

Scriptable Render Pipelines in Unity

GameDev Meetup #18
Kaunas 2018



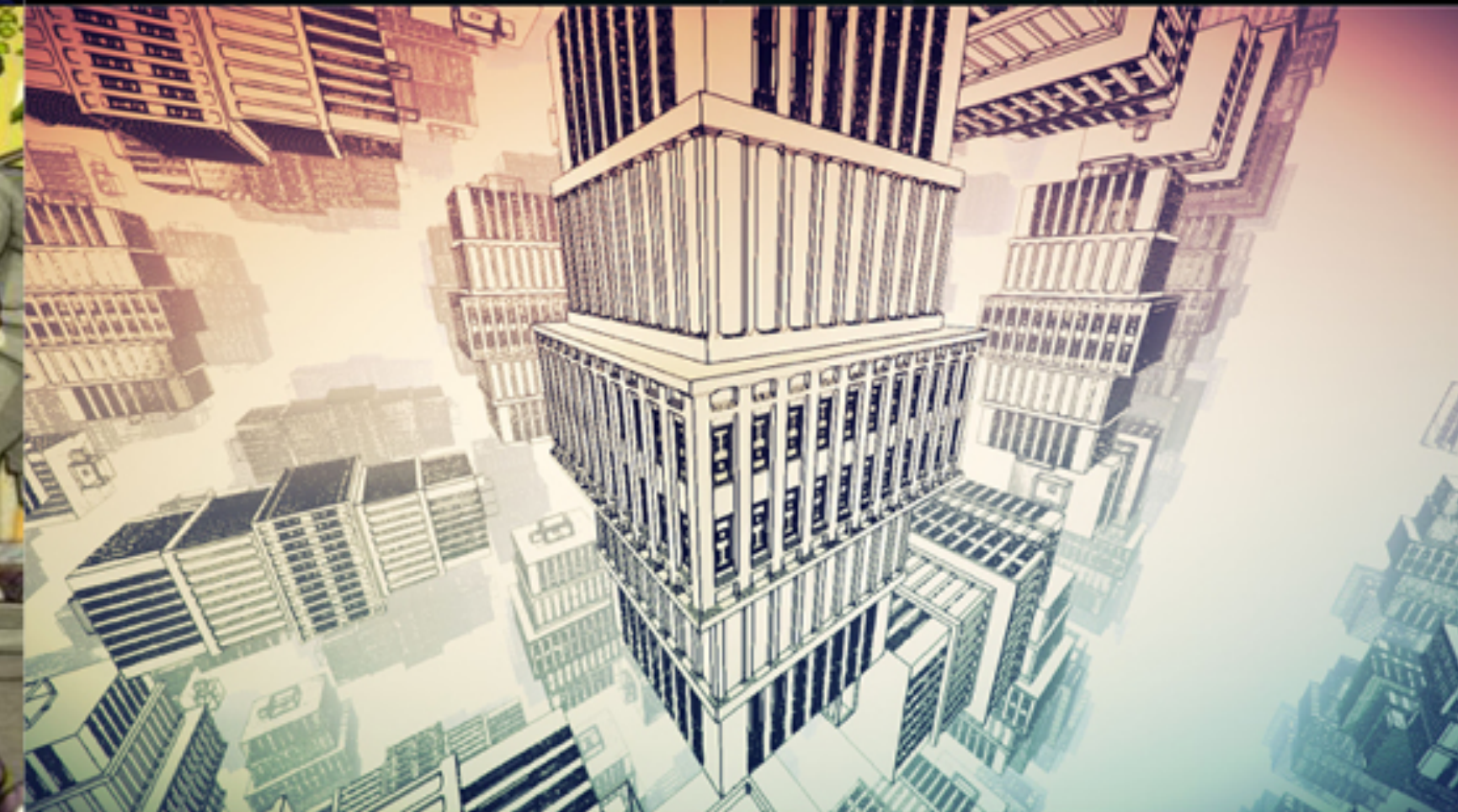
Aras Pranckevičius

- Internal build systems engineer
 - What does that have to do with graphics?
 - Nothing! ...however
- At Unity since 2006
- Been doing graphics until 2017
 - Still remember a thing or two... maybe

