::Create( parentWidget, resource, uiInstanceName,
    [weakController = WeakHandleFromThis< CastomGameController >()]
( const THandle<ink::Widget>& spawnedWidget, const THandle< ink::IWidgetController >& spawnedGameController, const THandle< ink::anim::Proxy >& introAnimationProxy, const THandle<ISerializable>& userData )
{
    // logic that has to be executed after spawning
}, userData );

QueueEvent( CreateHandle < ink::AsyncSpawnRequestEvent > ( std::move( request ) ) );
```

USER ID_98877 LENS DRVR 3.0.03
STATUS: ONLINE SERIAL>34 73.32

TAKE AWAY

SEPARATE LOGIC BASED ON TICK FROM EVENTS

- Event based logic is relatively lightweight
- Keep strict control over tick based logic
- Disable tick always if possible
- Avoid data pulling mechanism on a big scale

OPTIMIZATIONS

S/N::: 1 / 052423
TASK STATUS: COMPLETE
TASK NAME:
GROUP UI INSTANCES

S/N::: 2 / 052423
TASK STATUS: COMPLETE
TASK NAME:
REDUCE UI UPDATES

S/N::: 3 / 052423
TASK STATUS: ACTIVE
TASK NAME:
REDUCE UI ANIMATION UPDATES

S/N::: 4 / 052423
TASK STATUS: TO-DO
TASK NAME:
UI INSTANCE CULLING

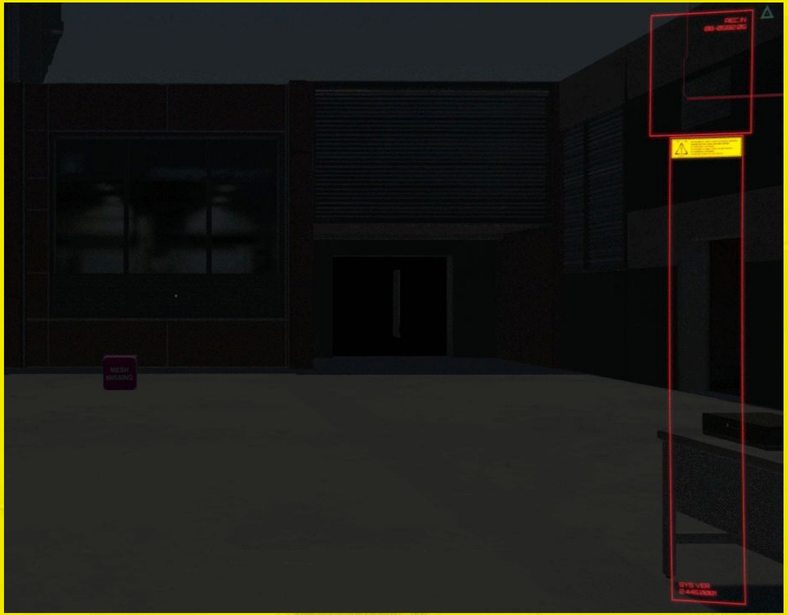
S/N::: 5 / 052423
TASK STATUS: TO-DO
TASK NAME:
DEFERRED UPDATE & DRAW

S/N::: 6 / 052423
TASK STATUS: TO-DO
TASK NAME:
HLOD FOR UI IN 3D-WORLD

EXAMPLE OF ANIMATION CHALLENGE

BRIEFINGS

- fully animated
- contain videos
- almost every text is localized
- can be split based on player choices
- any part can be played any time



UI TERMS DEFINITION - PART 3/3

Animation Definition (template)

Object that contains animation interpolators and events placed on timeline in a specific order. Order and properties don't change in runtime.

Animation Interpolator

Object that contains information how to interpolate specific animated property

Animation Instance

Refers to the animation definition and stores current values of interpolators while playing animation timeline based on delivered playback options. Can be changed in runtime.

