Make Your Game Run on the Quest

(And Look  Too)

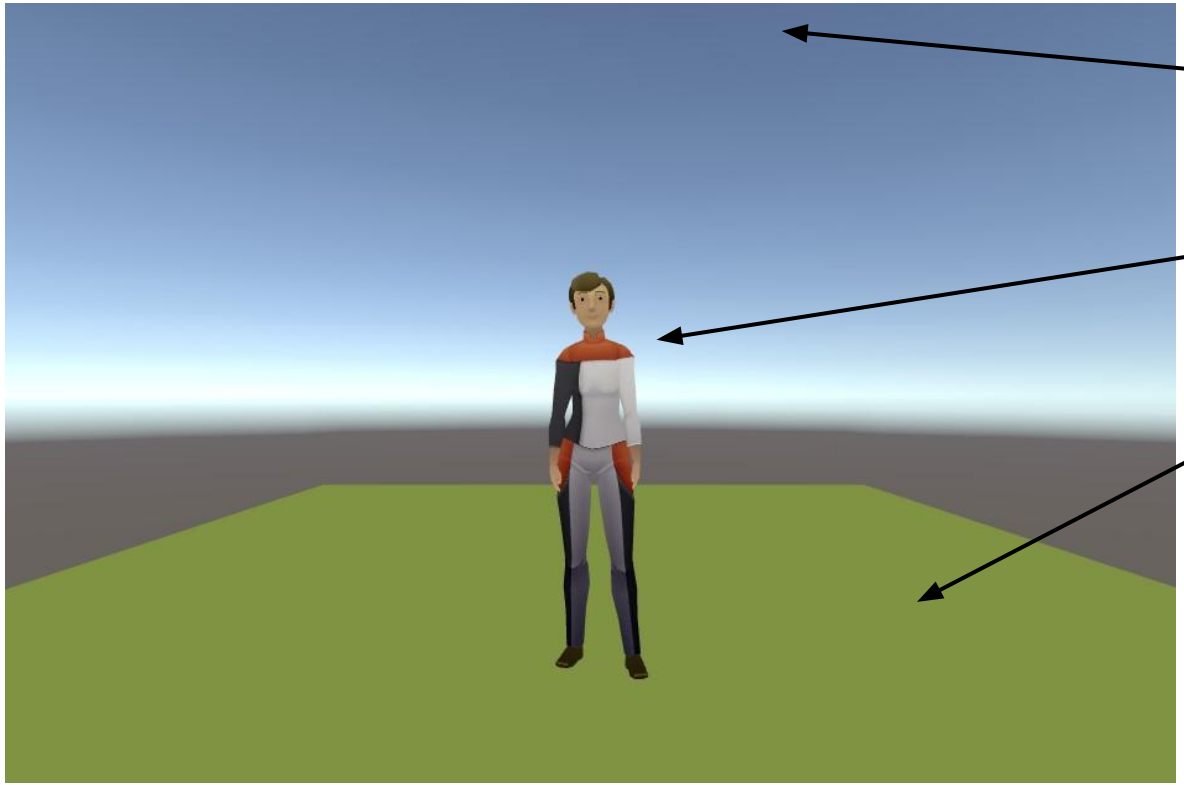

luna
Funomena

What is this Talk?

The Quest!