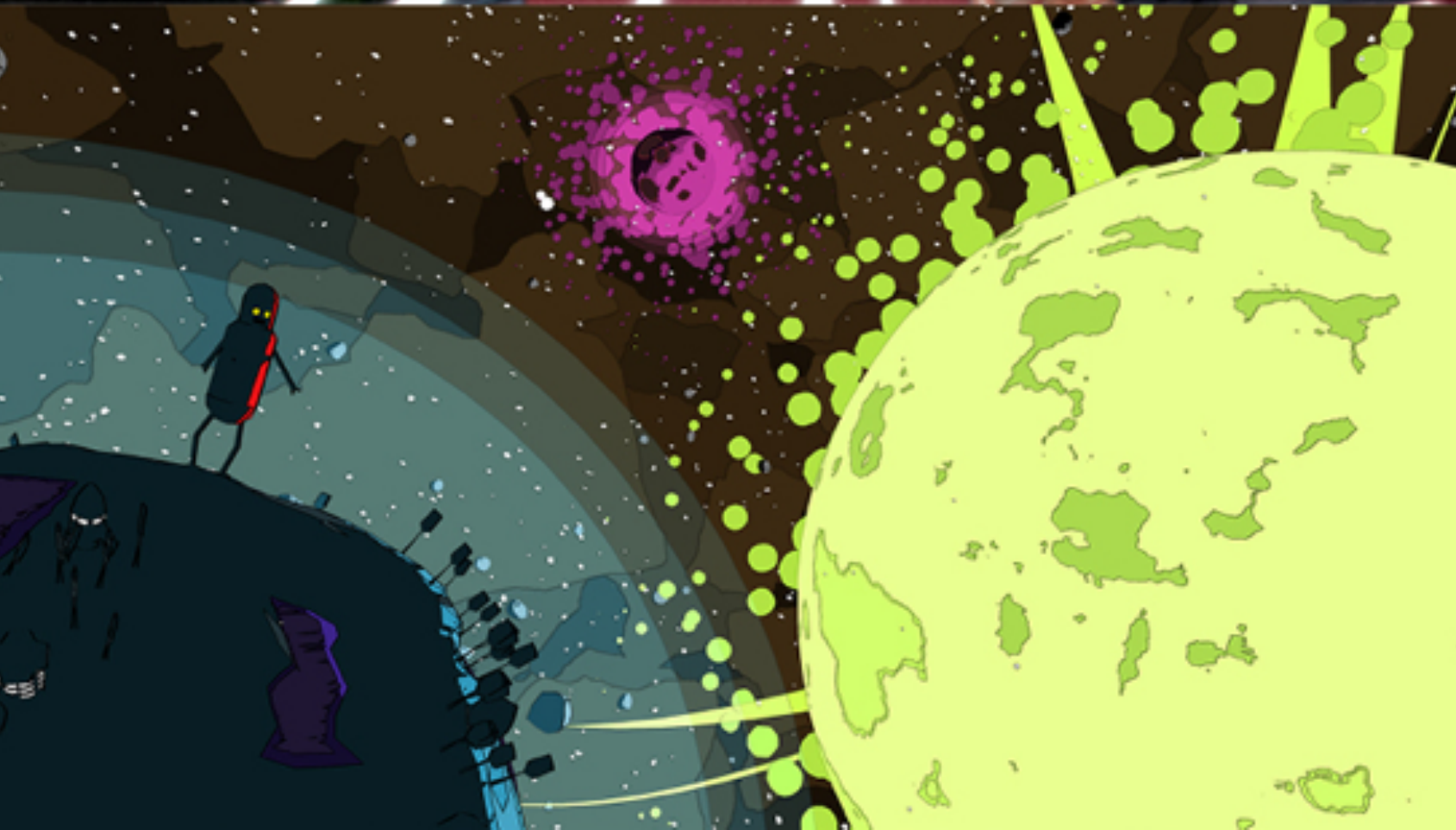