ANIMATION PROCESSOR

- Active mode
  - Increment animation time
  - Asynchronous interpolators evaluation
  - Asynchronous values applying process
    - Sequential inside dependency bucket
  - Firing all events
- Passive mode
  - Increment animation time
  - Firing relevant events

TAKE AWAY

OPTIMIZATIONS FOR UI ANIMATIONS

- Keep single animation template in memory
- Use lightweight animation metadata for each instance
- Calculate and apply animated values only if effect is visible for the player
- For invisible animations update only their time

OPTIMIZATIONS

S/N::: 1 / 052423
TASK STATUS: COMPLETE
TASK NAME:
GROUP UI INSTANCES

S/N::: 2 / 052423
TASK STATUS: COMPLETE
TASK NAME:
REDUCE UI UPDATES

S/N::: 3 / 052423
TASK STATUS: COMPLETE
TASK NAME:
REDUCE UI ANIMATION UPDATES

S/N::: 4 / 052423
TASK STATUS: ACTIVE
TASK NAME:
UI INSTANCE CULLING

S/N::: 5 / 052423
TASK STATUS: TO-DO
TASK NAME:
DEFERRED UPDATE & DRAW

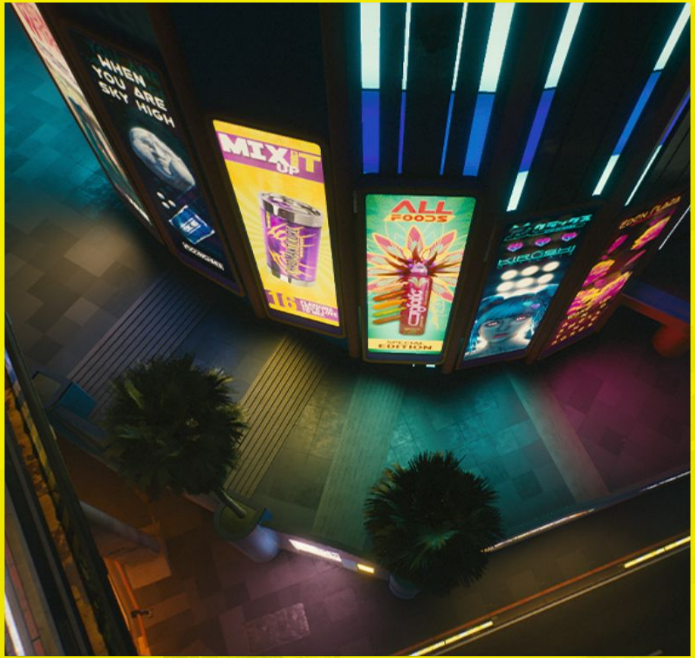
S/N::: 6 / 052423
TASK STATUS: TO-DO
TASK NAME:
HLOD FOR UI IN 3D-WORLD

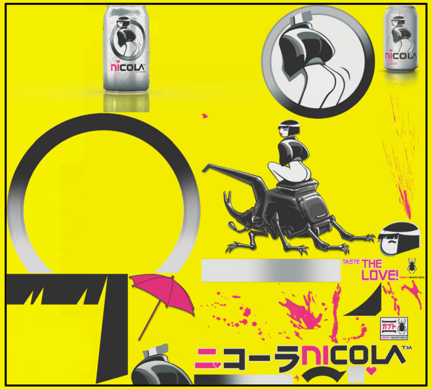
UI FRAMEWORK DEVELOPMENT TIMELINE

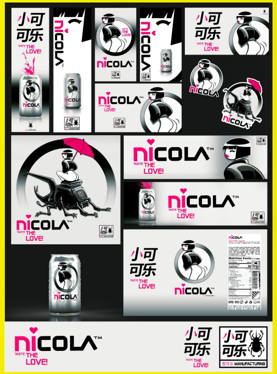
DECEMBER 2017	AUGUST 2018	SEPTEMBER 2018	JANUARY 2019	APRIL 2019
