Tips and Tricks to Get the Most out of Your Virtual-Reality Experiences in Stingray

Olivier Dionne Software Development Manager

Benjamin Slapcoff Software Engineer

Andrew Grant Product Manager

Outline

Stingray Renderer Overview
Profiler and VR Optimizations
Building Immersive Experiences
Content Tips and Tricks

Warhammer Vermintide - Fatshark, [PC | PS4 | XBox]

War of the Roses- Fatshark [PC]

Helldivers - Arrowhead (Sony) [PC | PS4 | PS3 | PS Vita]

Gauntlet - Arrowhead (Sony) [PS4, PC]

The Showdown Effect - Arrowhead [PC]

Autodesk Stingray V1.6

Autodesk Stingray V1.6

Autodesk Stingray V1.6

Autodesk Stingray V1.6

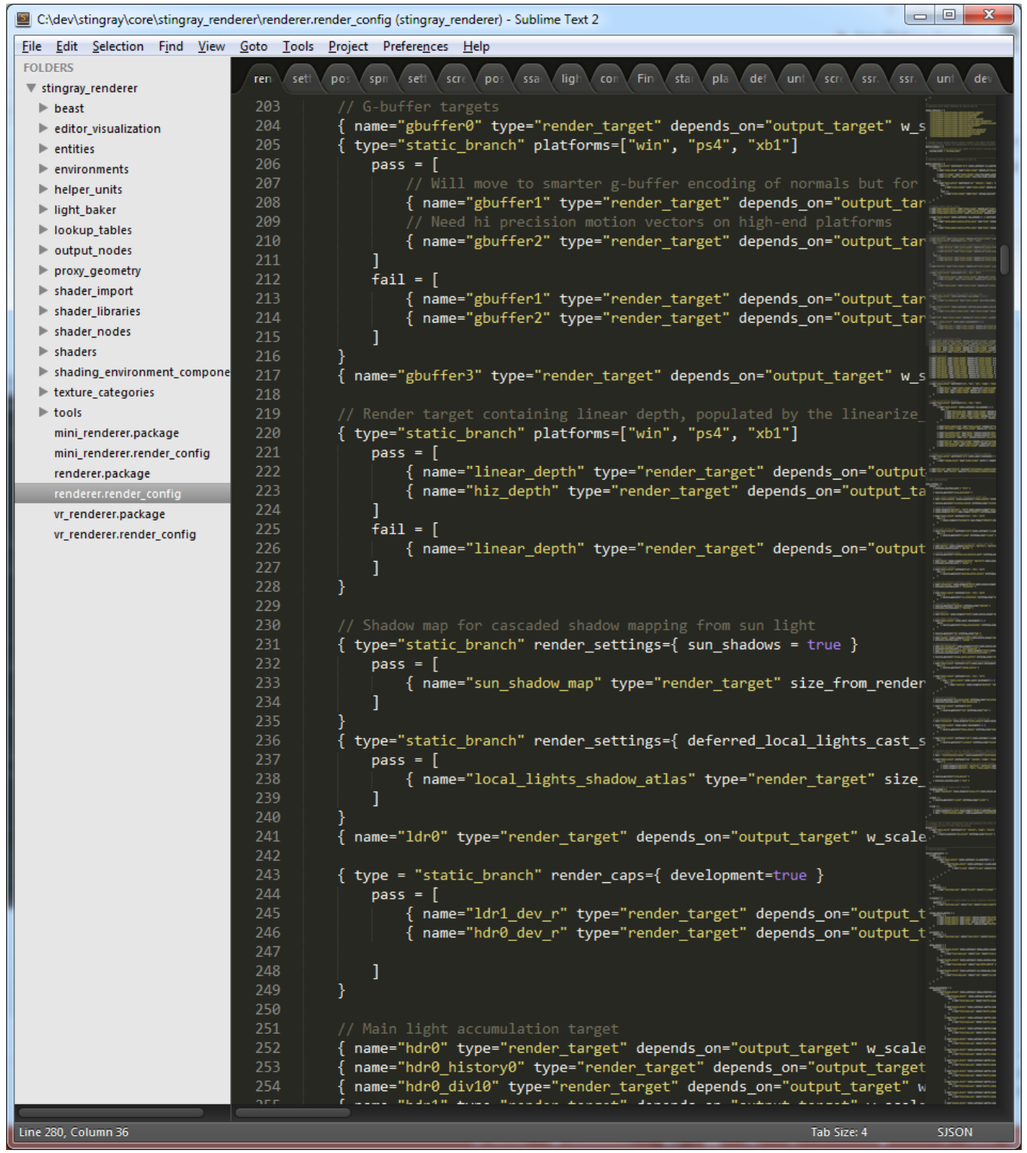
Stingray VR Project
Autodesk Stingray V1.6

Flexible Data-Driven Renderer

- Shaders, resource creation, resource manipulation and flow of rendering pipeline entirely defined in ***data***
- Data is expressed in SJSON
  - Hot-reloadable for quick iteration times
  - Allows for fast experimentation and debugging

Stingray render_config

- Ties all rendering sub-systems
- Dictates the order of operations for a frame
- Defines quality settings, device capabilities and default shader libraries to load
- Three key ingredients
  - Resource sets – memory allocations / deallocations
  - Resource generators – resource updating
  - Layer configurations – frame scheduling



Challenges in VR

- High refresh rates: 90Hz (11.11ms per frame)
- Frame buffer: 2160x1200
- Super sampled to reduce aliasing artifacts
  - Off-screen buffer scaled by 1.5x in each dimension (3240x1800)

- Shaded visible pixels

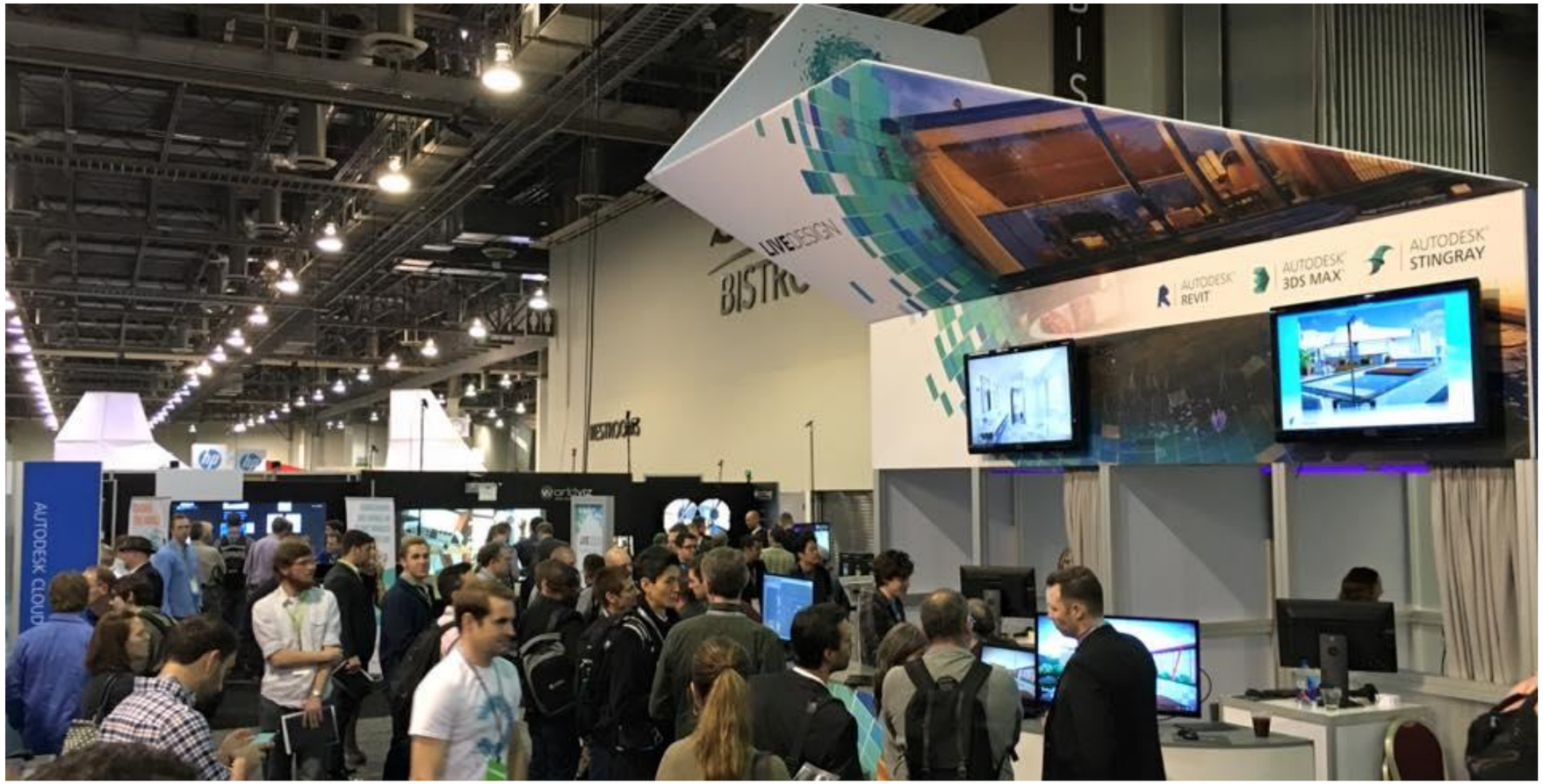
Resolution	Refresh Rate (Hz)	MPixels per Second
720p	60	55
1080p	60	124
2160p	60	498
VR (3240x1800)	**90**	**525**

Stingray VR – Humble beginnings…

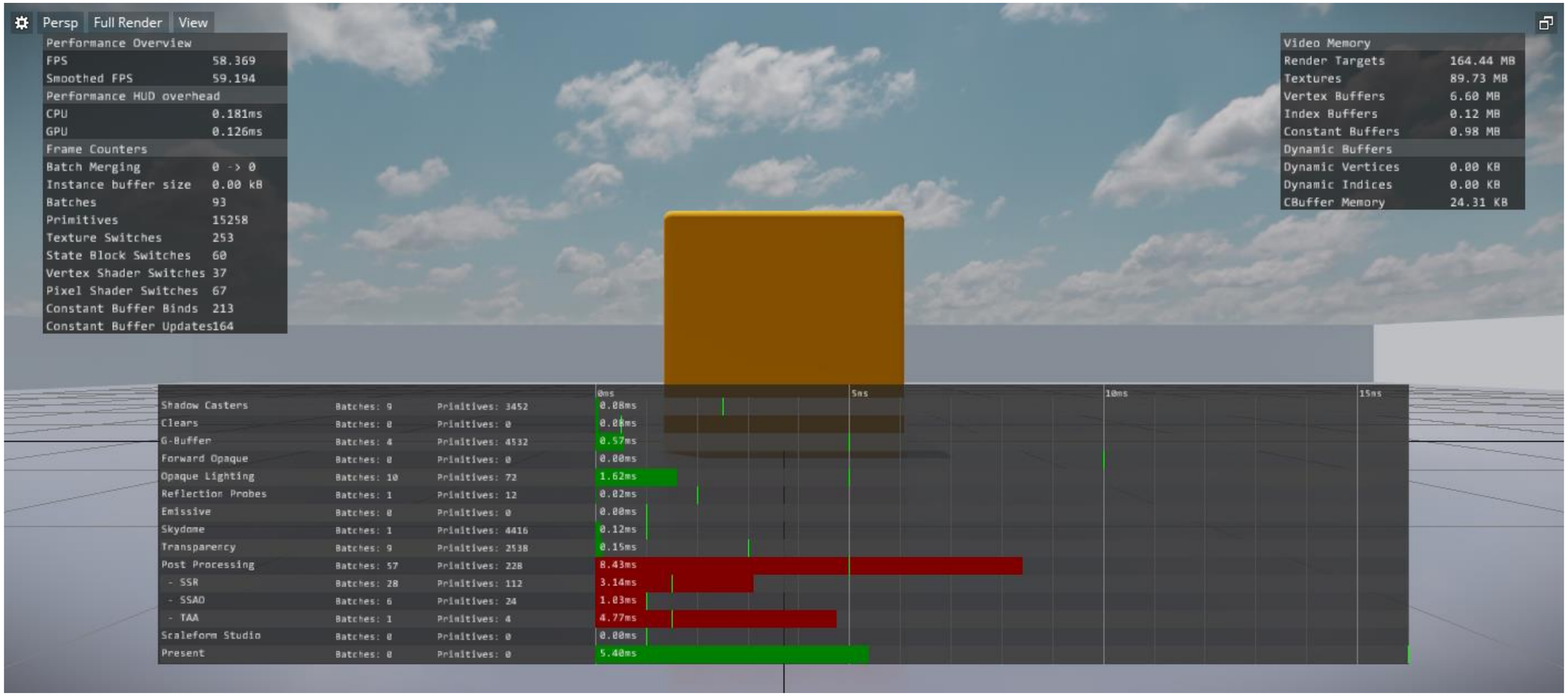
- Sequential stereo rendering
  - Two distinct scene cameras to represent eyes. Submit entire scene for left eye, then resubmit scene again for right eye.
- Pros:
  - Easiest strategy to start supporting VR devices and their tracking systems
- Cons:
  - Inefficient: Draw calls and state changes are doubled
  - Not CPU or GPU cache friendly!