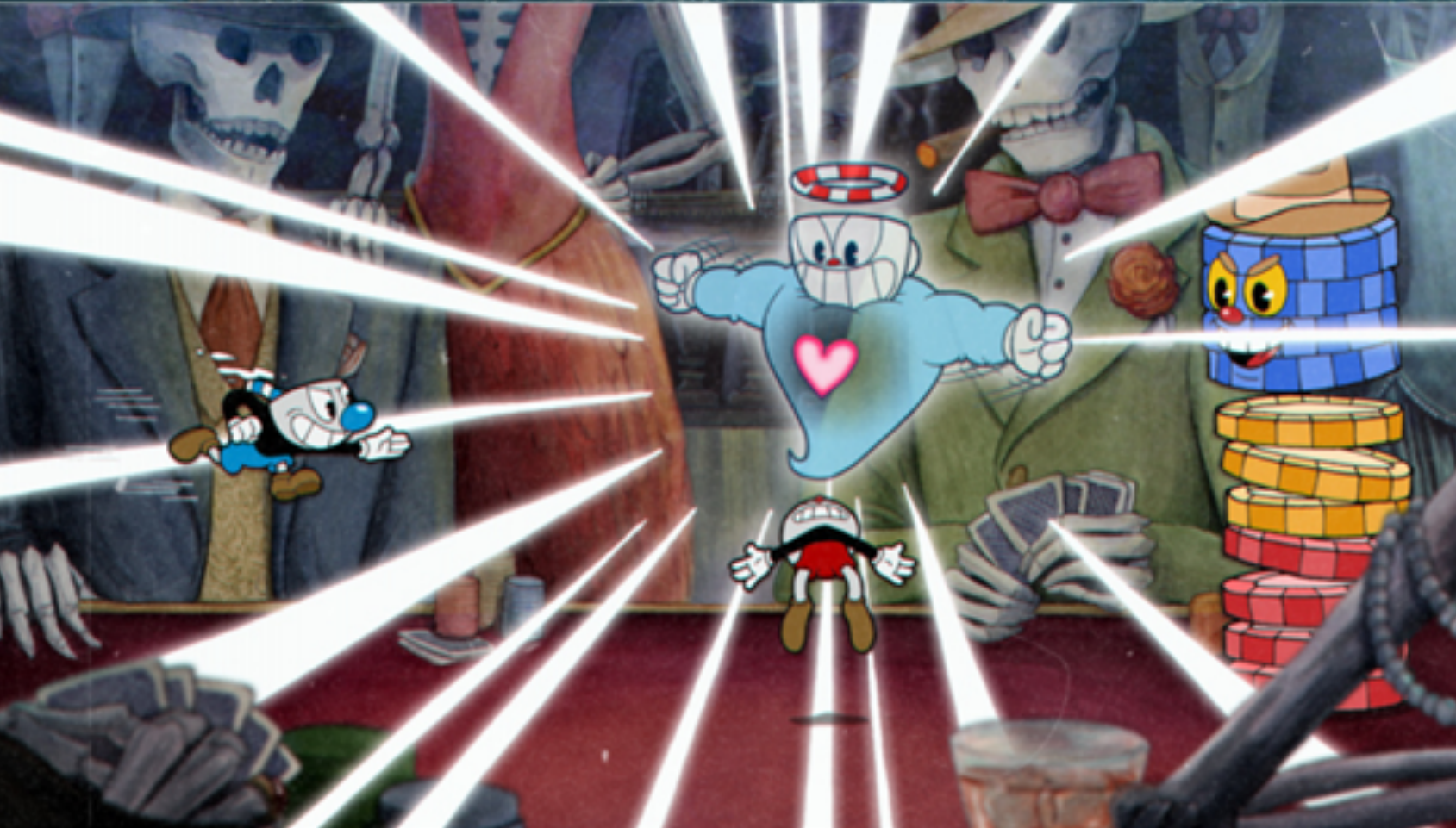
Traditional render pipeline in Unity