The first in-game UI Internal demo	Public Gameplay Reveal Gamescom demo	Budgets are introduced into the game. Big optimization has been started!	Layer mechanism is introduced The first set of layers decoupled UI correctly	**Advertisements layer was introduced**

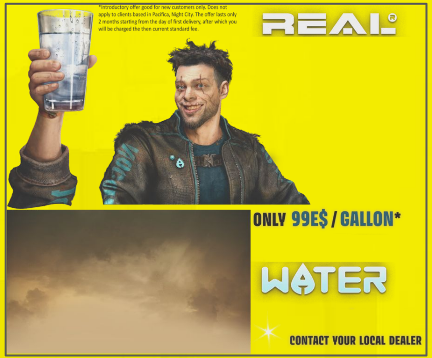
ADVERTISMENTS DESIGN

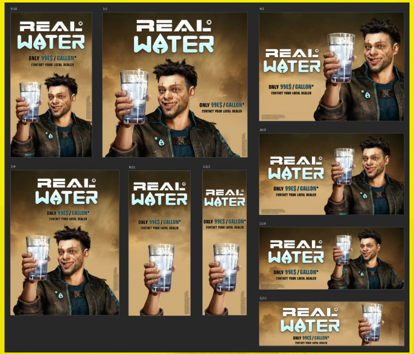
- Texture memory reduction
  - one texture atlas
  - many advertisements layouts
  - reusing render target memory
    - spawn and draw what is visible
- Possibility to animate advertisements
- Censorship filter
- Runtime randomization
- Custom lightning support







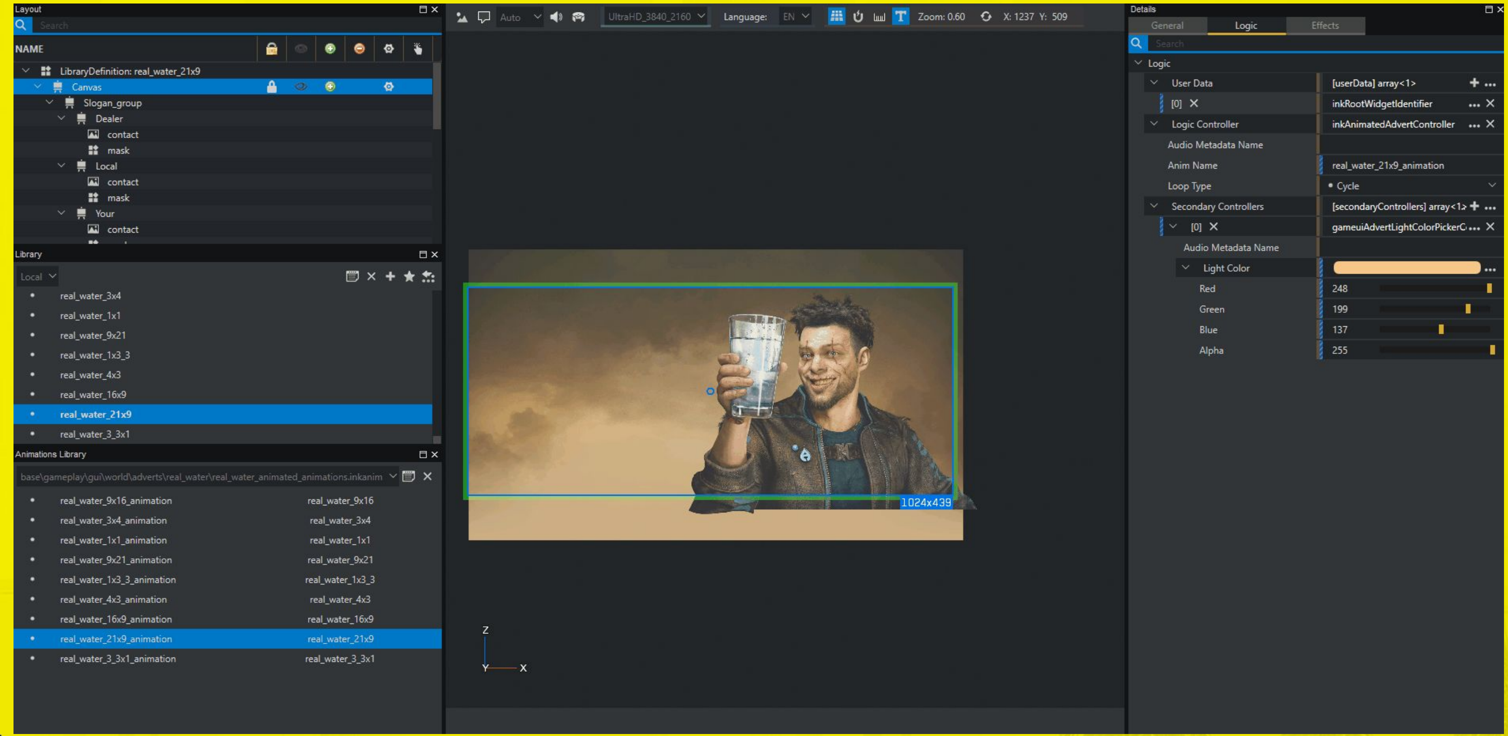




One texture atlas file

Many different advert layouts

ANIMATED VERSION FOR EACH ADVERT



Layout

NAME

- LibraryDefinition: real_water_21x9
  - Canvas
    - Slogan_group
      - Dealer
        - contact
        - mask
      - Local
        - contact
        - mask
      - Your
        - contact

Library

Local

- real_water_3x4
- real_water_1x1
- real_water_9x21
- real_water_1x3_3
- real_water_4x3
- real_water_16x9
- real_water_21x9
- real_water_3_3x1

Animations Library

base\gameplay\gui\world\adverts\real_water\real_water_animated_animations.inkanim

real_water_9x16_animation	real_water_9x16