Stingray VR – Humble beginnings…

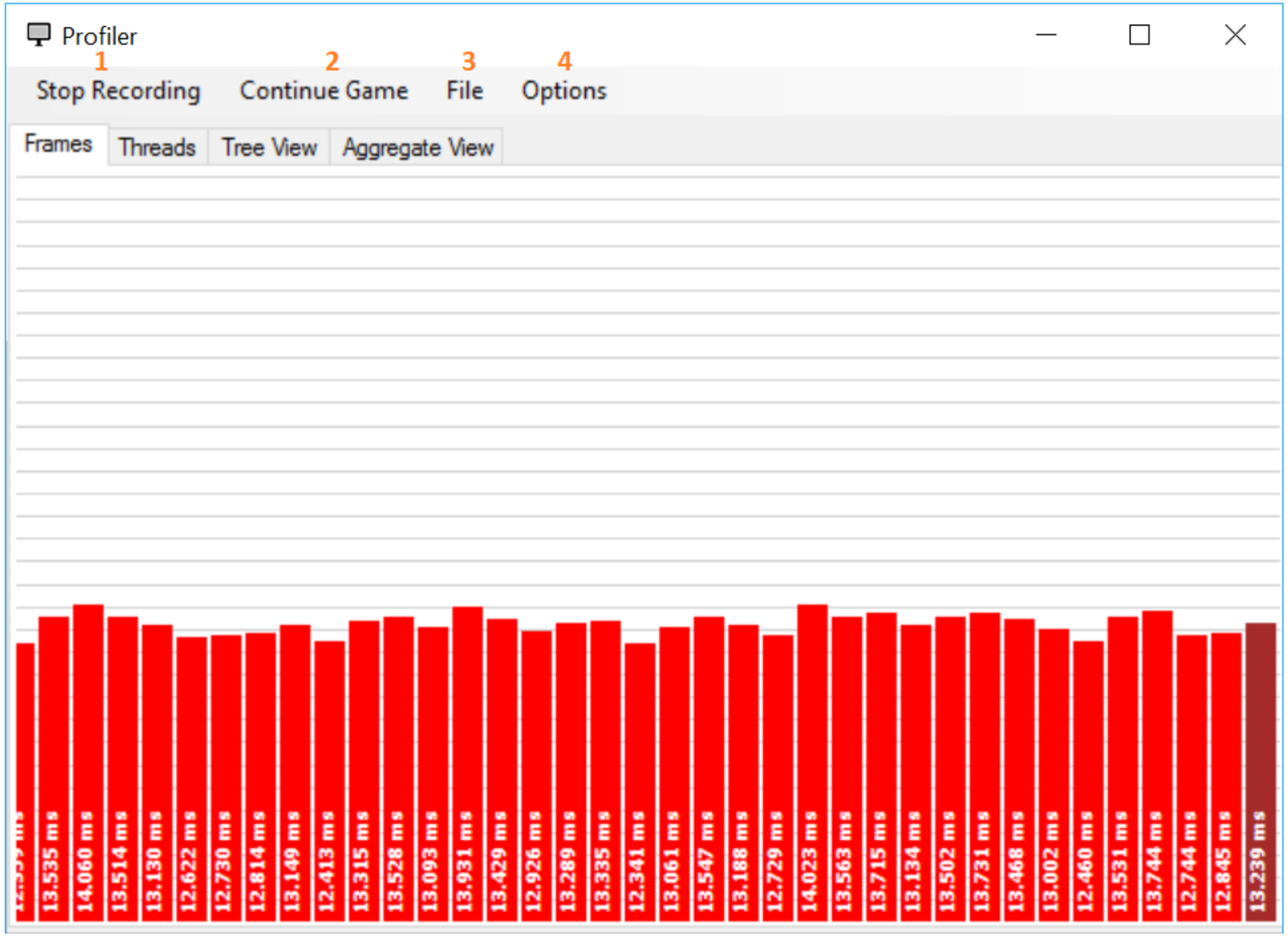
- Live Design Booth AU2015



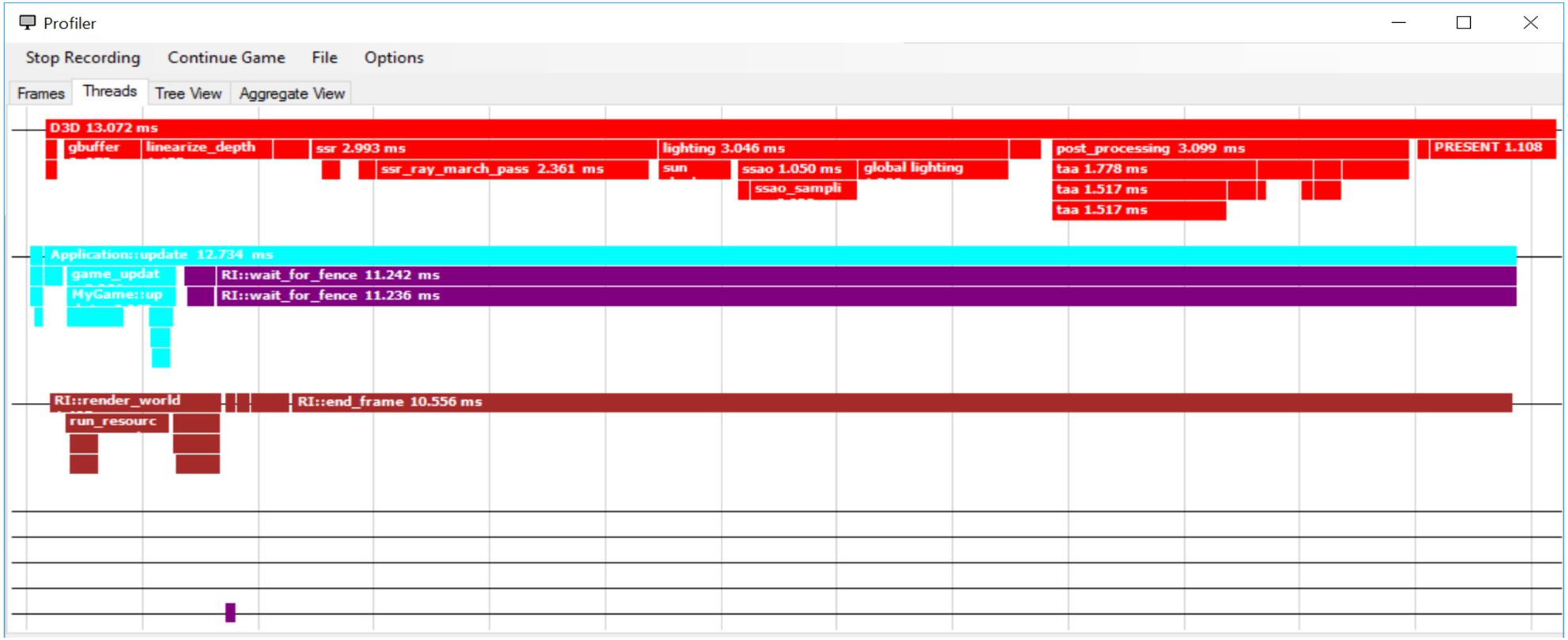
Stingray Profiler



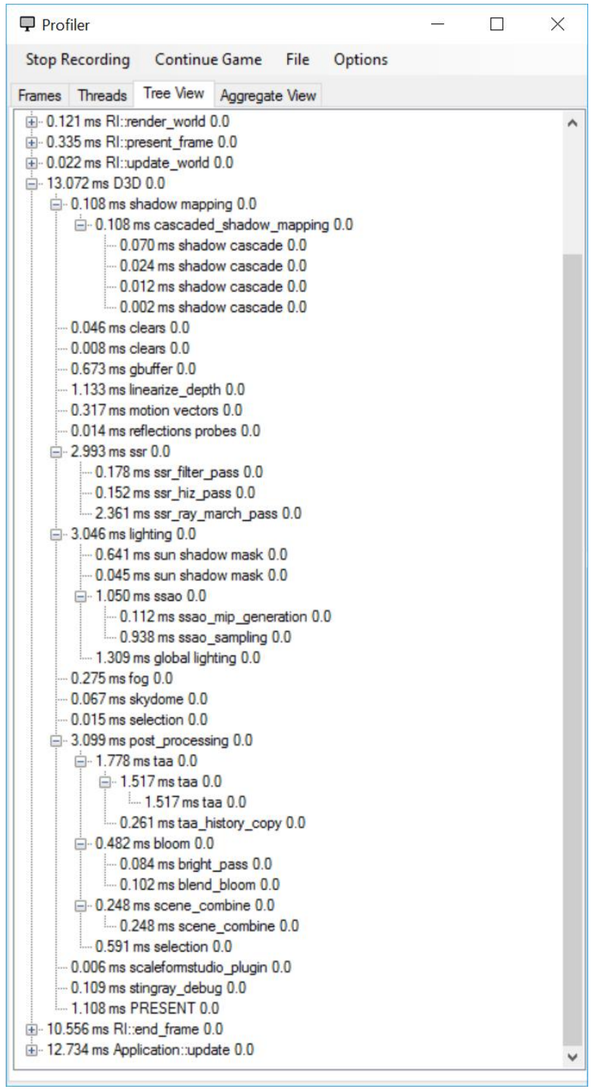
Stingray Profiler

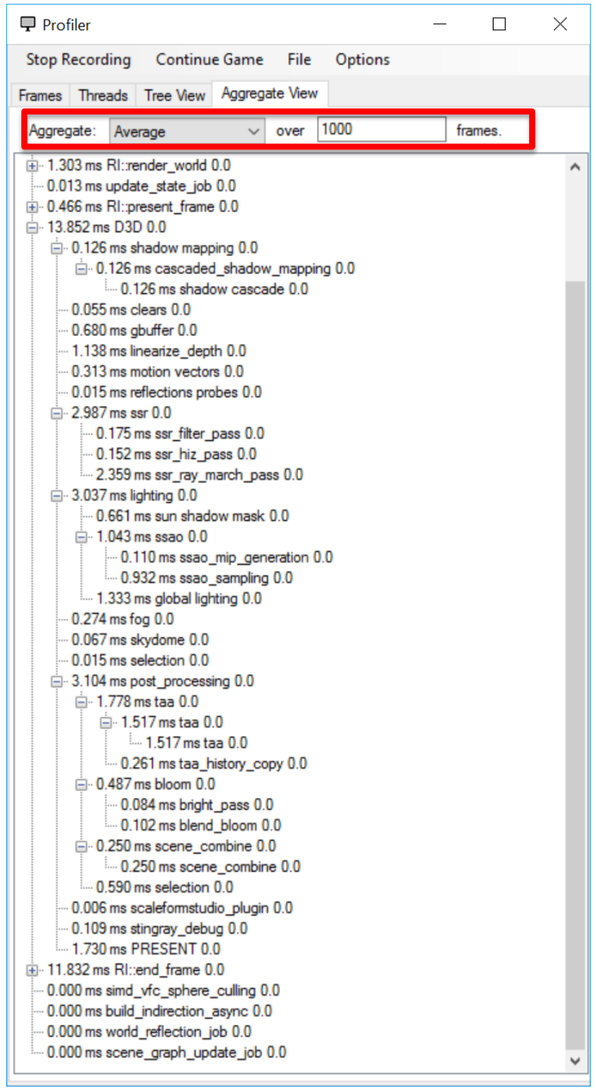


Stingray Profiler



Stingray Profiler





Stingray VR – Humble beginnings…

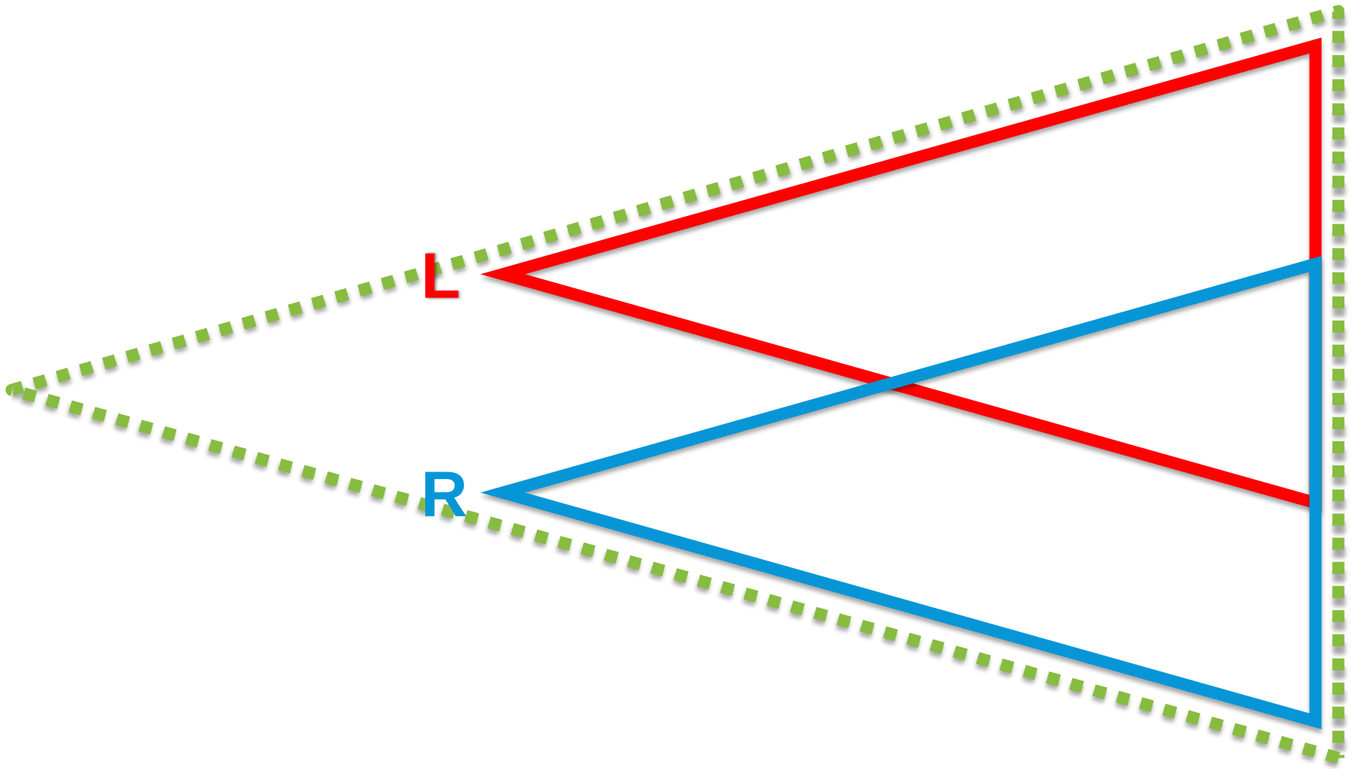
"Gentlemen, we can rebuild him. We have the technology…"