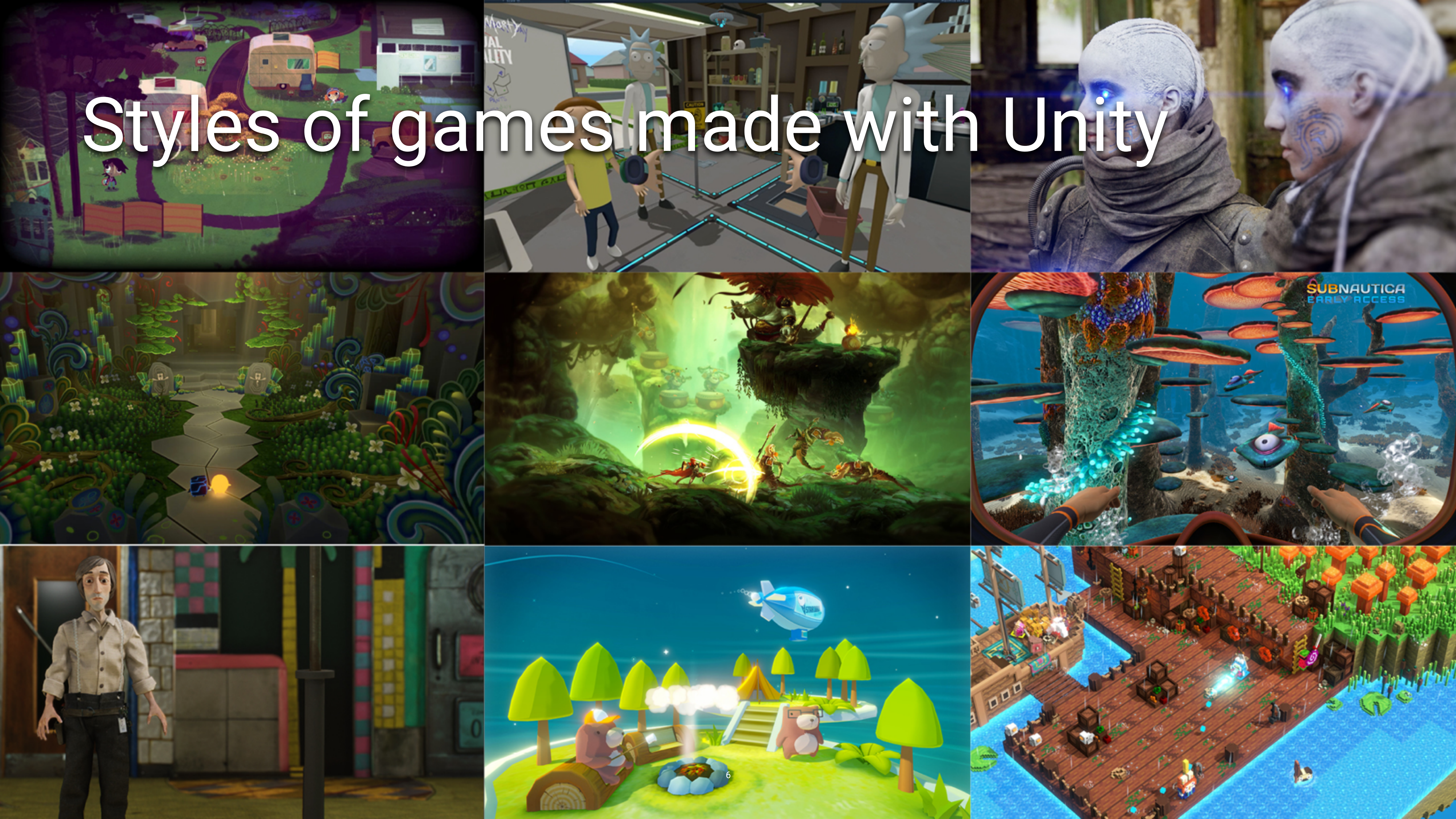
- Forward or Deferred
- A whole bunch of options & knobs
- Shaders mostly customizable
- Render pipeline itself less so
- Black box, complex, fragile
- Still enables quite different games, so that's good :)