real_water_3x4_animation	real_water_3x4
real_water_1x1_animation	real_water_1x1
real_water_9x21_animation	real_water_9x21
real_water_1x3_3_animation	real_water_1x3_3
real_water_4x3_animation	real_water_4x3
real_water_16x9_animation	real_water_16x9
real_water_21x9_animation	real_water_21x9
real_water_3_3x1_animation	real_water_3_3x1

Auto | UltraHD_3840_2160 | Language: EN | Zoom: 0.60 | X: 1237 Y: 509

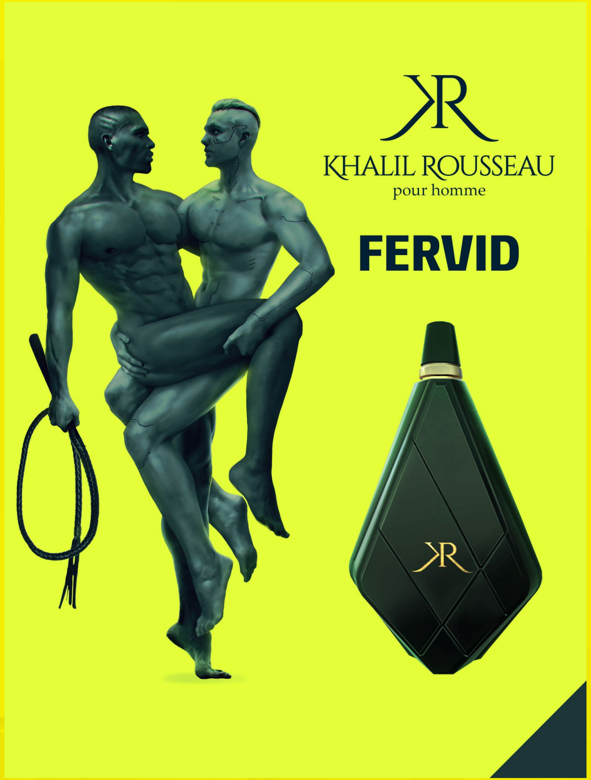
1024x439

Details

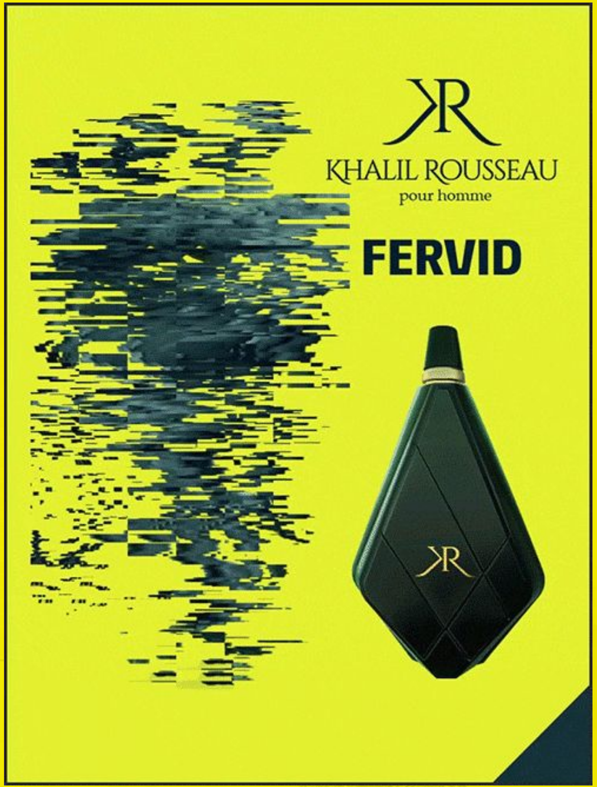
General | Logic | Effects

Logic

User Data	[userData] array<1>
[0]	inkRootWidgetIdentifier
Logic Controller	inkAnimatedAdvertController
Audio Metadata Name	
Anim Name	real_water_21x9_animation
Loop Type	Cycle
Secondary Controllers	[secondaryControllers] array<1>
[0]	gameuiAdvertLightColorPickerC…
Audio Metadata Name	
Light Color	
Red	248
Green	199
Blue	137
Alpha	255

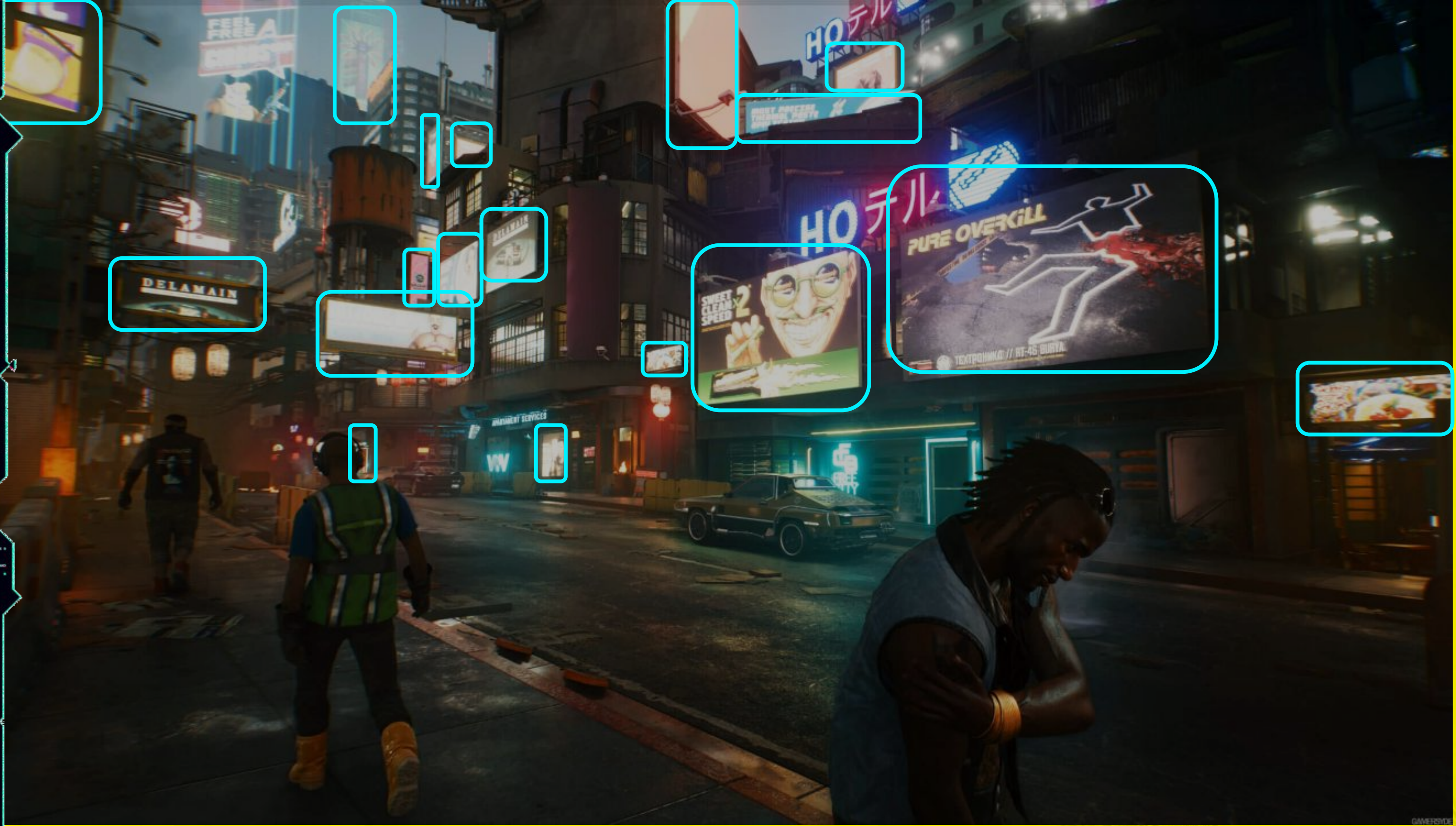


Original version



Censored version

HOW MANY OF THEM?



HOW MANY OF THEM?

DEVELOPMENT BUILD
50
455
10:00 AM
THE DAMNED
Go to The Moth bar.
All night. Every Night.
UPGRADE NOW
V3.0 FREE
2
SWEET CLEAN SPEED
BROSEPH
Draw Weapon
ALT
3.7KM
375M
T
V
X
3
3

PLAYER VIEW OPTIMIZATIONS

- Distance check