Stingray VR – Optimizations

- Frustums are a big deal…

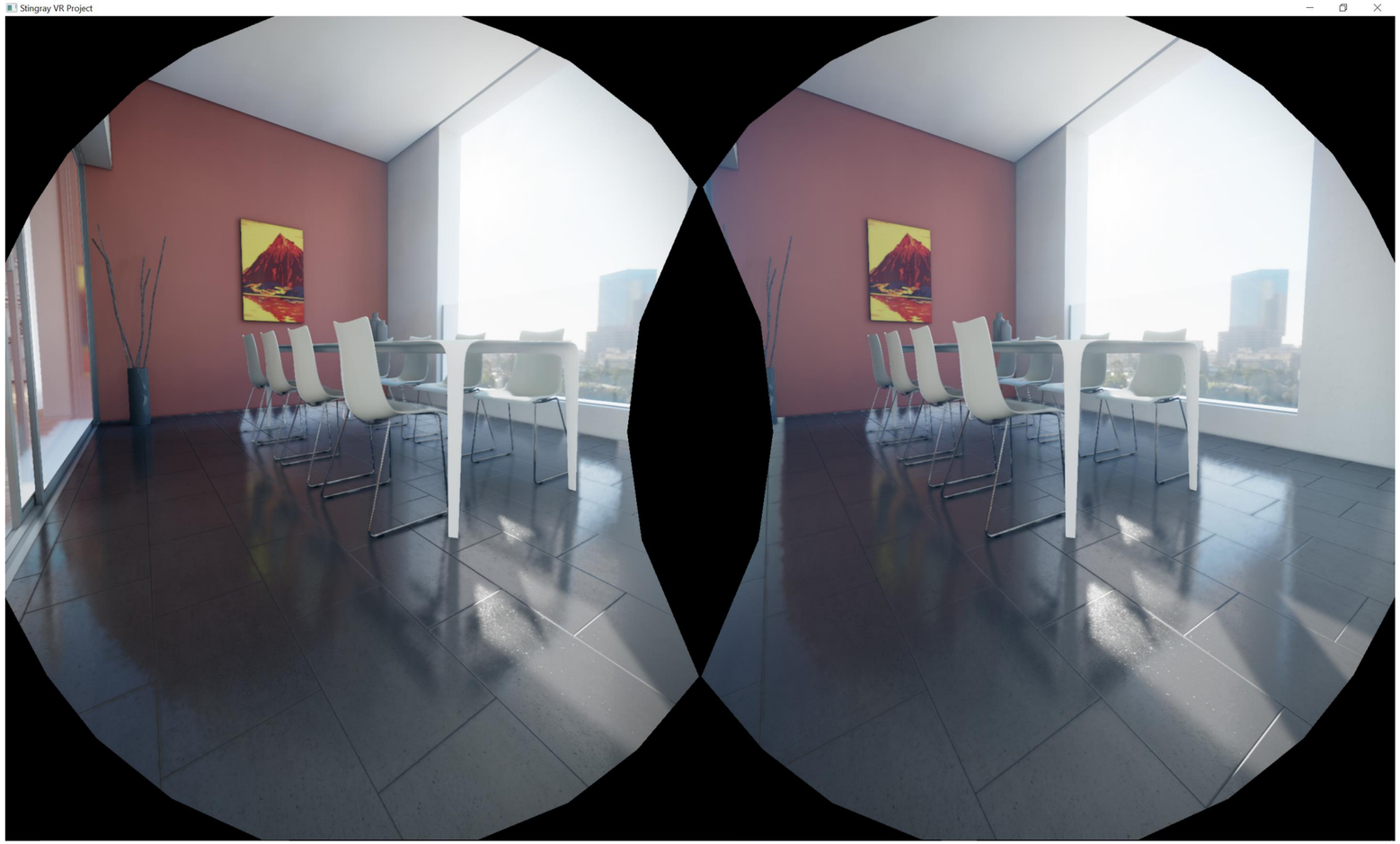
Stingray VR – Optimizations

- Compute single compound frustum to incorporate both eyes



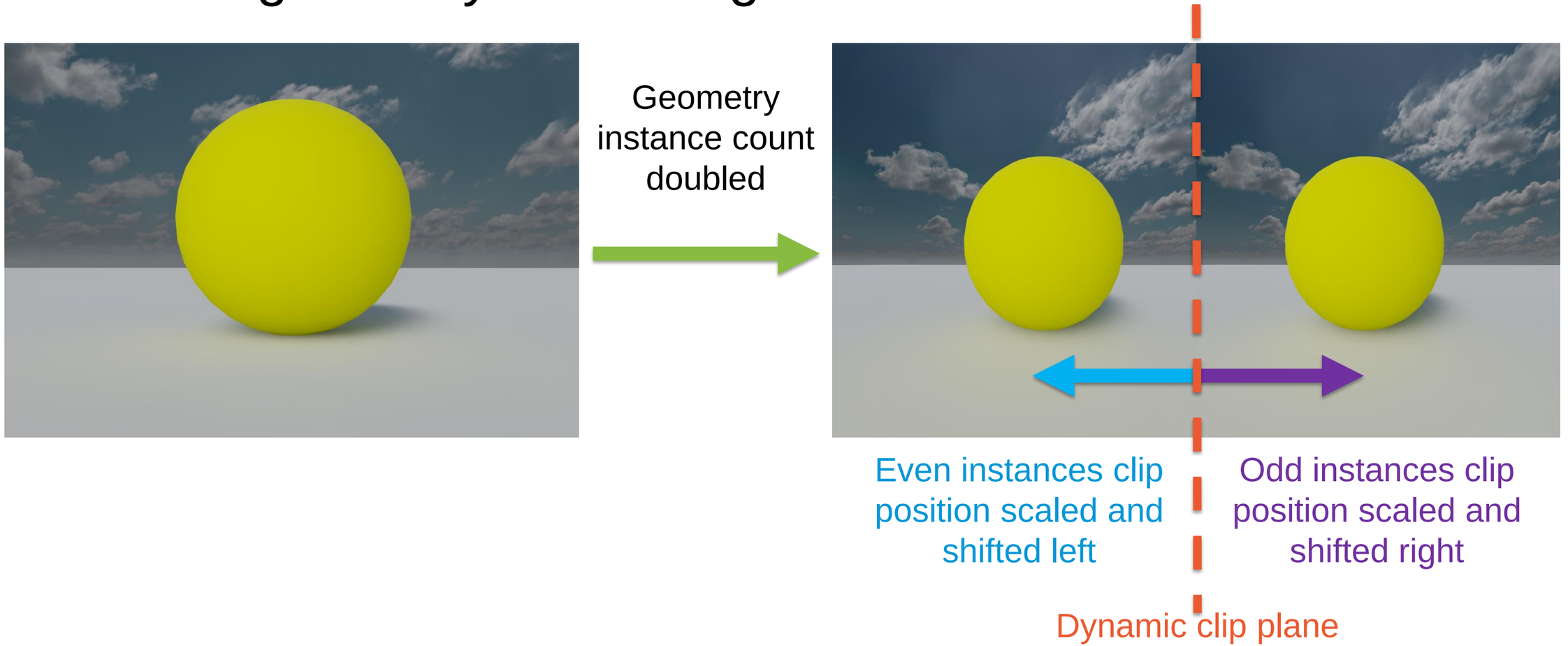
Stingray VR – Optimizations

- Bail early on non-visible (depth and stencil compare)

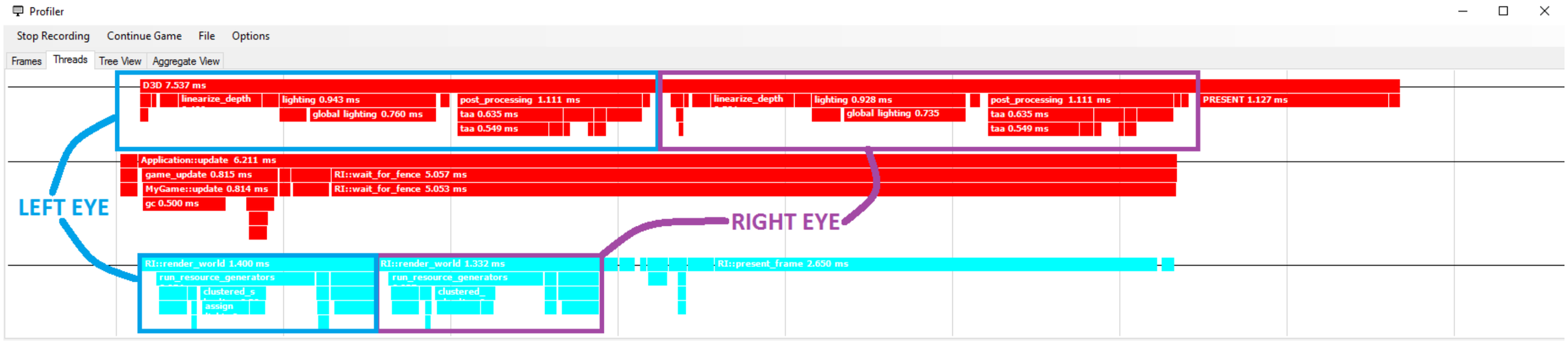


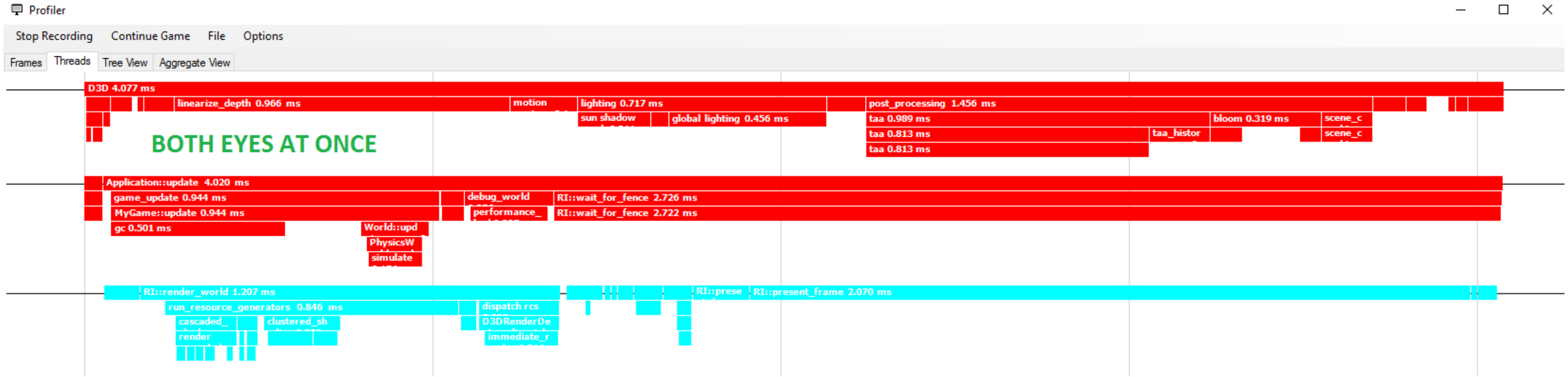
Stingray VR – Optimizations

- GPU geometry instancing to render in stereo



Stingray VR - Today





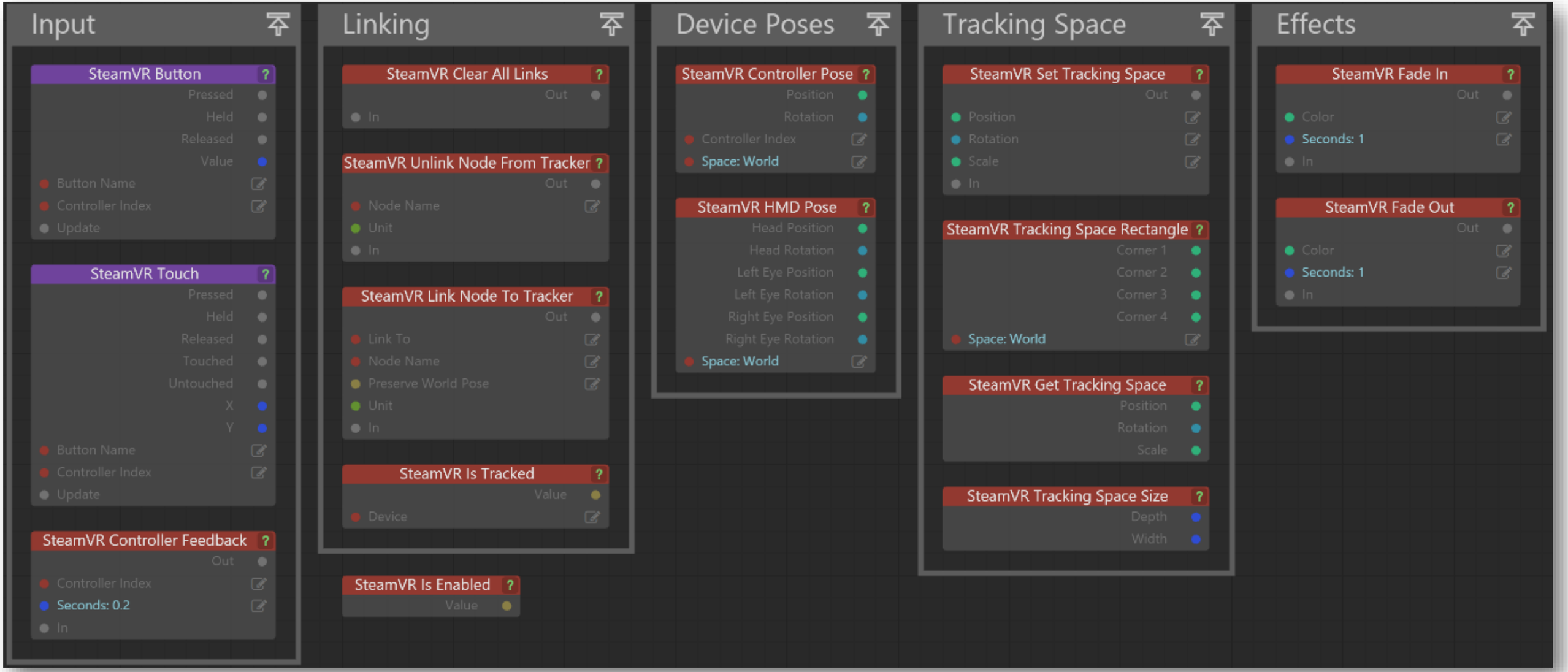
Stingray VR – What's Next

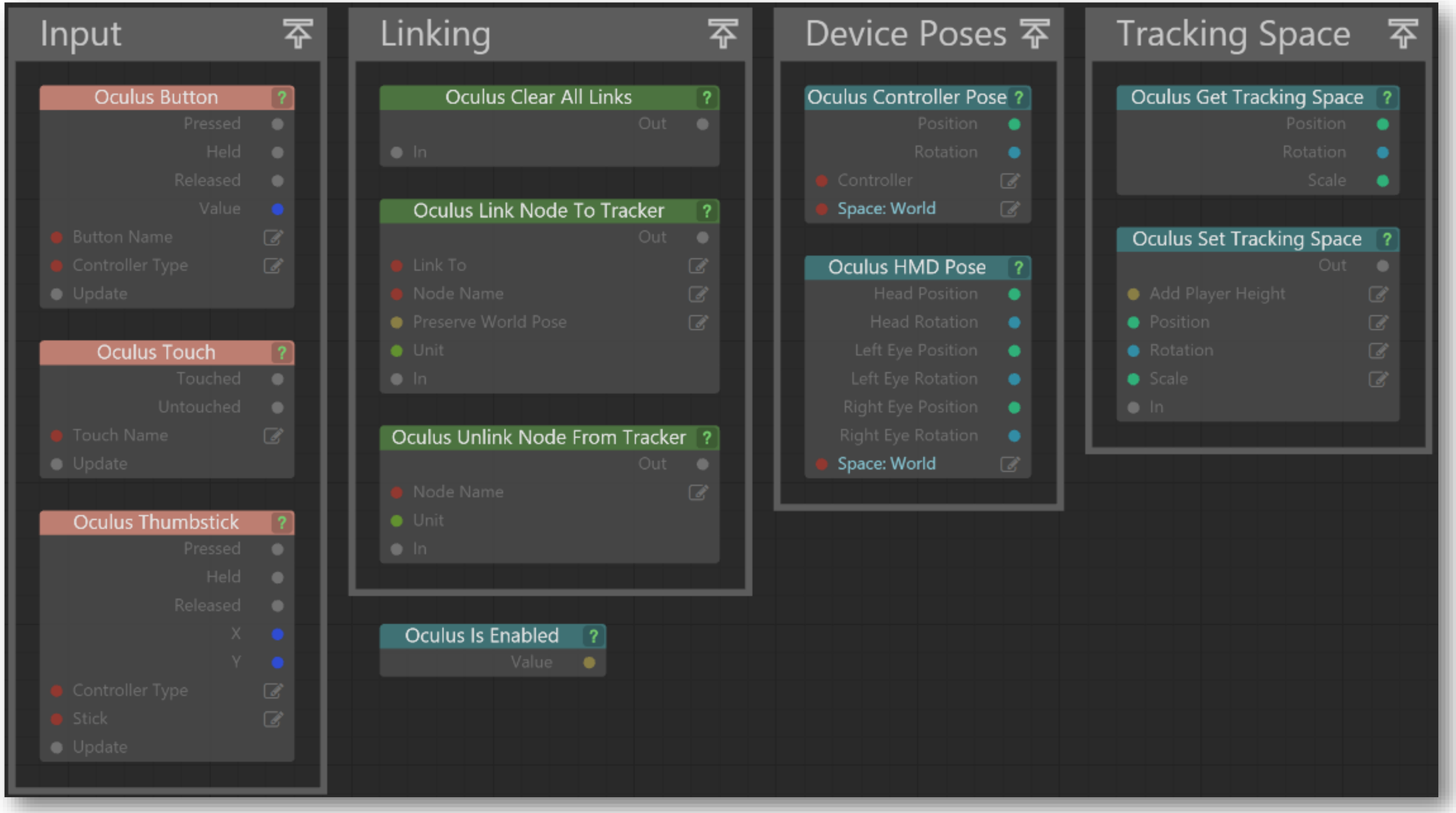
- Adaptive quality
- Foveated rendering
- Multi GPU support
- Mobile optimizations