Let’s Talk About Draw Calls

DrawCalls



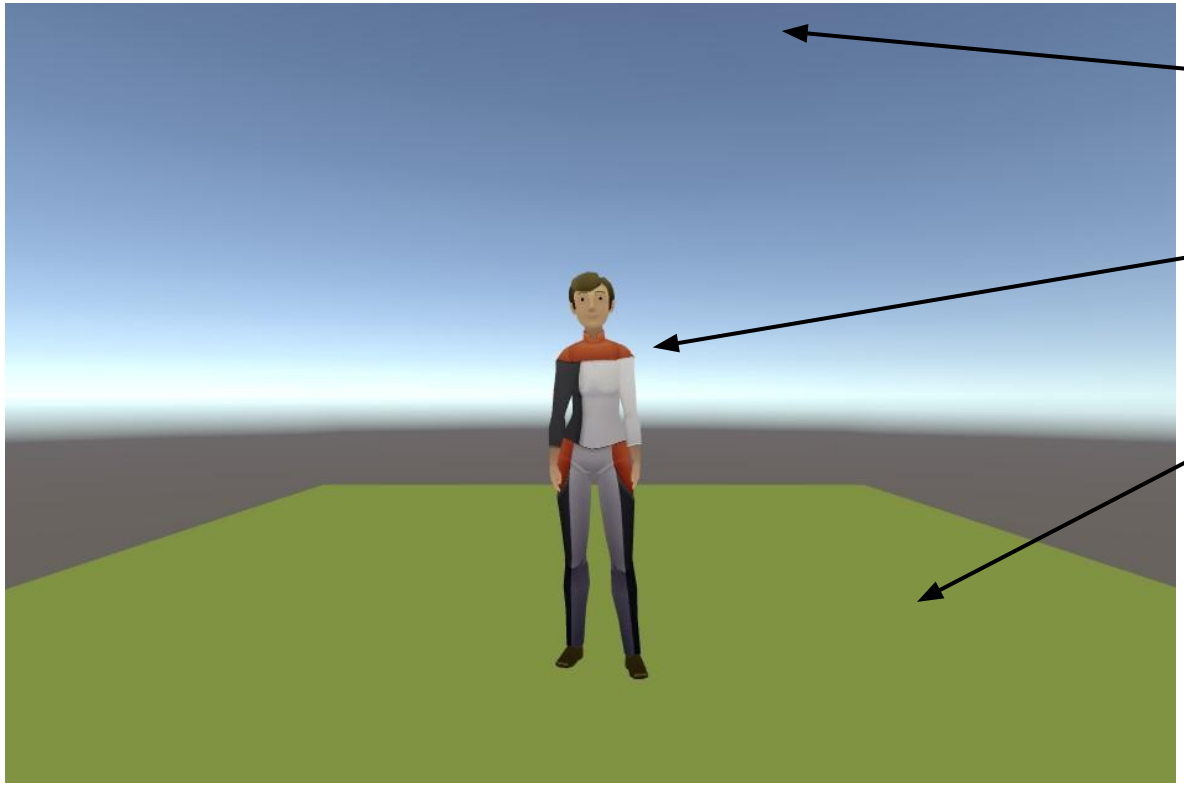
Skybox
1

Character
1

Ground plane
1

3

DrawCalls



Skybox
1

Character
4

Ground plane
1

6

DrawCalls

- 1 Call per mesh/object
- 1 Call per unique material (or material instance) on that object
- Limit number of materials on a mesh
- Atlas textures to reduce number of materials
- Merge meshes where you can

Tools for Art Performance

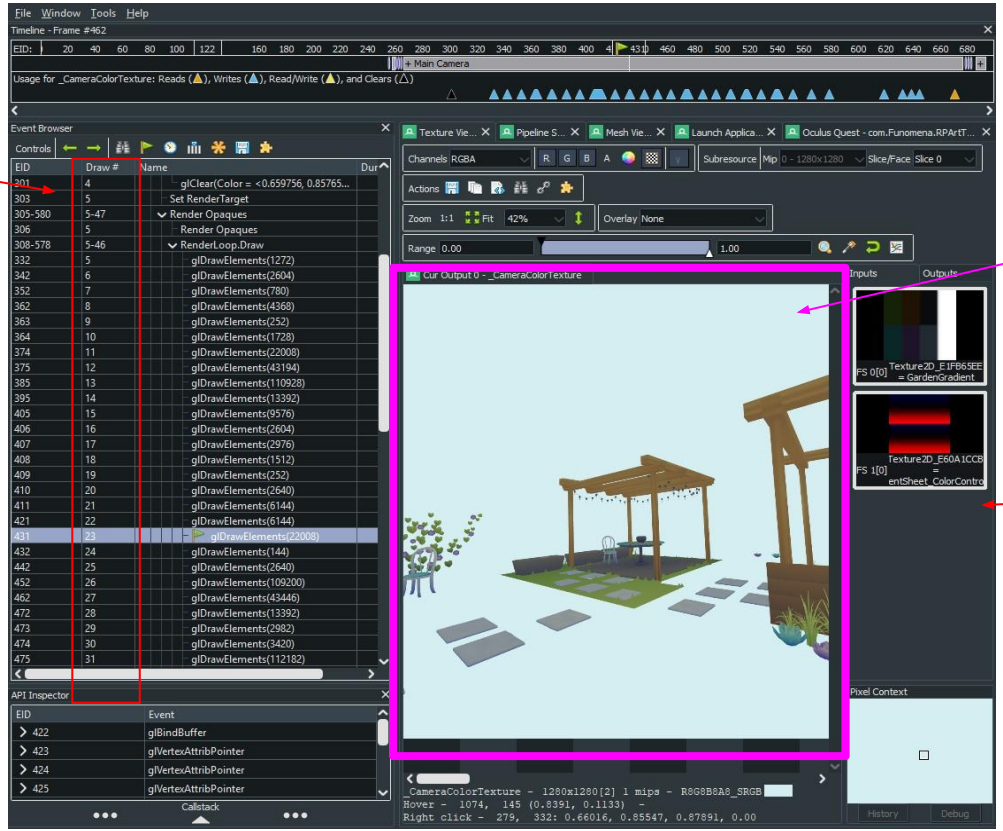
RenderDoc

- Uses connection to the Quest to capture a frame as the Quest would draw it
- See total number of draw calls
- Step through individual draw calls
- Find potential problems for performance.