- Frustum culling
  - Rotation/movement prediction
  - Inertia mechanism
- Occlusion culling
  - Custom software implementation
- Screen coverage
  - Problem with "weapon plane"
- Static texture replacements
- “In a car” case
  - Streaming delay
  - Skipping update and draw

TAKE AWAY

VIEW OPTIMIZATIONS

- Use optimization pipelines as for normal 3d geometry
- Use passive mode if UI instance is invisible
- Adjust render quality of UI instance to its screen coverage

OPTIMIZATIONS

S/N::: 1 / 052423
TASK STATUS: COMPLETE
TASK NAME:
GROUP UI INSTANCES

S/N::: 2 / 052423
TASK STATUS: COMPLETE
TASK NAME:
REDUCE UI UPDATES

S/N::: 3 / 052423
TASK STATUS: COMPLETE
TASK NAME:
REDUCE UI ANIMATION UPDATES

S/N::: 4 / 052423
TASK STATUS: COMPLETE
TASK NAME:
UI INSTANCE CULLING

S/N::: 5 / 052423
TASK STATUS: ACTIVE
TASK NAME:
DEFERRED UPDATE & DRAW

S/N::: 6 / 052423
TASK STATUS: TO-DO
TASK NAME:
HLOD FOR UI IN 3D-WORLD

MORE AND MORE UI INSTANCES IN WORLD

- 5 global TV channels
  - 3 draw passes each
- Many UI instances in cars
- Big variety of icons

Visible hitches

UI FRAMEWORK DEVELOPMENT TIMELINE

AUGUST 2018

Public Gameplay Reveal

Gamescom demo

SEPTEMBER 2018

Budgets are introduced into the game.

Big optimization has been started!