Building VR Experiences using Stingray

Stingray VR Templates

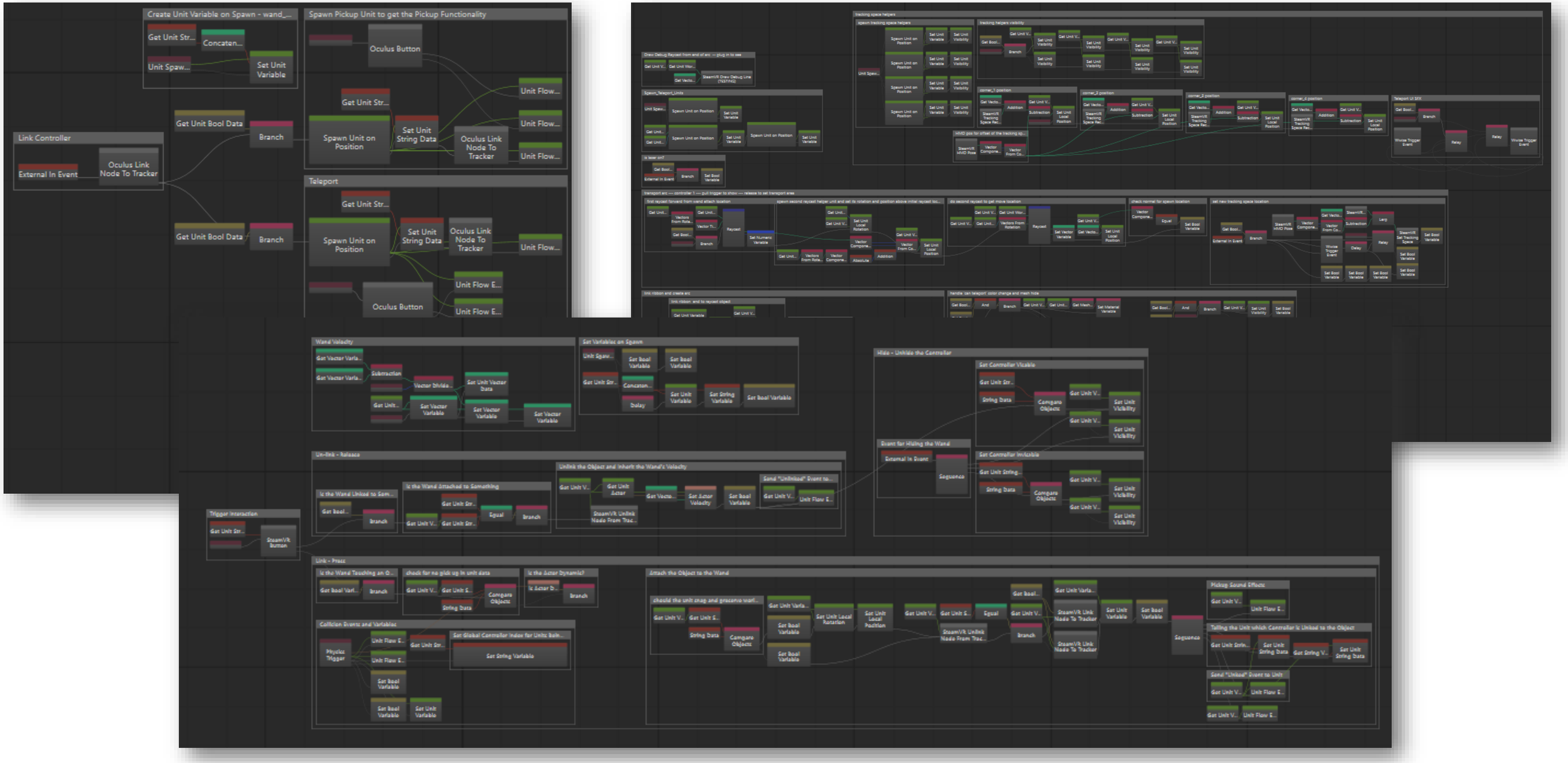
- Template for both Oculus and HTC Vive
- Very similar functionality exposed in both the Lua API and Flow


Input
SteamVR Button
Pressed
Held
Released
Value
Button Name
Controller Index
Update
SteamVR Touch
Pressed
Held
Released
Touched
Untouched
X
Y
Button Name
Controller Index
Update
SteamVR Controller Feedback
Out
Controller Index
Seconds: 0.2
In
Linking
SteamVR Clear All Links
Out
In
SteamVR Unlink Node From Tracker
Out
Node Name
Unit
In
SteamVR Link Node To Tracker
Out
Link To
Node Name
Preserve World Pose
Unit
In
SteamVR Is Tracked
Value
Device
SteamVR Is Enabled
Value
Device Poses
SteamVR Controller Pose
Position
Rotation
Controller Index
Space: World
SteamVR HMD Pose
Head Position
Head Rotation
Left Eye Position
Left Eye Rotation
Right Eye Position
Right Eye Rotation
Space: World
Tracking Space
SteamVR Set Tracking Space
Out
Position
Rotation
Scale
In
SteamVR Tracking Space Rectangle
Corner 1
Corner 2
Corner 3
Corner 4
Space: World
SteamVR Get Tracking Space
Position
Rotation
Scale
SteamVR Tracking Space Size
Depth
Width
Effects
SteamVR Fade In
Out
Color
Seconds: 1
In
SteamVR Fade Out
Out
Color
Seconds: 1
In


Input
Oculus Button
Pressed
Held
Released
Value
Button Name
Controller Type
Update
Oculus Touch
Touched
Untouched
Touch Name
Update
Oculus Thumbstick
Pressed
Held
Released
X
Y
Controller Type
Stick
Update
Linking
Oculus Clear All Links
Out
In
Oculus Link Node To Tracker
Out
Link To
Node Name
Preserve World Pose
Unit
In
Oculus Unlink Node From Tracker
Out
Node Name
Unit
In
Oculus Is Enabled
Value
Device Poses
Oculus Controller Pose
Position
Rotation
Controller
Space: World
Oculus HMD Pose
Head Position
Head Rotation
Left Eye Position
Left Eye Rotation
Right Eye Position
Right Eye Rotation
Space: World
Tracking Space
Oculus Get Tracking Space
Position
Rotation
Scale
Oculus Set Tracking Space
Out
Add Player Height
Position
Rotation
Scale
In

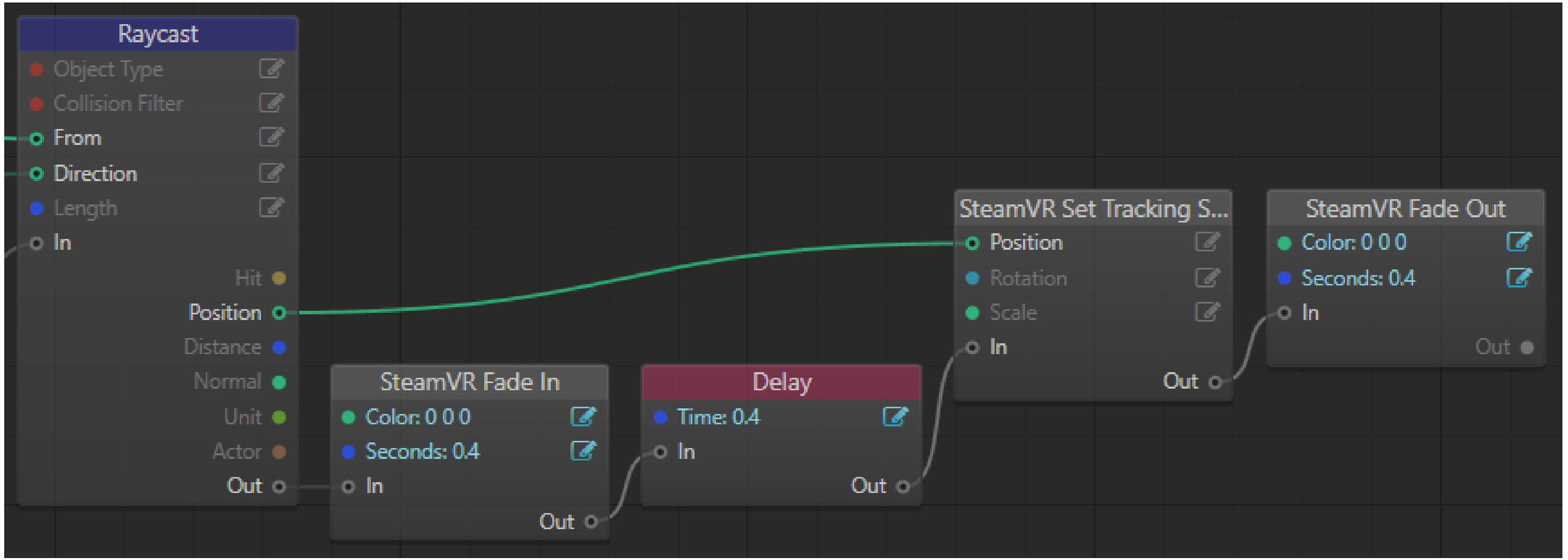
Stingray VR Templates

- Template for both Oculus and HTC Vive
- Very similar functionality exposed in both the Lua API and Flow
- Initialization in respective lua files (steam_vr.lua/oculus.lua)
- VR settings in **settings.ini**
- A lot of premade functionality present in the templates

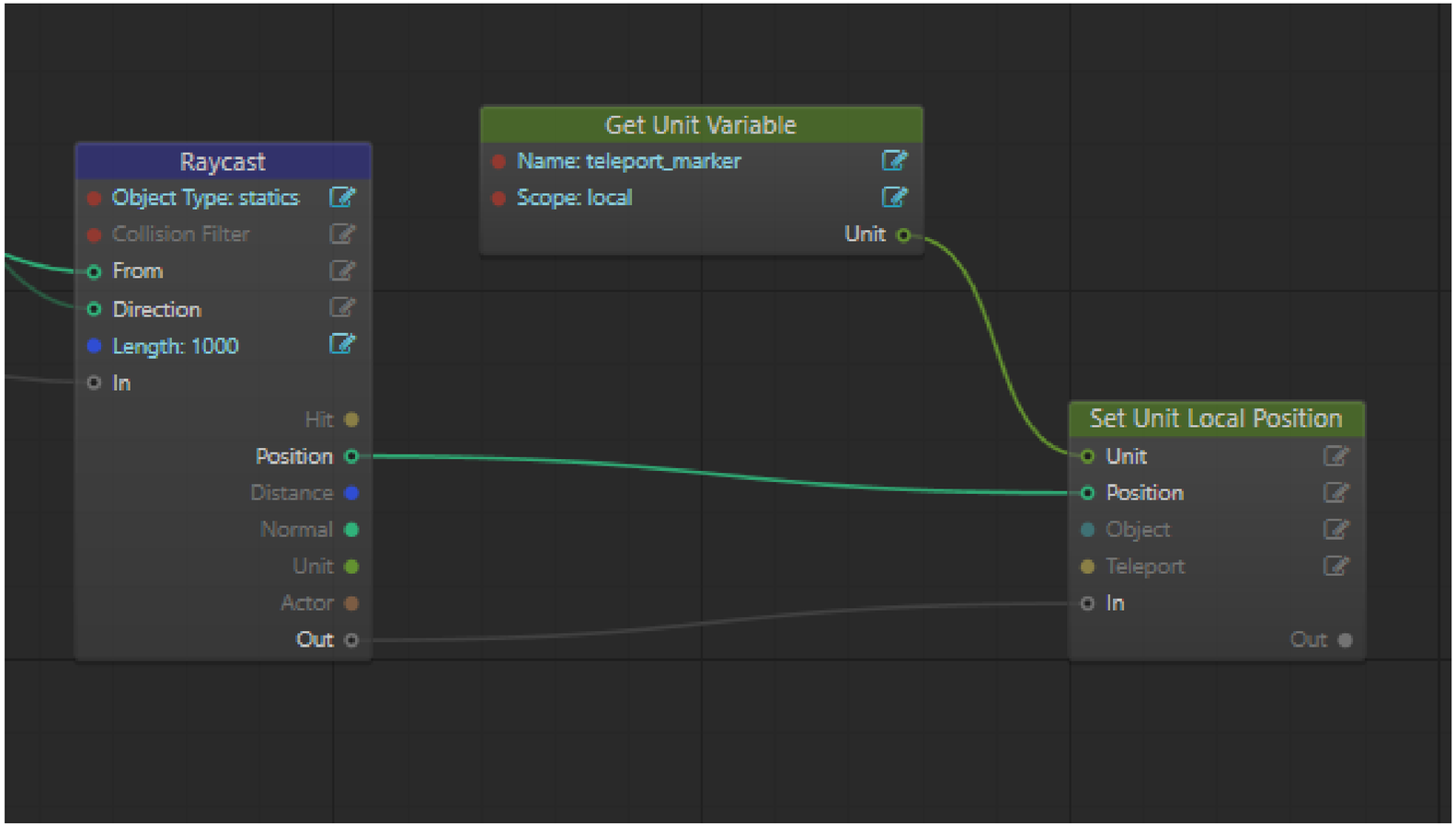

Create Unit Variable on Spawn - wand_...
Get Unit Str...
Concaten...
Unit Spaw...
Set Unit Variable
Spawn Pickup Unit to get the Pickup Functionality
Oculus Button
Get Unit Str...
Spawn Unit on Position
Set Unit String Data
Oculus Link Node To Tracker
Unit Flow...
Unit Flow...
Unit Flow...
Get Unit Bool Data
Branch
Link Controller
External In Event
Oculus Link Node To Tracker
Get Unit Bool Data
Branch
Teleport
Get Unit Str...
Spawn Unit on Position
Set Unit String Data
Oculus Link Node To Tracker
Unit Flow...
Unit Flow E...
Oculus Button
Unit Flow E...
tracking space helpers
spawn tracking space helpers
tracking helpers visibility
Draw Debug Raycast from end of arc --- plug in to see
Spawn_Teleport_Units
is laser on?
transport arc --- controller 1 --- pull trigger to show --- release to set transport area
first raycast forward from wand attach location
spawn second raycast helper unit and set its rotation and position above initial raycast loc...
do second raycast to get move location
check normal for spawn location
set new tracking space location
link ribbon and create arc
link ribbon and to raycast object
handle can teleport color change and mesh hide
corner_1 position
corner_2 position
corner_3 position
corner_4 position
Teleport UI SFX
HMD pos for offset of the tracking sp...
Wand Velocity
Set Variables on Spawn
Hide - Unhide the Controller
Set Controller Visable
Set Controller Invisible
Event for Hiding the Wand
Un-link - Release
Is the Wand Linked to Som...
Is the Wand Attached to Something
Unlink the Object and Inherit the Wand's Velocity
Trigger Interaction
Link - Proxc
Is the Wand Touching an O...
check for no pick up in unit data
Is the Actor Dynamic?
Collision Events and Variables
Attach the Object to the Wand
should the unit snap and preserve worl...
Pickup Sound Effects
Telling the Unit which Controller it Linked to the Object
Send "Linked" Event to Unit