RenderDoc

Draw Call Number

These are the individual draw calls, ok



This is what the render looks like at this point in the calls

Hey cool, the textures being used in this call!

OVRMetrics/ FPS Counter

- FPS is the most important indicator of performance!
- Ideally keep it at 72, but at least above 65
- Use a FPS counter to see what the framerate is at runtime

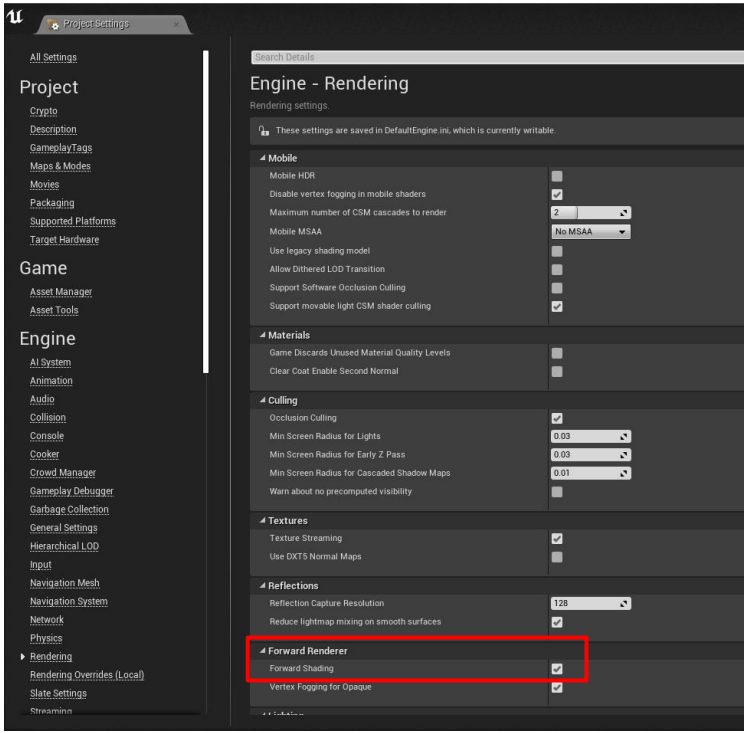
OVRMetrics/ FPS Counter

The Basics

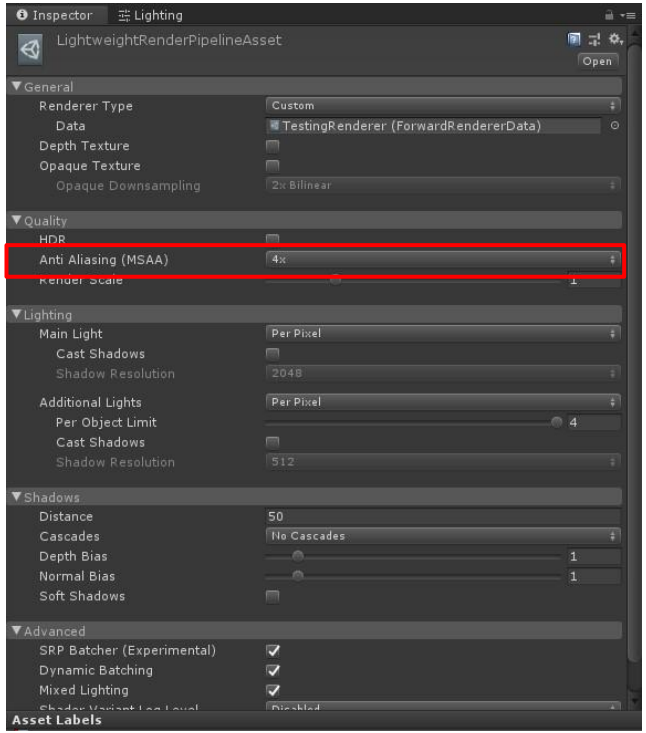
The Basics

- Use Forward Rendering
- No Depth Rendering
- Single Pass Stereo
- Anti-Aliasing
- Fixed Foveated Rendering