JANUARY 2019

Layer mechanism is introduced

The first set of layers decoupled UI correctly

APRIL 2019

Advertisements layer was introduced

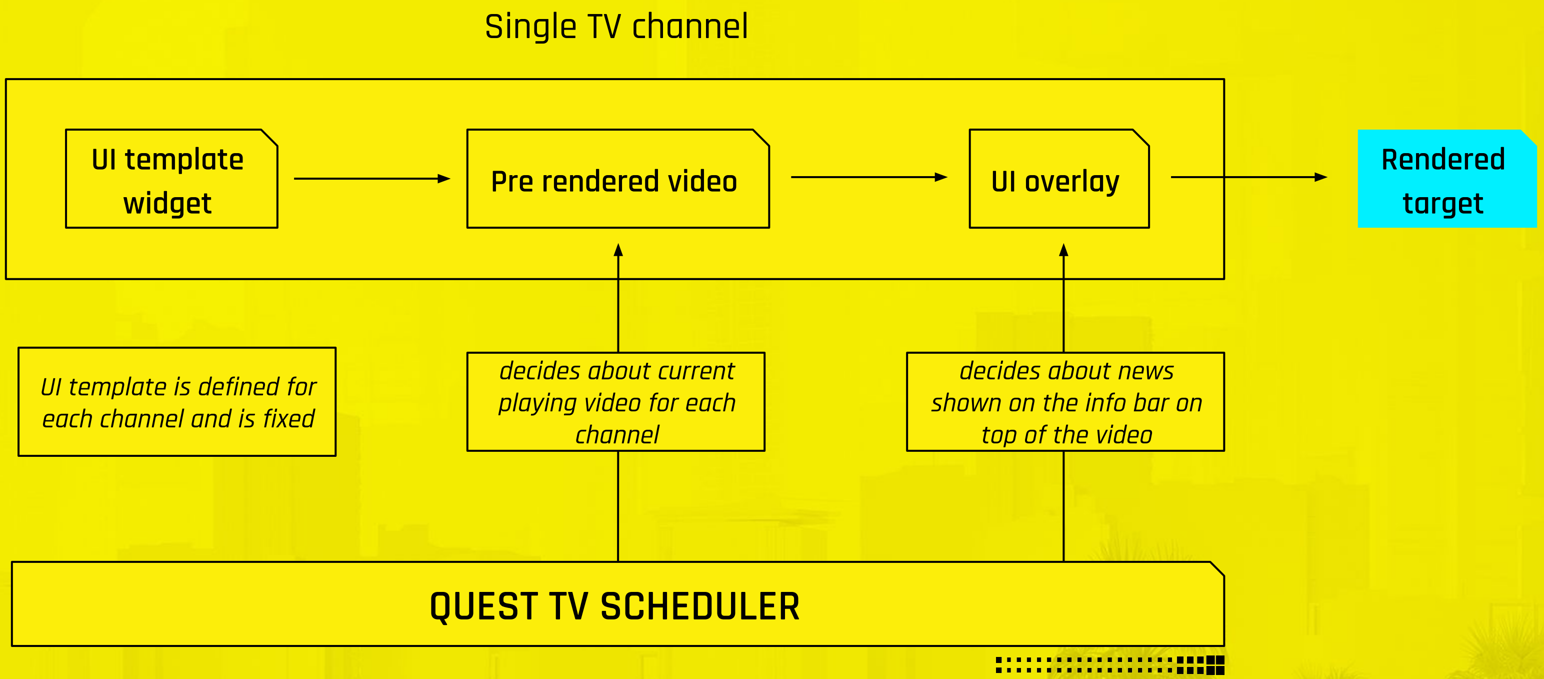
JULY 2019

Offscreen layer was introduced

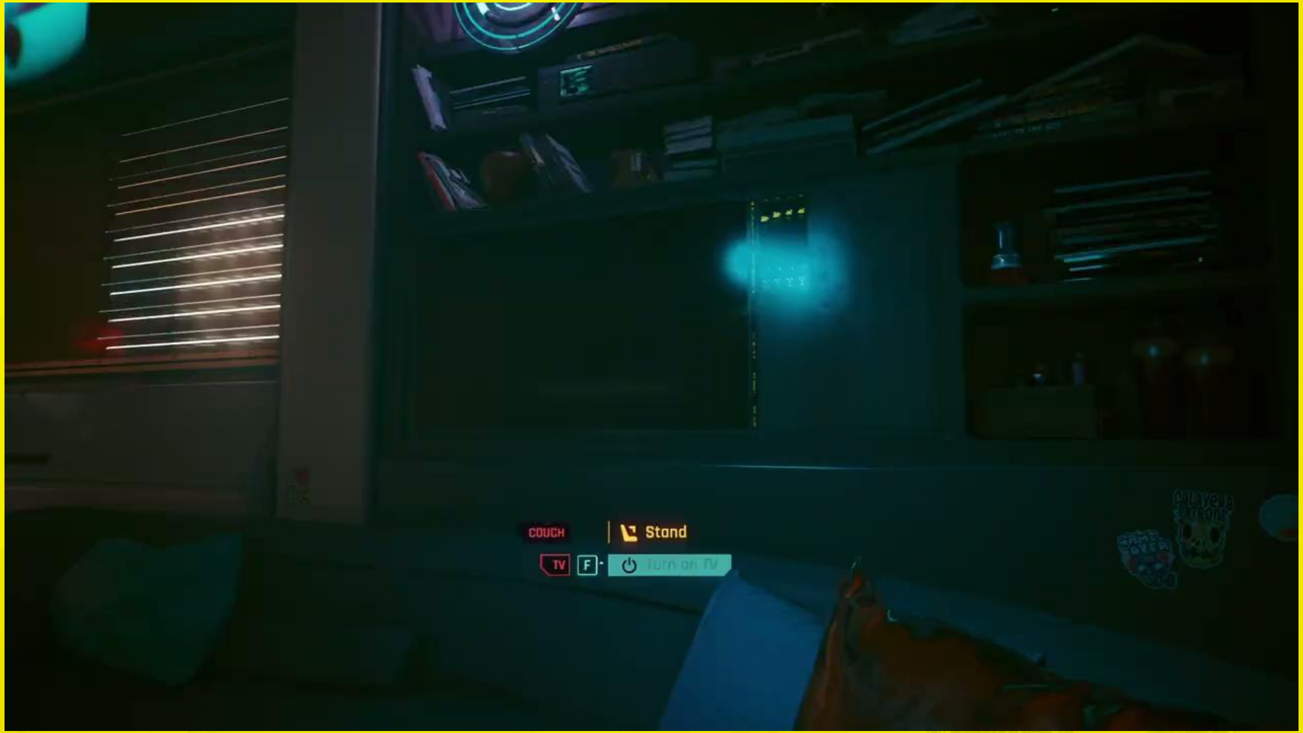
OFFSCREEN LAYER

- Mixes in-world and fullscreen approach
- Deferred processing
- Not blocking
- Works on pairs (UI resource and render target)
- Depends on the frame condition
- Use case examples:
  - inventory icons
  - global TV overlay
  - dynamic masks for complex effects

EXAMPLE OF OFFSCREEN PROCESSING


Single TV channel
UI template widget
Pre rendered video
UI overlay
Rendered target
UI template is defined for each channel and is fixed
decides about current playing video for each channel
decides about news shown on the info bar on top of the video
QUEST TV SCHEDULER

EXAMPLE RESULT OF OFFSCREEN DRAWING



TAKE AWAY

OFFSCREEN LAYER

- Cache and reuse as much as you can
- Use frame time only when it is available
- Fire and forget using independent render targets

OPTIMIZATIONS

S/N::: 1 / 052423
TASK STATUS: COMPLETE
TASK NAME:
GROUP UI INSTANCES

S/N::: 2 / 052423
TASK STATUS: COMPLETE
TASK NAME:
REDUCE UI UPDATES

S/N::: 3 / 052423
TASK STATUS: COMPLETE
TASK NAME:
REDUCE UI ANIMATION UPDATES

S/N::: 4 / 052423
TASK STATUS: COMPLETE
TASK NAME:
UI INSTANCE CULLING

S/N::: 5 / 052423
TASK STATUS: COMPLETE
TASK NAME:
DEFERRED UPDATE & DRAW

S/N::: 6 / 052423
TASK STATUS: ACTIVE
TASK NAME:
HLOD FOR UI IN 3D-WORLD

UI FRAMEWORK DEVELOPMENT TIMELINE

SEPTEMBER 2018

Budgets are introduced into the game.

Big optimization has been started!

JANUARY 2019

Layer mechanism is introduced

The first set of layers decoupled UI correctly

APRIL 2019