First Example: Teleportation

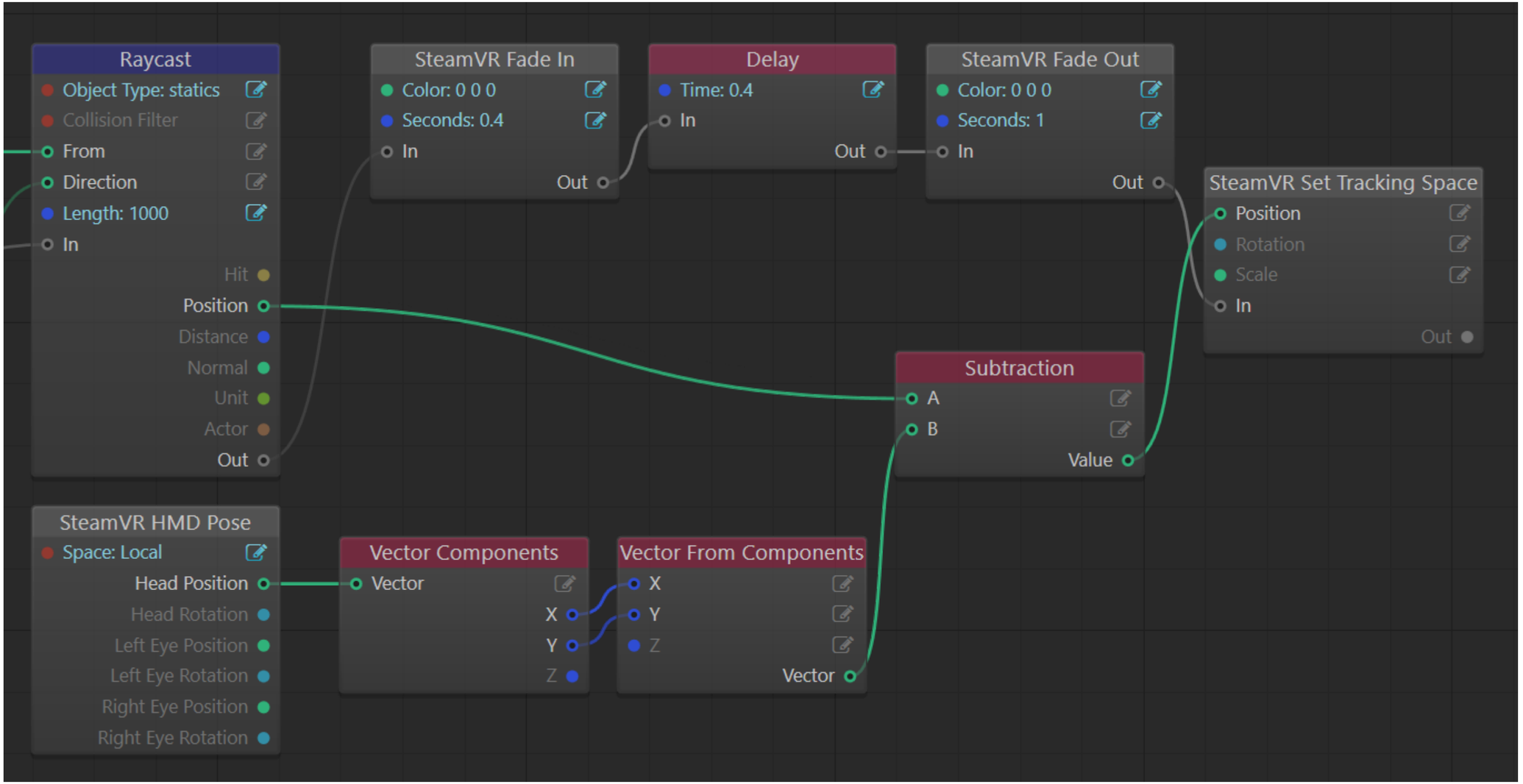
Teleporting with a fade transition…



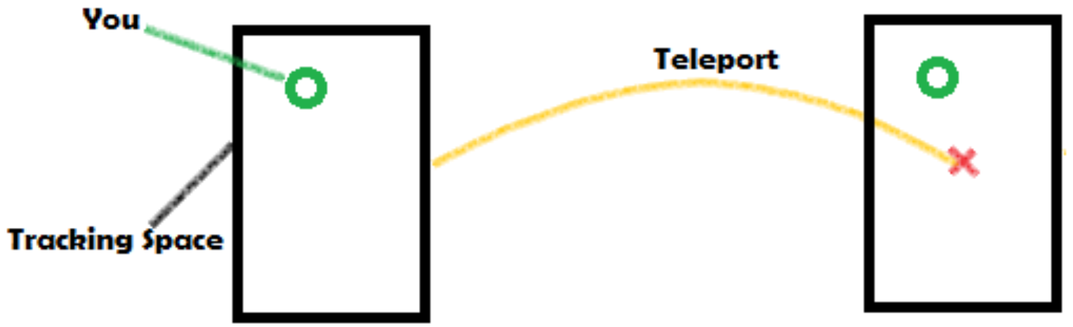
Teleportation Marker



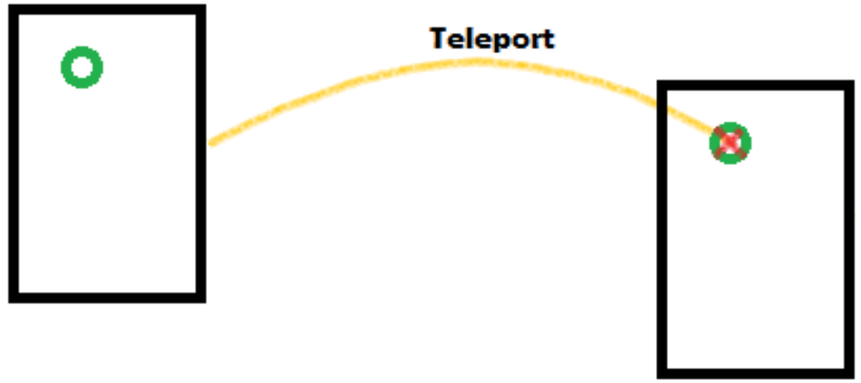
Last improvement…



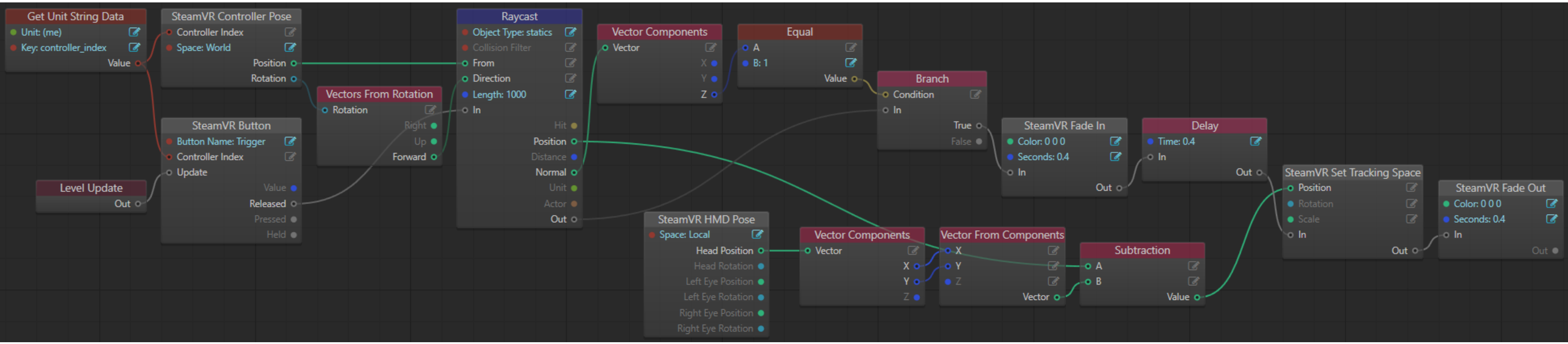
Without Local Position Adjustment



With Local Position Adjustment

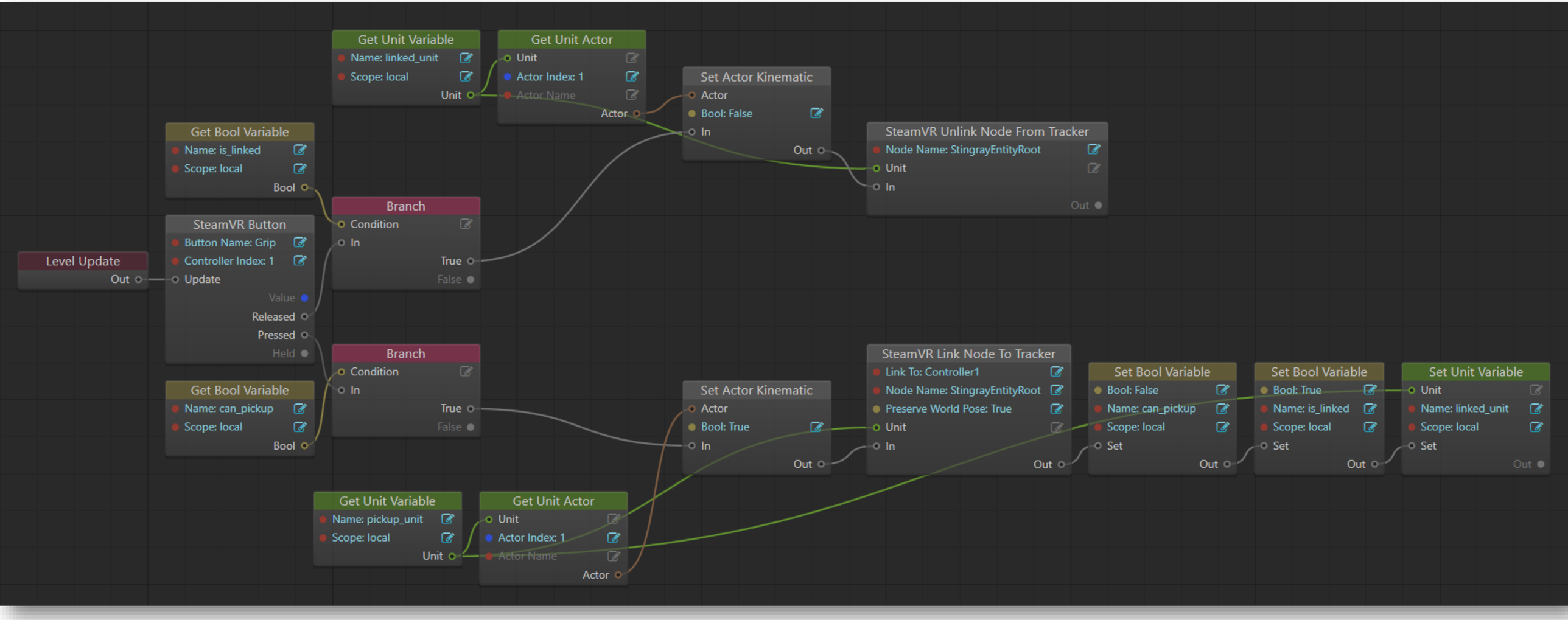


Final Result

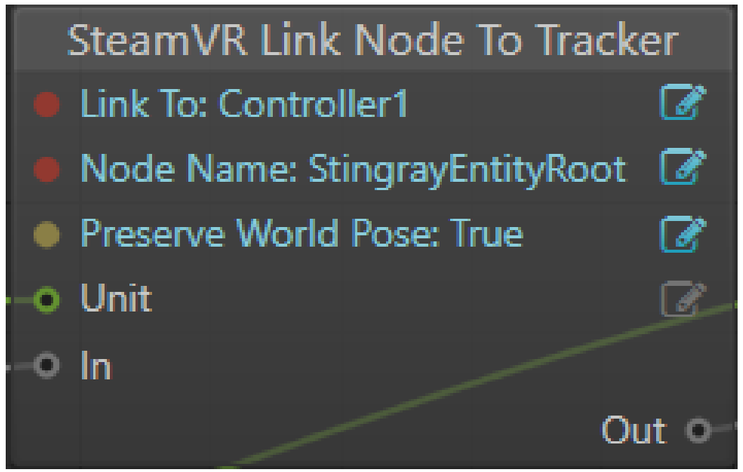


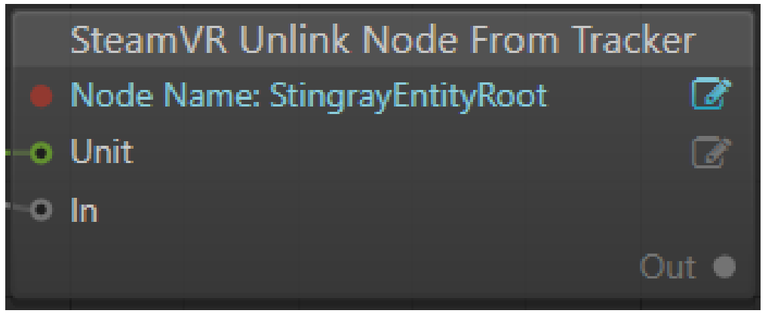
Second example: Picking up objects

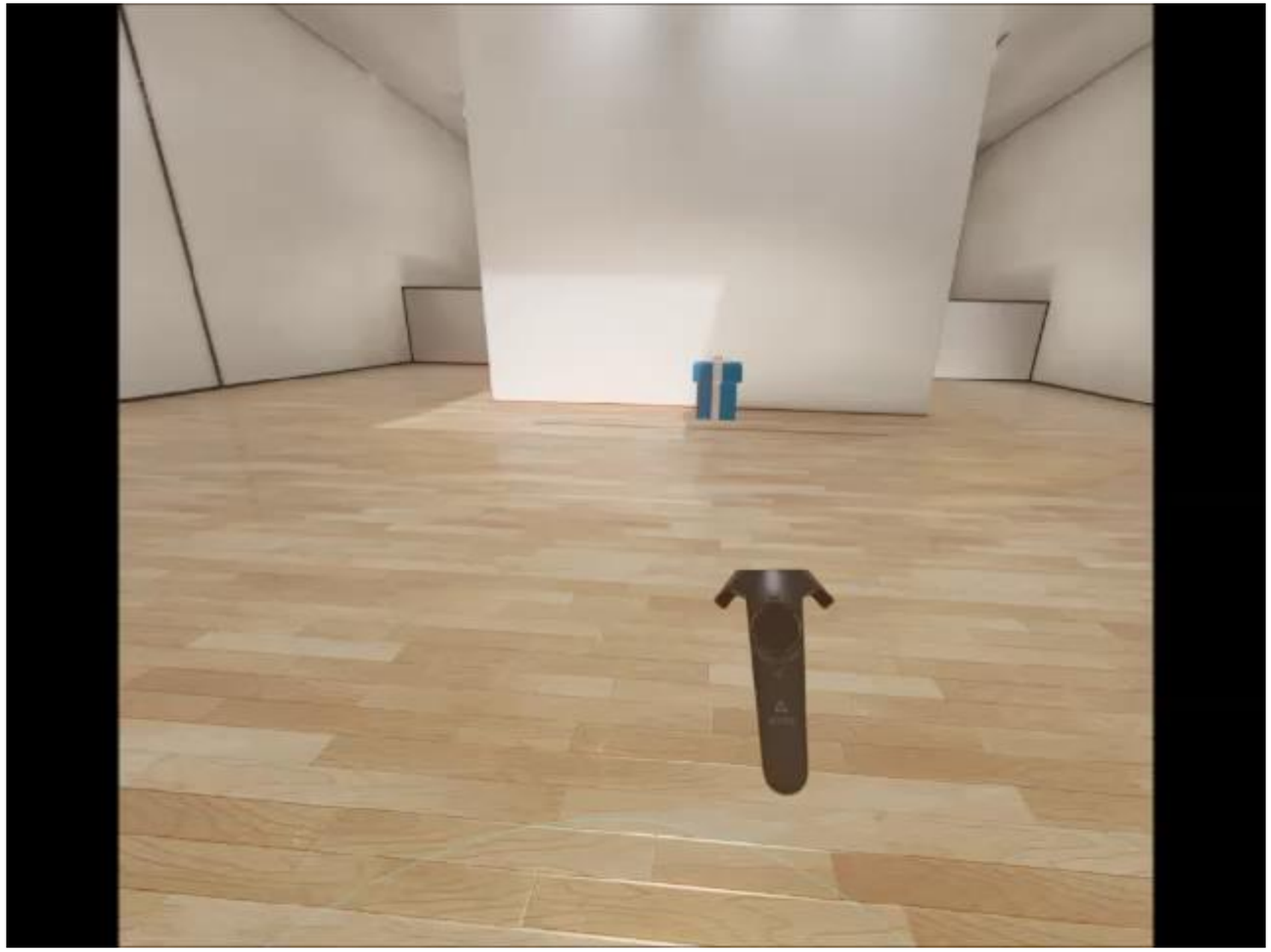
Second example: Picking up objects



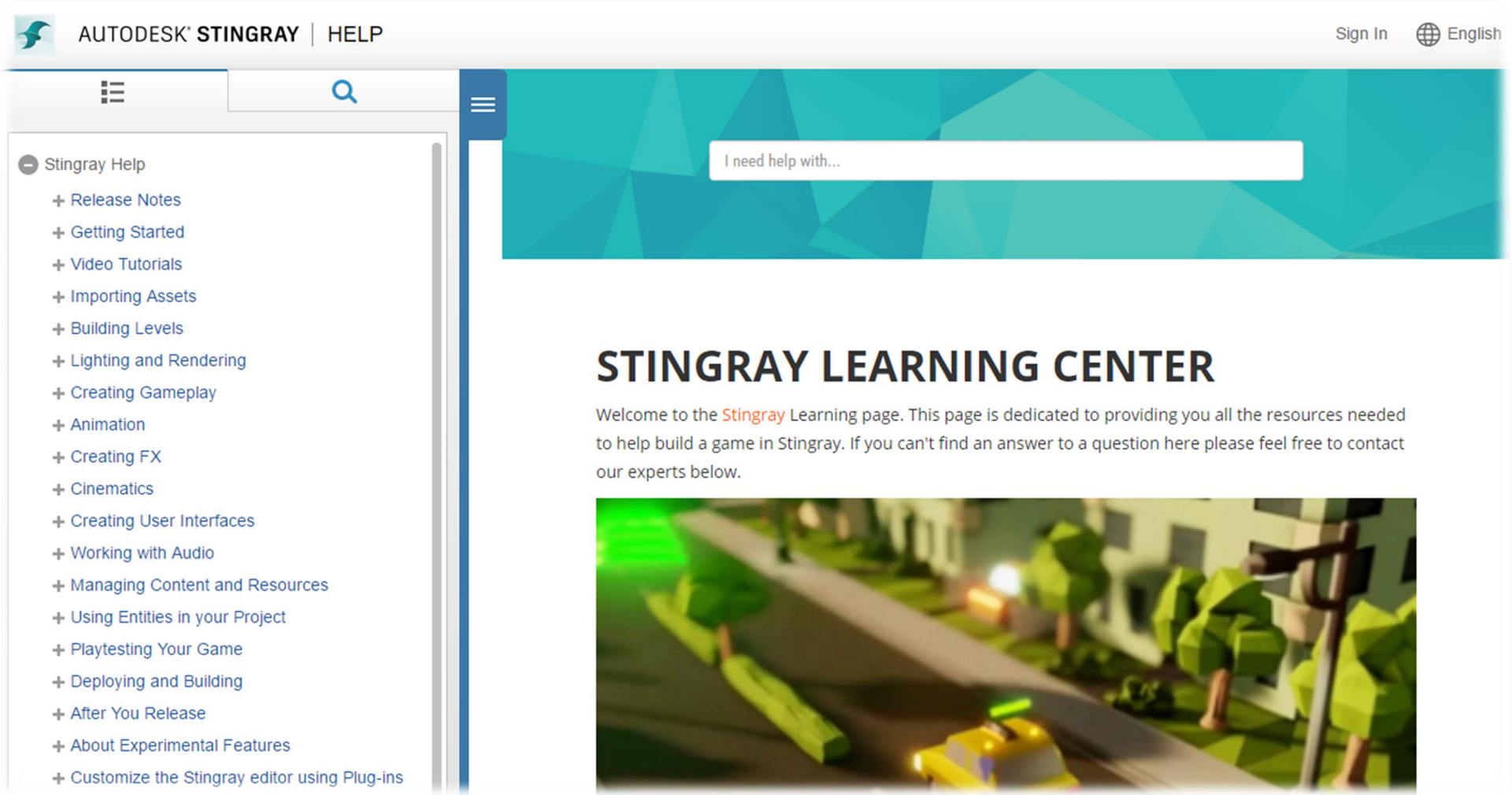
- Multi-threaded game engine (Game thread, Render thread, GPU thread, etc).
- GPU thread operates 2 frames behind Game Thread
- VR Trackers updated on Render Thread







Documentation and Online Resources



Optimizing Content For VR in Stingray

Don’t make me sick


AUTODESK
STINGRAY

Typical Optimization Checklist

- Is overdraw set too high?
- Am I running too many post-processes?
- Am I rendering too many polygons?
- Am I running out of GPU texture memory?