Styles of games made with Unity



Styles of games made with Unity





Styles of games made with Unity

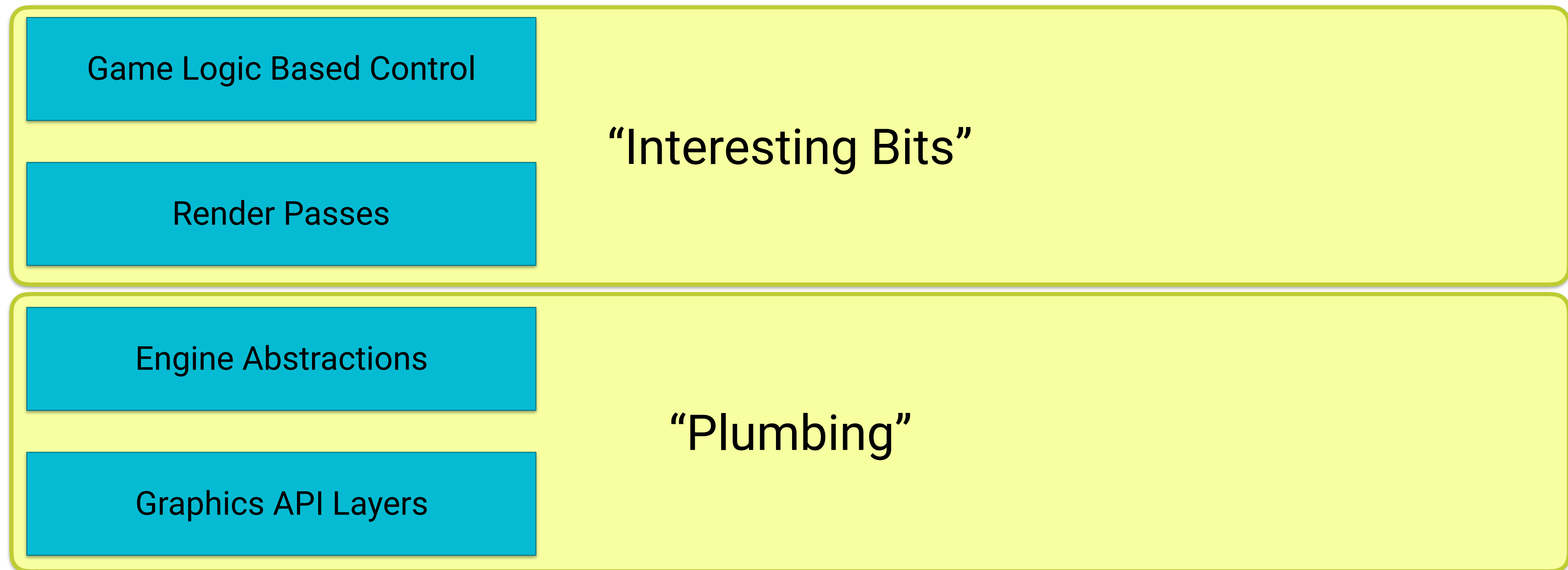
Styles of games made with Unity



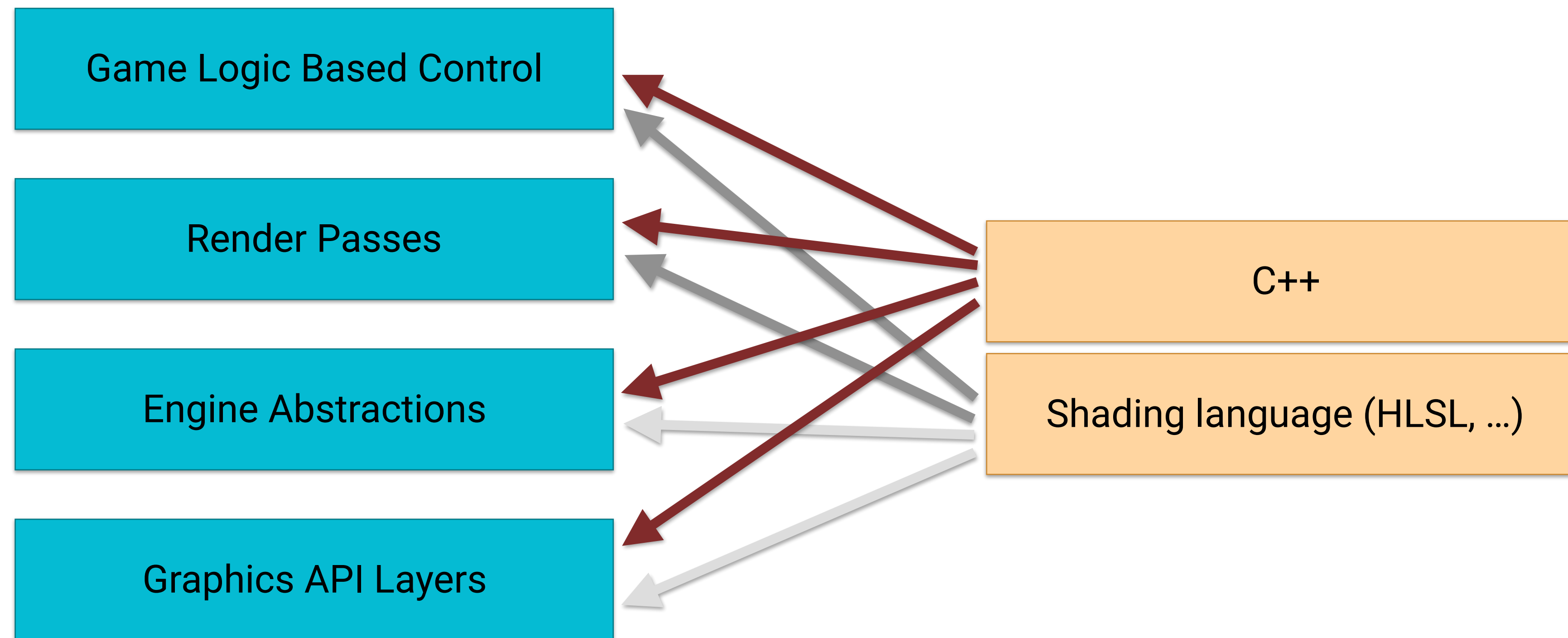
Problem

Hard to serve all of them with one
render pipeline

Typical Graphics Engine Pipeline



Typical Graphics Engine Pipeline

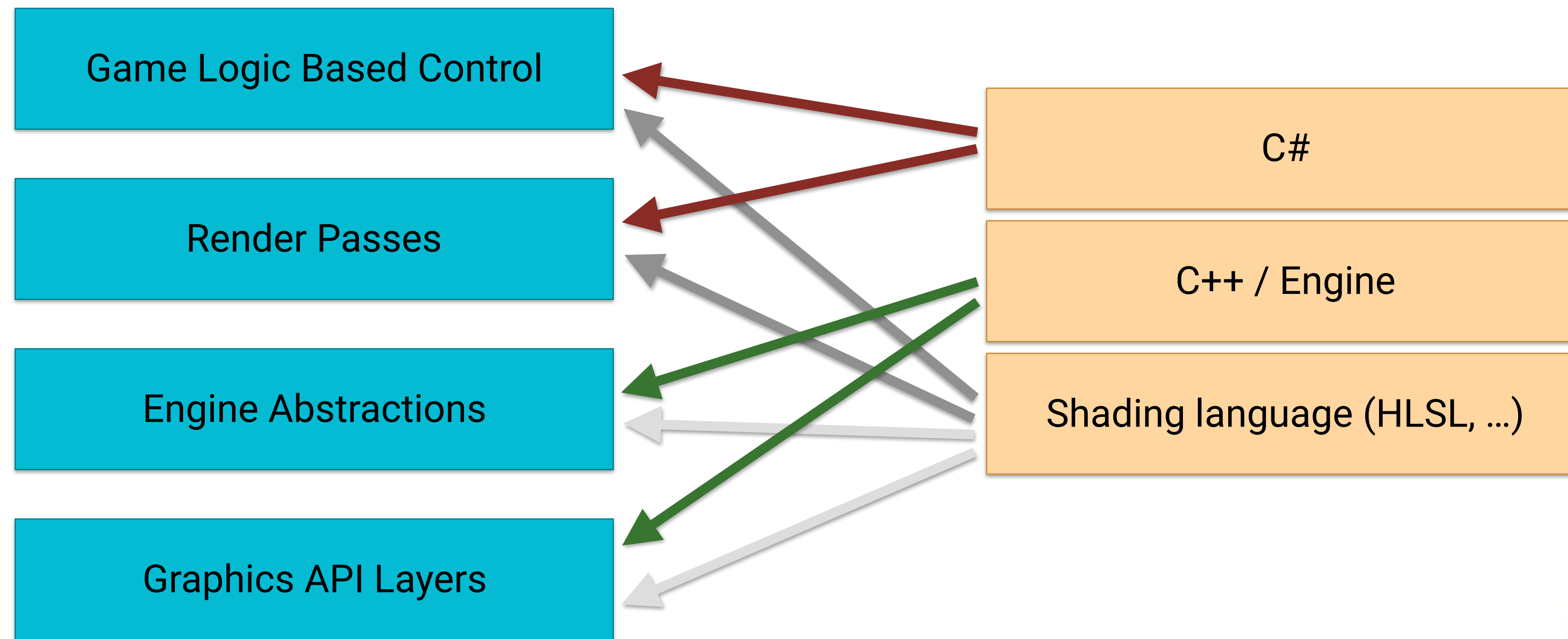


Our wishes

- Source code available
- Less black box & special cased
- More modern
- Performant



Scriptable Render Pipelines! (SRP)



SRP Concept

- What to render: culling/filtering. World -> sets of objects
- Render: draw sets of objects with some flags/params
- Setup render passes around all that
- Setup per-frame/renderpass data

<https://blogs.unity3d.com/2018/01/31/srp-overview/>

SRP High/Low Level Split

- Perf-critical things (culling, drawing sets of objects, ...): C++
 - Might move to C#/Burst* at some point
- Control/logic, render pass setup: C#
- GPU code (shaders, compute): HLSL