Advertisements layer was introduced

JULY 2019

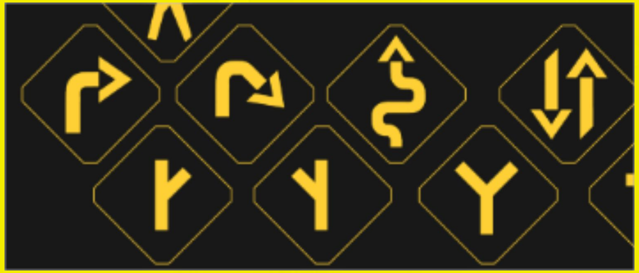
Offscreen layer was introduced

SEPTEMBER 2019

Street Signs extraction to separate layer

STREET SIGNS LAYER ASSUMPTIONS

Trinity St
KABUKI
WATSON



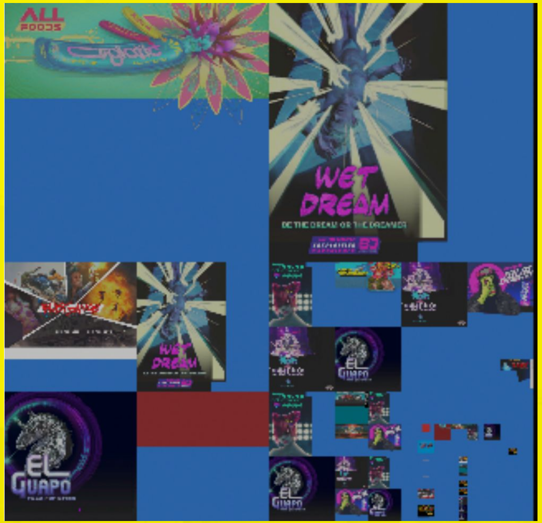
Orrell
CHARTER HILL
WESTBROOK

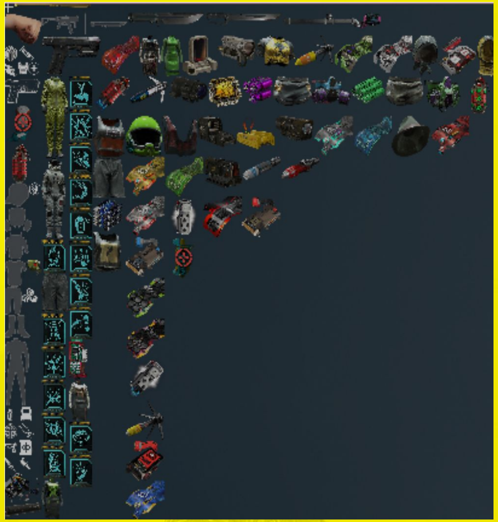
- Similar to advertisements
- Localized but not randomized
- Assembled in runtime
- Hundreds of them across the Night City
- Quick render target fragmentation

RENDER TARGETS MANAGER

- Solution for render target fragmentation
- Render targets as texture atlases
- Uses wrappers over render targets
- Complex matching mechanism
    - Supports many edge cases
    - Special render rules for icons
- Separate render target pools (3D UI & Effects)

EXAMPLE RESULT OF HLOD RENDER TARGET





TAKE AWAY

RENDER TARGET MANAGER

- In-world UI can be very often downscaled
- UI screen coverage is a good indicator of quality factor
- Draw to smaller area inside the render target instead of downscaling render target itself

OPTIMIZATIONS

S/N::: 1 / 052423
TASK STATUS: COMPLETE
TASK NAME:
GROU

S/N::: 2 / 0524
TASK STAT
TASK NAME:
REDU

S/N::: 3 / 0524
TASK STAT
TASK NAME:
REDUCE OF ANIMATION
UPDATES

S/N::: 4 / 052423
TASK STATUS: COMPLETE

HLOD FOR UI IN
3D-WORLD

345 / 209 / DBJ

ALL TASKS COMPLETED

WHY STILL THAT LONG?

Many many other UI mechanisms for optimization but it is a story for another time!

- Hierarchy caching
  - cache widget, HUD caching
- Data optimizer
  - font, regex reduction
- Data driven approach
  - inventory item pooling, web pages templates, device templates, styles
- Text animation
- Text parameters
- Markup language
- Interaction with UI in 3D on complex meshes