- Am I casting too many shadows? (batches)

Render Settings – overdraw and anti-aliasing

- Reduce overdraw to 1.4
  - Templates are 1.6 by default

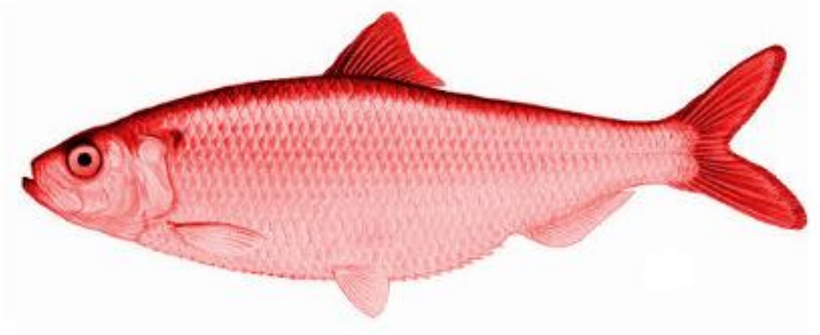
- Disable TAA
- Enable FXAA

Post Processes

- Disable all un-necessary post processes
- Keep:
  - Bloom
  - Auto-Exposure

Poly Reduction

- Red Herring



Poly Reduction Tools

- 3DS Max:
  - Optimize, ProOptimizer
- Maya
  - PolyReduce

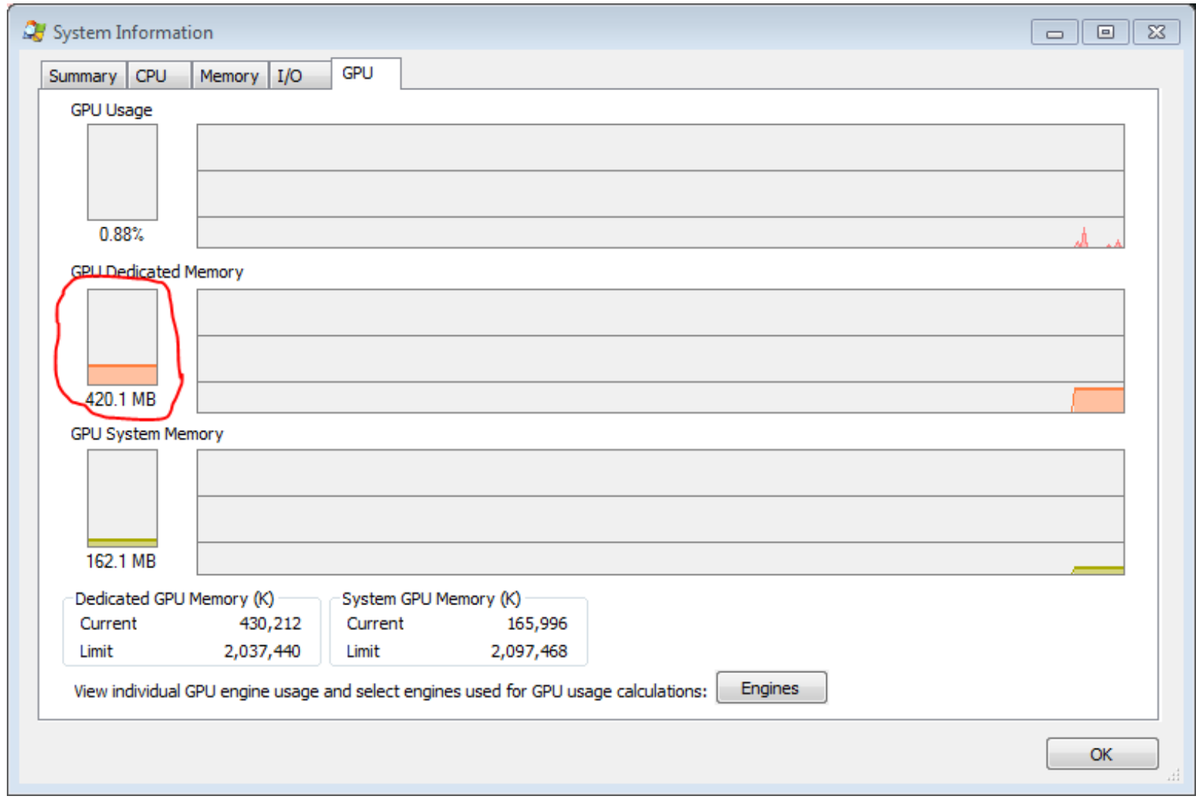
- Simplygon
- Instant Field Aligned Meshes:
  - https://github.com/wjakob/instant-meshes

Texture Blowout

Optimization Checklist: Textures

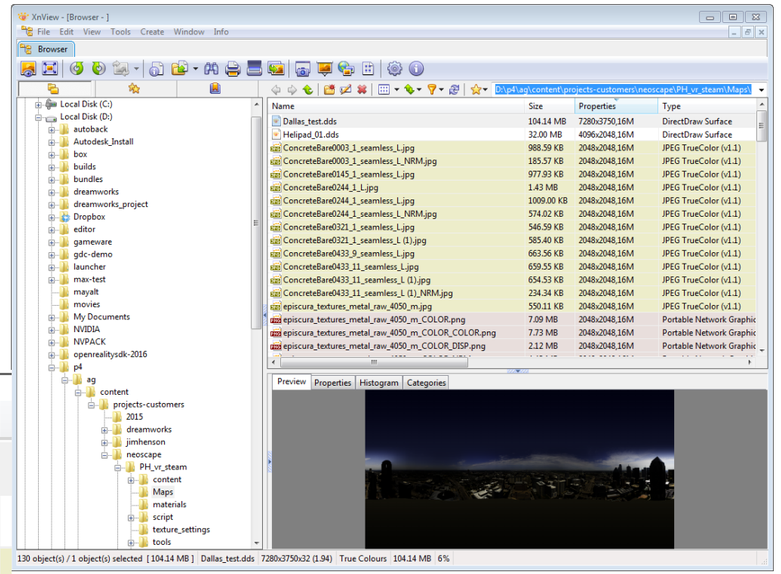
- Am I maxing out my texture memory?

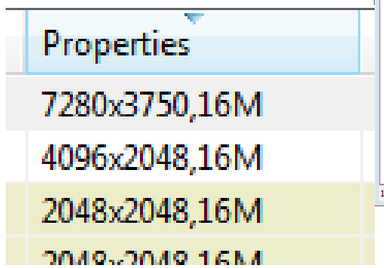
- Tool: Procexp
  - GPU Tab
  - GPU Dedicated Memory



Texture Optimization Basics

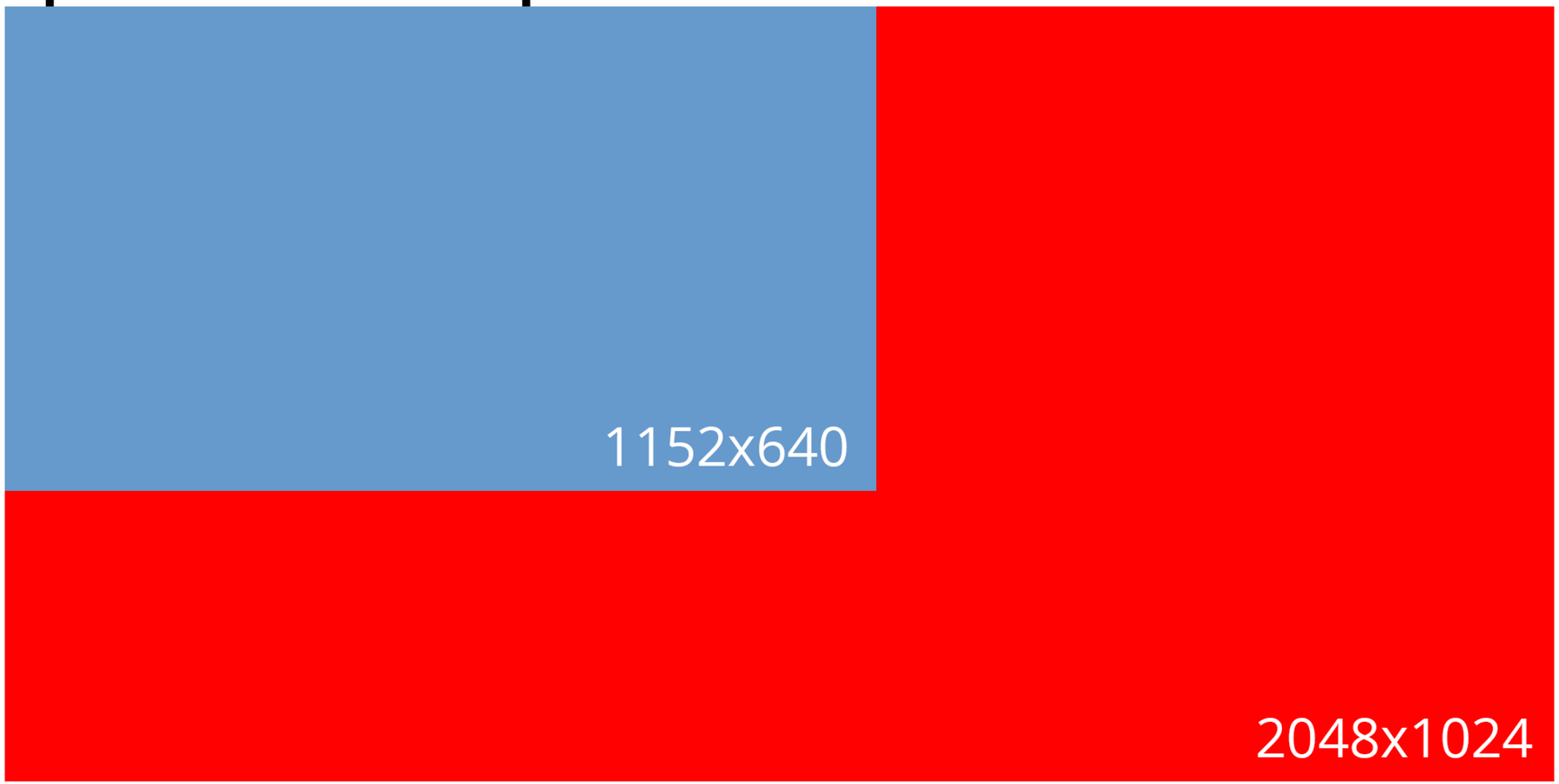
- Resize all textures to be powers-of-two and of reasonable size
  - 1024x1024, 256x512, etc
- Tool:
  - XnView
    - Show All Files (recursive)
    - Sort by Properties
    - Batch processing





Why power of two?