 - Maybe subset of C# at some point?

* Unity Burst Compiler: LLVM-based compiler for a high performance subset of C#
<https://unity3d.com/unity/features/job-system-ECS>

SRP Advantages

- Quick iteration of new algorithms
- All benefits of Unity engine/tooling
- Focus on algorithm, not busywork/plumbing
- Hot reload of C#/shader code

SRP Disadvantages (*today*)

- Needs new sets of shaders
 - Use ShaderGraph
 - Or write in HLSL
- Documentation is... a bit lacking :)
- If something needs native code tweaks/additions, it needs a new Unity release
- Not all the latest graphics features are exposed by Unity yet
 - Raytracing, conservative raster, bindless, ...
 - We're trying to catch up though

Built-in SRP: Lightweight

- Simpler
- Runs on all platforms*
- Optimized for mobile / VR
- Single pass forward renderer



* At the very moment does not work on WebGL yet due to lack of threads/jobs

Built-in SRP: High-Definition

- More features!
 - Materials: SSS, Anisotropic, Clearcoat, Iridescent, Rough Refraction, Layered
 - Lighting: Area lights, better probes, better shadows, volumetrics, ...
 - Lots of debug views
- Requires compute (DX11 HW)
- Tile/Clustered Forward/Deferred
- “The Road toward Unified Rendering”,
Lagarde @SIGGRAPH’18
 - <http://advances.realtimerendering.com/s2018>



Built-in SRPs

- Full live source code of both LWRP & HDRP
 - <https://github.com/Unity-Technologies/ScriptableRenderPipeline>
- Look at how things are done!
- Extend, simplify or modify them!

Modifying built-in SRPs

- “Customizing a production pipeline”, Lira @SIGGRAPH’18
 - <https://www.slideshare.net/PhilLira1/customizing-a-production-pipeline-110018900>
- “Boat Attack” sample project
 - <https://github.com/VerasI/BoatAttack/>

Build your own SRP!

- “Render Pipeline From Scratch”, Bastian @SIGGRAPH 2018
 - <https://github.com/pbbastian/SRPFromScratch/blob/master/Slides.pdf>
 - <https://github.com/pbbastian/SRPFromScratch>

Ask me questions