UI FRAMEWORK DEVELOPMENT TIMELINE

- **JANUARY 2019** — Layer mechanism is introduced

  The first set of layers decoupled UI correctly
- **APRIL 2019** — Advertisements layer was introduced
- **JULY 2019** — Offscreen layer was introduced
- **SEPTEMBER 2019** — Street Signs extraction to separate layer
- **DECEMBER 2020** — **Cyberpunk 2077 release**

USER ID_98877 LENS DRVR 3.0.03
STATUS: ONLINE SERIAL>34 73.32

Model No.: CS-AUS551NB
Part No.: C31-S551, 31CP7/65/80
Rating: 11.1VDC=4500mAh

TAKE AWAY

GLOBAL TAKEAWAYS

- Deadlines and budgets are your friends
- Code decoupling and parallel execution were key solutions
- You must have an amazing team to achieve amazing things

UI TEAM

- Started as a team of 3 developers
- Finished as 3 UI teams (art, design, code)
- Max peak in UI Code: 11 developers

CREDITS TO:

Adamo Maiorano, Yaroslav Getsevich, Przemysław Machniewski, Galust Saakov, Bartłomiej Wardziński, Natalia Nowacka, Jeremiasz Kacprzak, Adam Dumała

Additional thanks:

Monika Janowska, Robert Bielecki, Jonathan Huot, Andrzej Uszakow, Aleksandra Lato, Artur Wyszyński, Przemysław Banasiak, Wojciech Czarny, UI Design Team and UI Art Team

UI TEAM

- Started as a team of 3 developers
- Finished as [illegible] 3 UI teams (art, design, code)
- Max peak in UI Code: 11 developers

CREDITS TO:

Adamo Maiorano, Yaroslav [illegible] ewski, Galust Saakov, Barttomiej Wardzinski, Natalia Nowacka, Jeremiasz Kacprzak, Adam Dumala

Additional thanks:

Monika Janowska, Robert Bielecki, Jonathan Huot, Andrzej Uszakow, Aleksandra Lata, Artur Wyszyński, Przemysław Banasiak, Wojciech Czarny, UI Design Team and UI Art Team

THANK YOU!

CD PROJEKT RED®

USER ID_98877 LENS DRVR 3.0.03
STATUS: ONLINE SERIAL>34 73.32

8234299 -AXRCG 0001