- Compression requirements

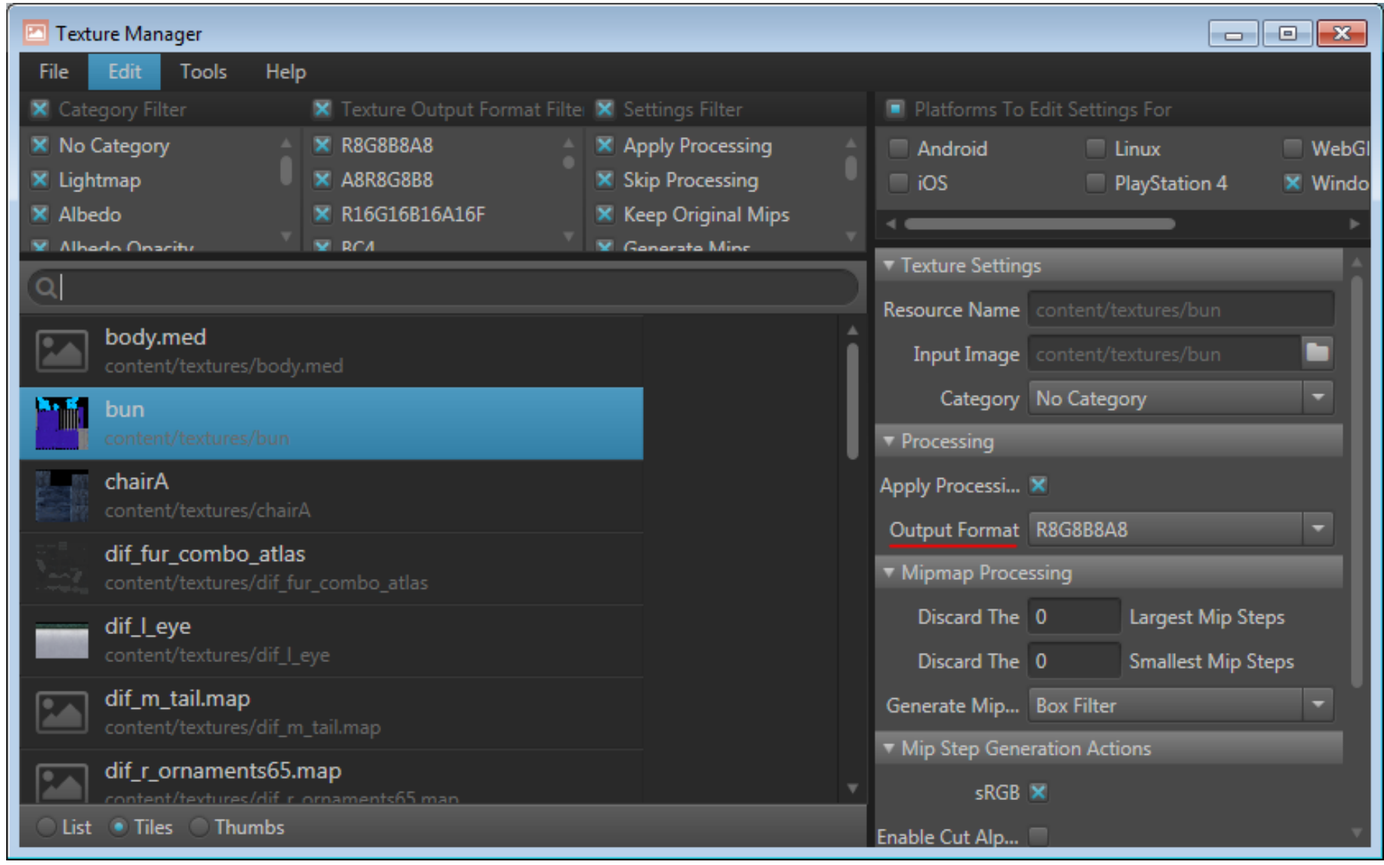


Texture Optimization Basics (for PC/Consoles)

- Enable compression on all textures:
  - Color(albedo) no alpha: DXT1
  - Color with alpha: DXT5
  - Normal: BC5
  - RMA: DXT1
  - Single-channel linear textures: BC4

Texture Optimization Basics

- Tool: Texture Manager
  - Output Format
  - Discard Largest MIPS



Optimization Checklist: Shadow Casting

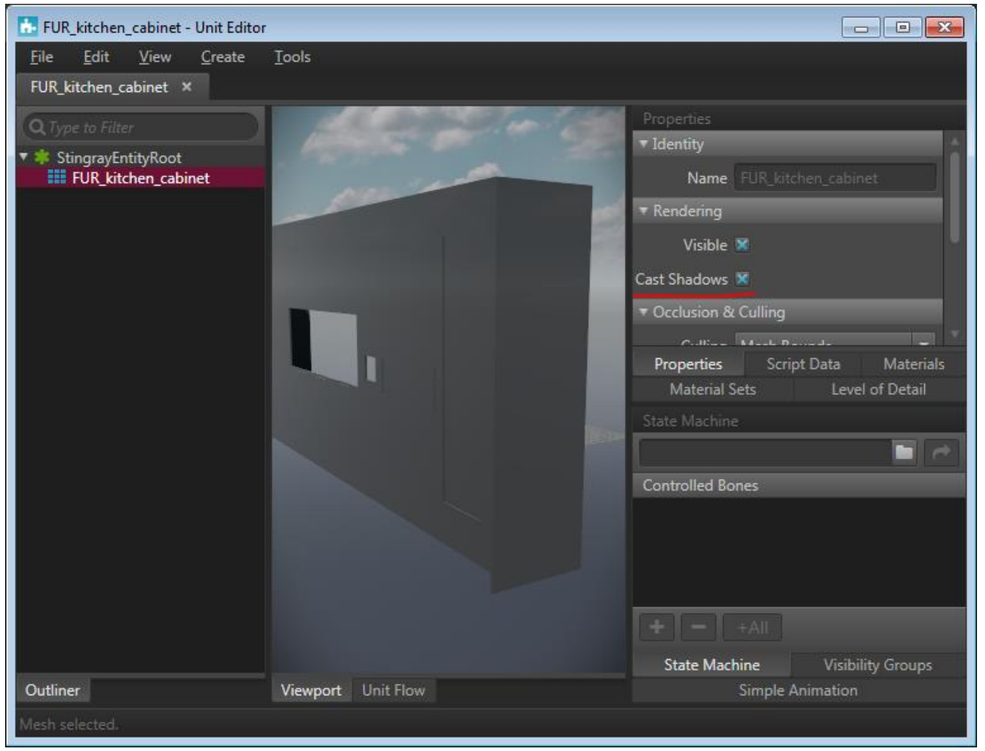
- Am I casting too many shadows?
- Tool: Artist Performance HUD

Optimization Checklist: Shadow Casting

- Disable shadow casting where unnecessary
  - Objects in shadow, floors, etc
- Convert point lights to Spotlights
  - 1/6 cost of point lights
- Bake Lighting

Optimization Checklist: Shadow Casting

- Shadow Proxies
- Tool: Unit Editor
  - Cast Shadows



Optimization Checklist: Batching

Optimization Checklist: Batching

- Tool: Artist Performance HUD

Performance Overview	
FPS	5.061
Smoothed FPS	5.000
Performance HUD overhead	
CPU	0.235ms
GPU	0.145ms
Frame Counters	
Batch Merging	0 -> 0
Instance buffer size	0.00 kB
Batches	93
Primitives	15258
Texture Switches	253
State Block Switches	60
Vertex Shader Switches	37
Pixel Shader Switches	67
Constant Buffer Binds	213
Constant Buffer Updates	164

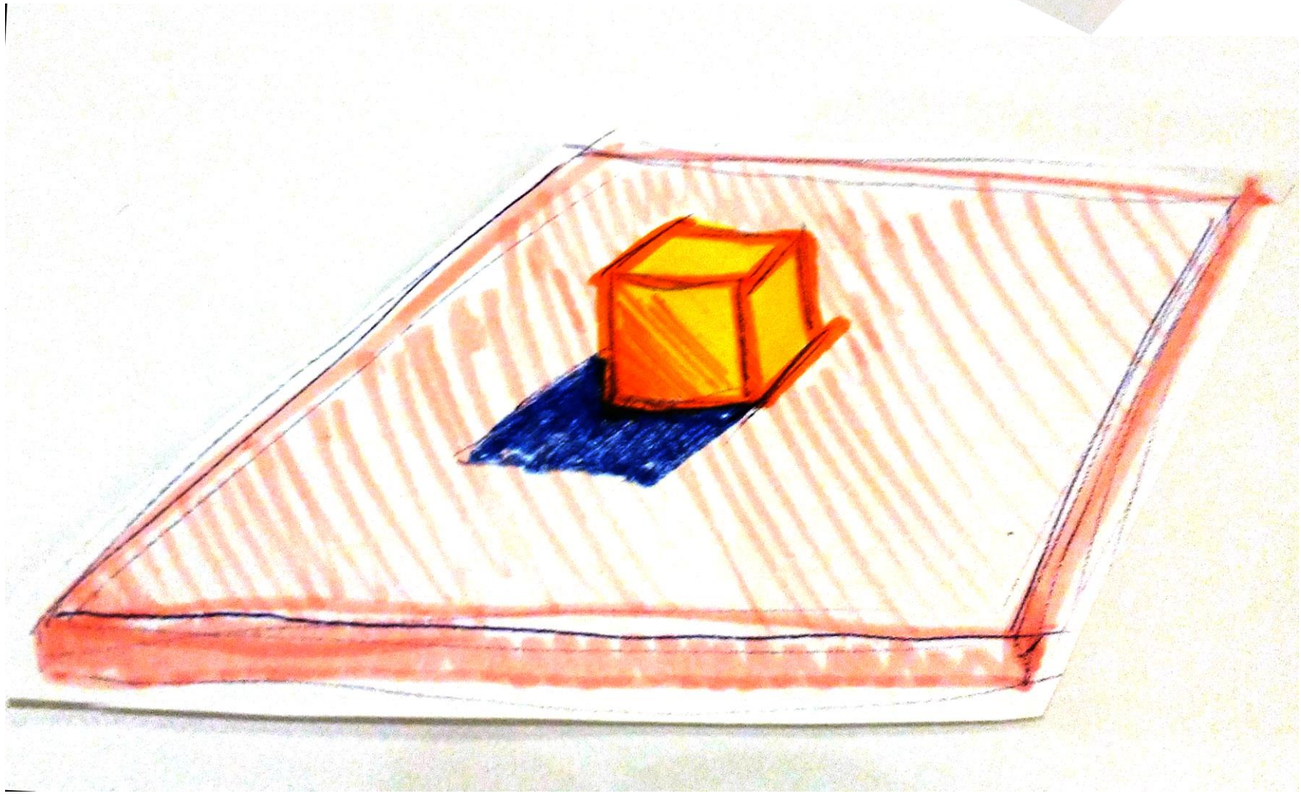
Shadow Casters	Batches: 9	Primitives: 3452
Clears	Batches: 0	Primitives: 0
G-Buffer	Batches: 4	Primitives: 4532
Forward Opaque	Batches: 0	Primitives: 0
Opaque Lighting	Batches: 10	Primitives: 72
Reflection Probes	Batches: 1	Primitives: 12
Emissive	Batches: 0	Primitives: 0
Skydome	Batches: 1	Primitives: 4416
Transparency	Batches: 9	Primitives: 2538
Post Processing	Batches: 57	Primitives: 228
- SSR	Batches: 28	Primitives: 112
- SSAO	Batches: 6	Primitives: 24
- TAA	Batches: 1	Primitives: 4
Scaleform Studio	Batches: 0	Primitives: 0
Present	Batches: 0	Primitives: 0

What Is a Batch?

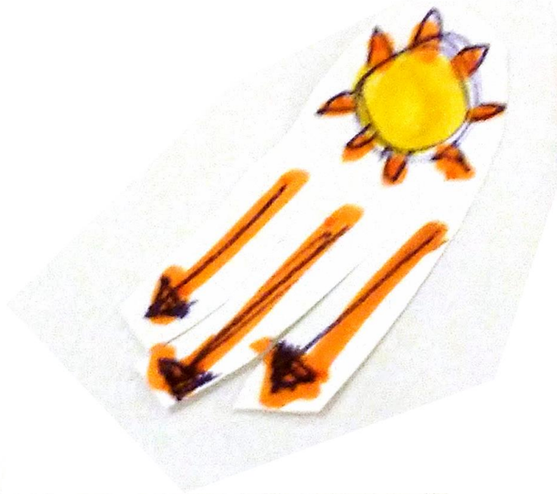
- One batch per material, per mesh, per shadow casting observer, per camera

- One Cube
- One Plane
- One Spotlight

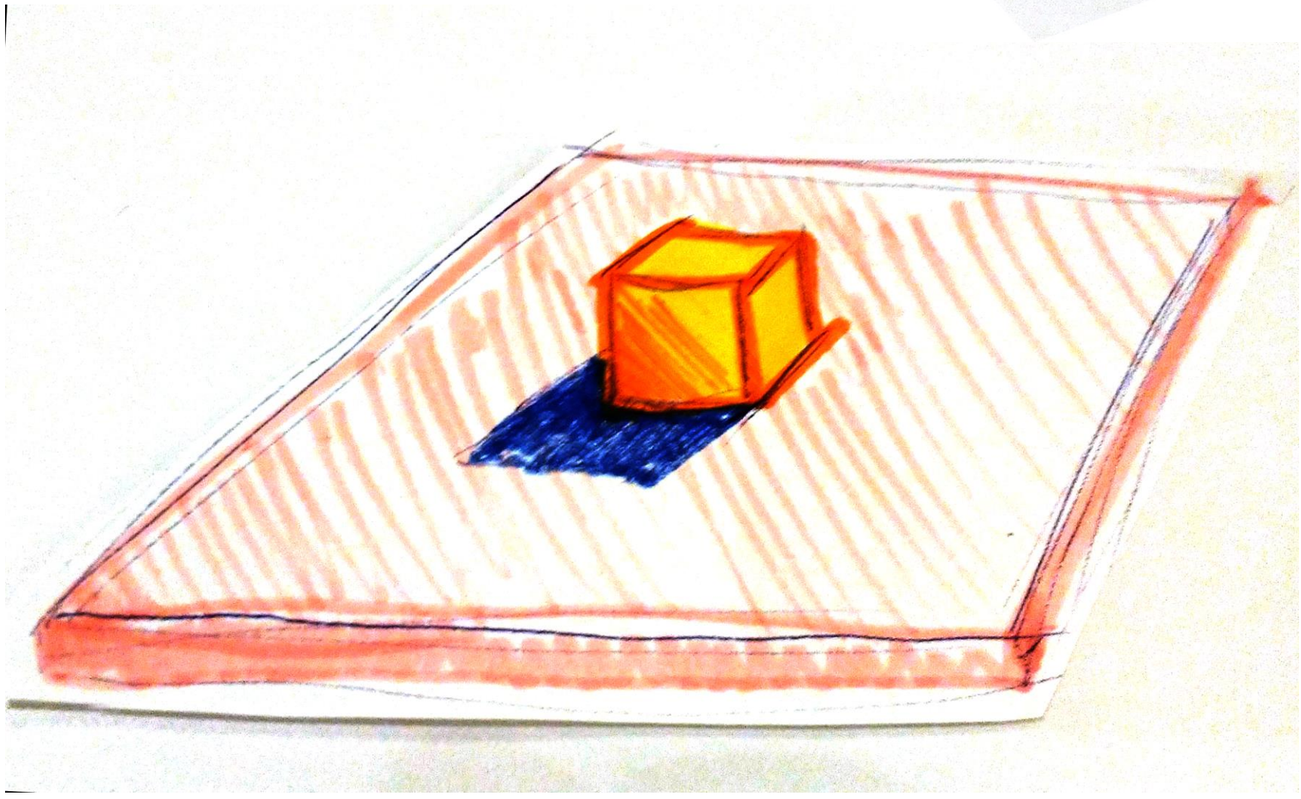
- 2 Shadow Casting Batches
- 2 G-Buffer Batches



- One Cube
- One Plane
- One Sunlight

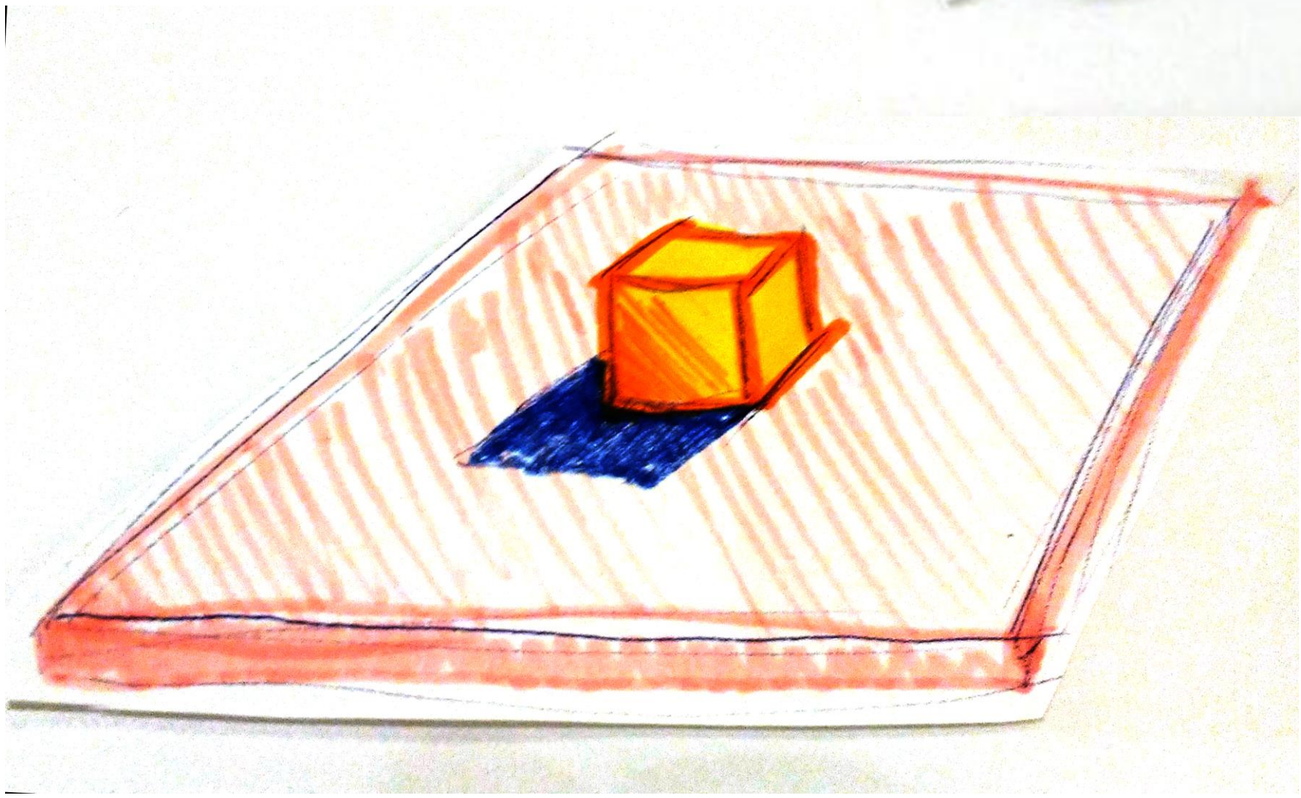


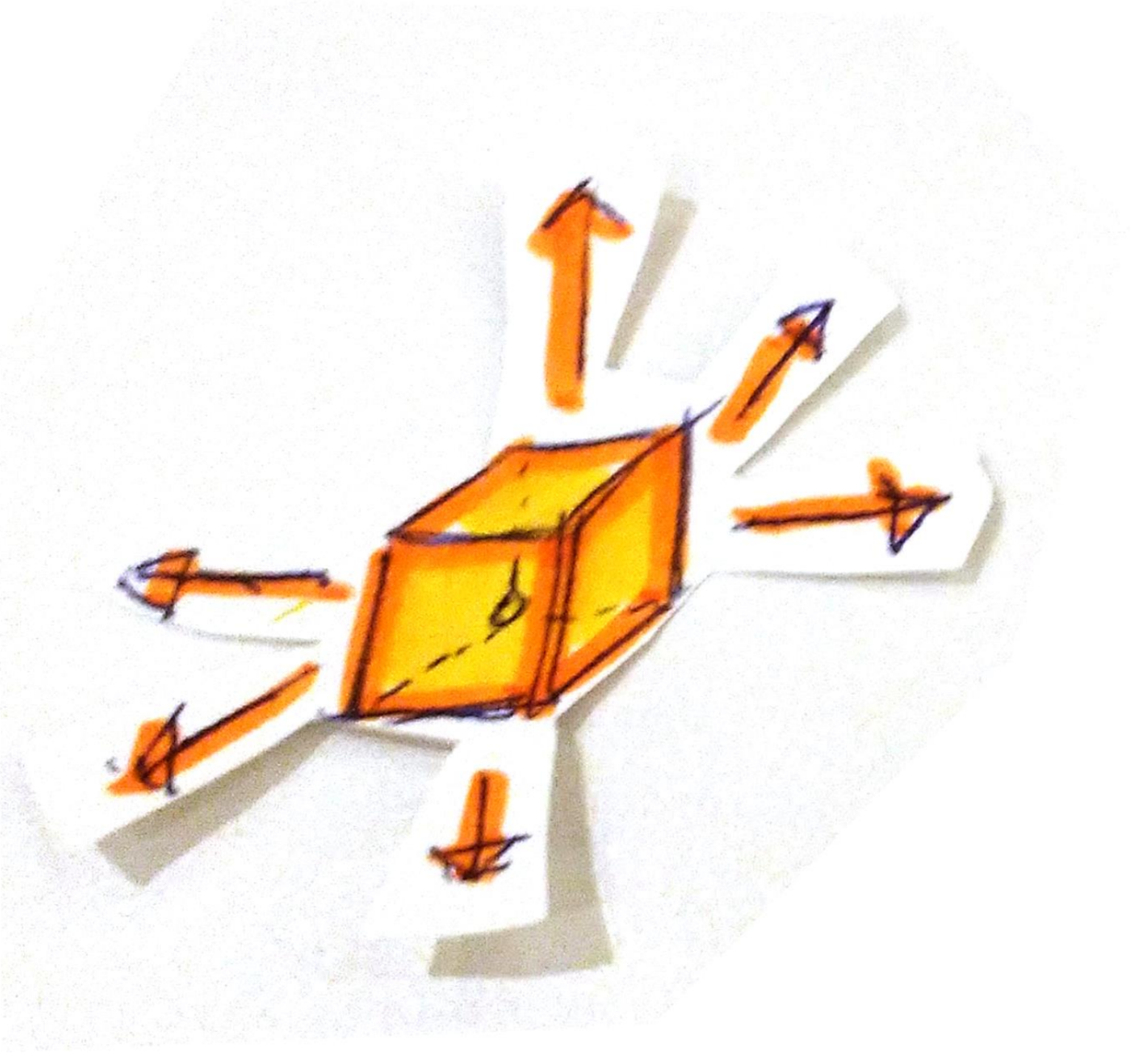
- 5 Shadow Casting Batches
- 2 G-Buffer Batches



- One Cube
- One Plane
- One Omni Light

- 6 Shadow Casting Batches
- 2 G-Buffer Batches



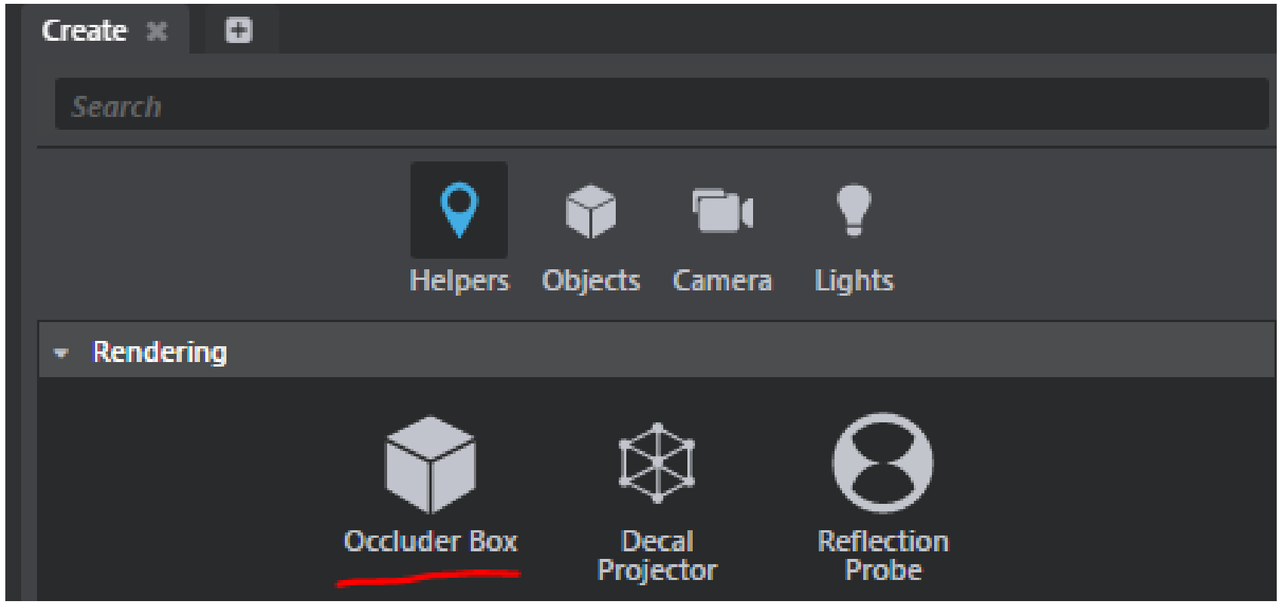


Optimization Checklist: Reduce Batching

- Merge meshes
  - Re-import
- Reduce Materials
- Create LODs

Optimization Checklist: Occluders

- Create Occluder Boxes



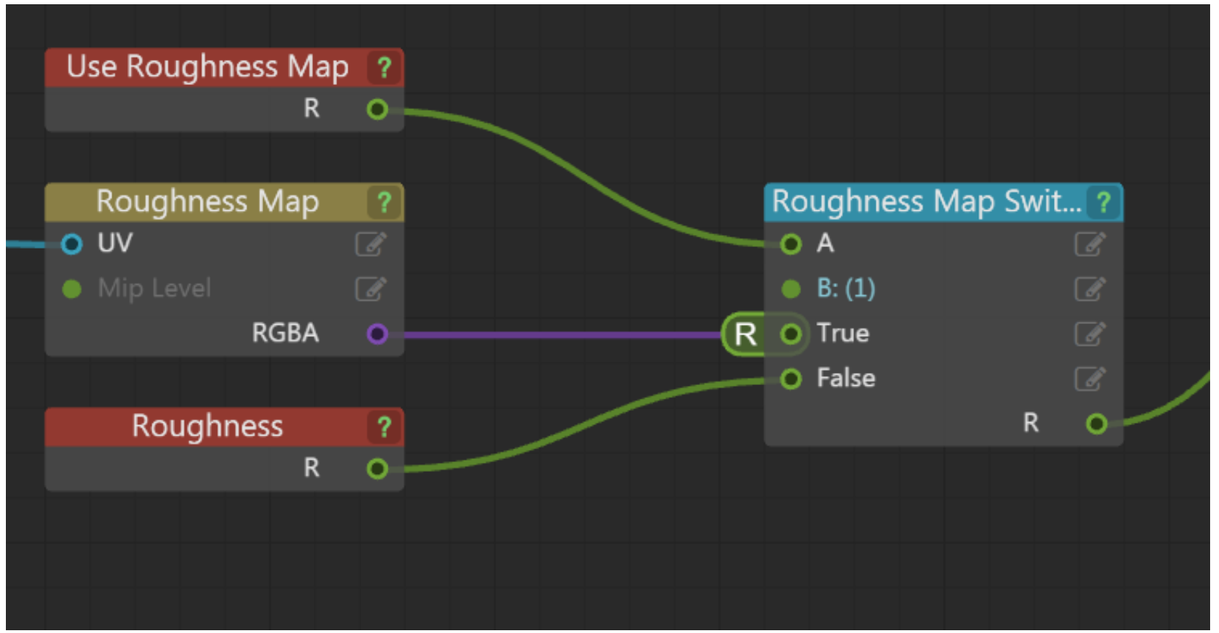
- Remember: Occluders occlude from all observers!

Batch Merging

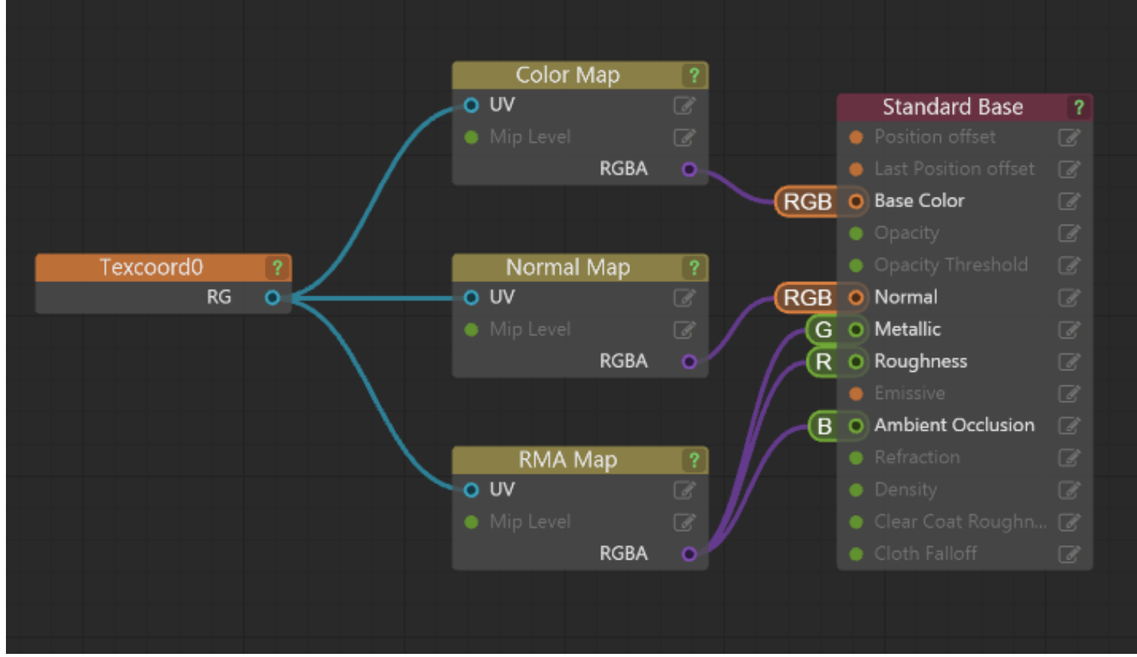
- Same Geometry
- Same Materials
- Material supports Instancing

Material Optimization

- Standard import materials are optimized for compatibility – not performance!



Material Optimization



- Create simple materials
- Use inheritance
  - Parent materials
- Use standard_rma materials
  - RMA = Roughness, Metalness, Ambient Occlusion
  - Included in v1.6

In Closing

- You can’t always get what you want
  - But if you try sometimes…
- Game artists can help!

Acknowledgements

- Tobias Persson
- Niklas Frykholm
- Mikael Hansson
- Andreas Asplund
- Jim Sagevid
- Jean-Philippe Grenier
- Anders Lindqvist
- Rex Hill
- Dan Matlack
- Paul Kind
- Matthew Harwood