Forward Rendering



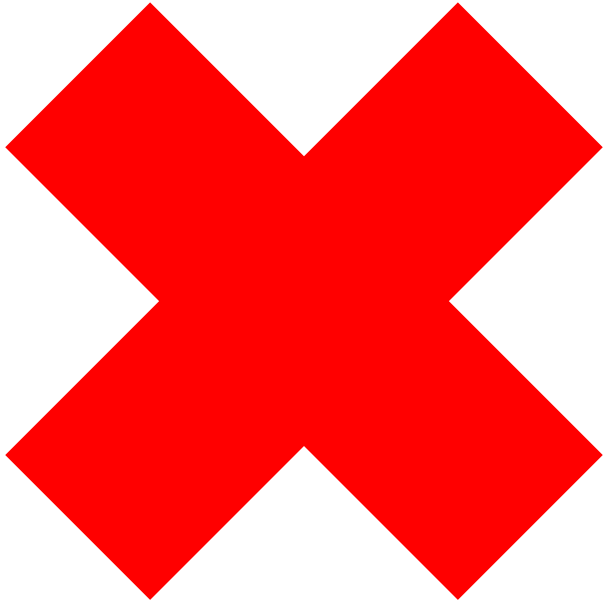
Unreal



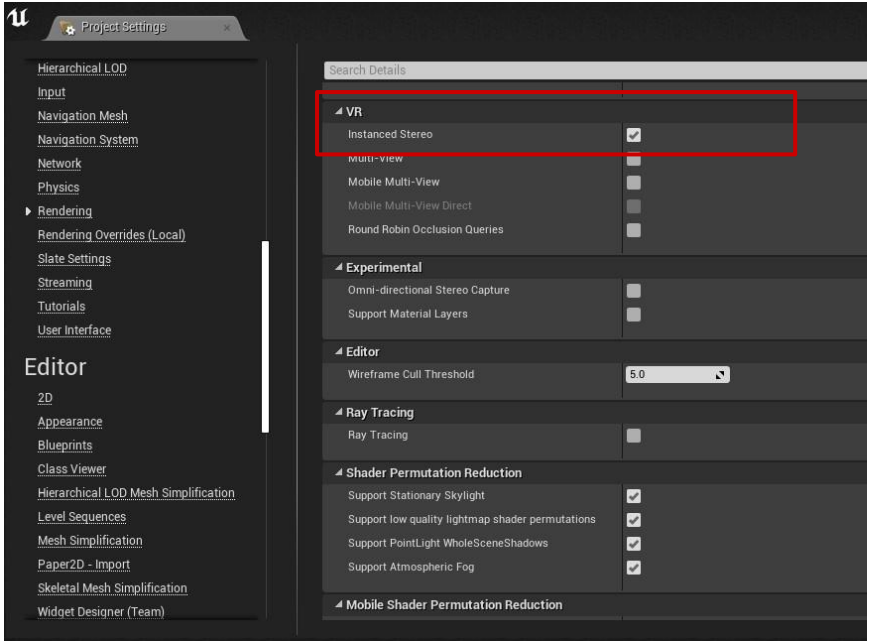
Unity

Depth Pass

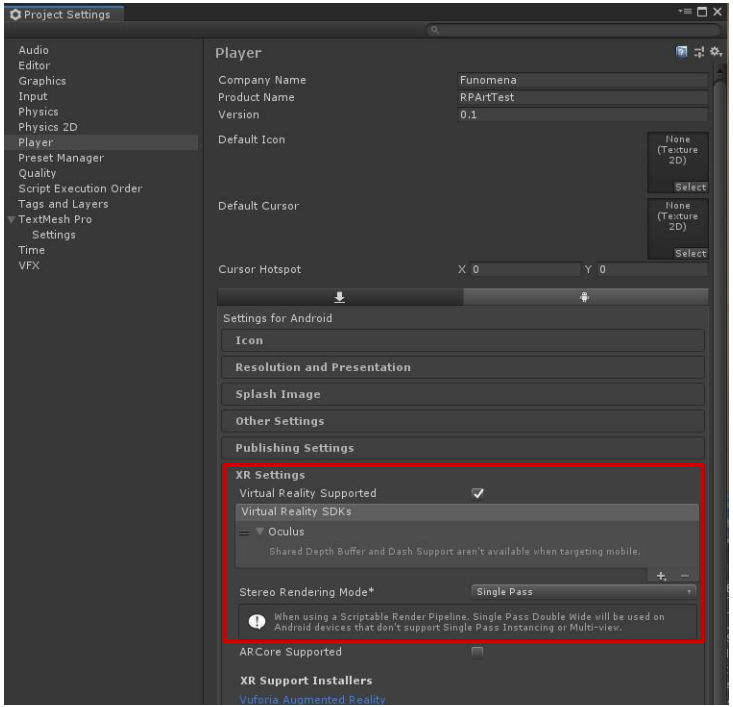
- Nope, sorry



Single Pass/ Instanced Stereo

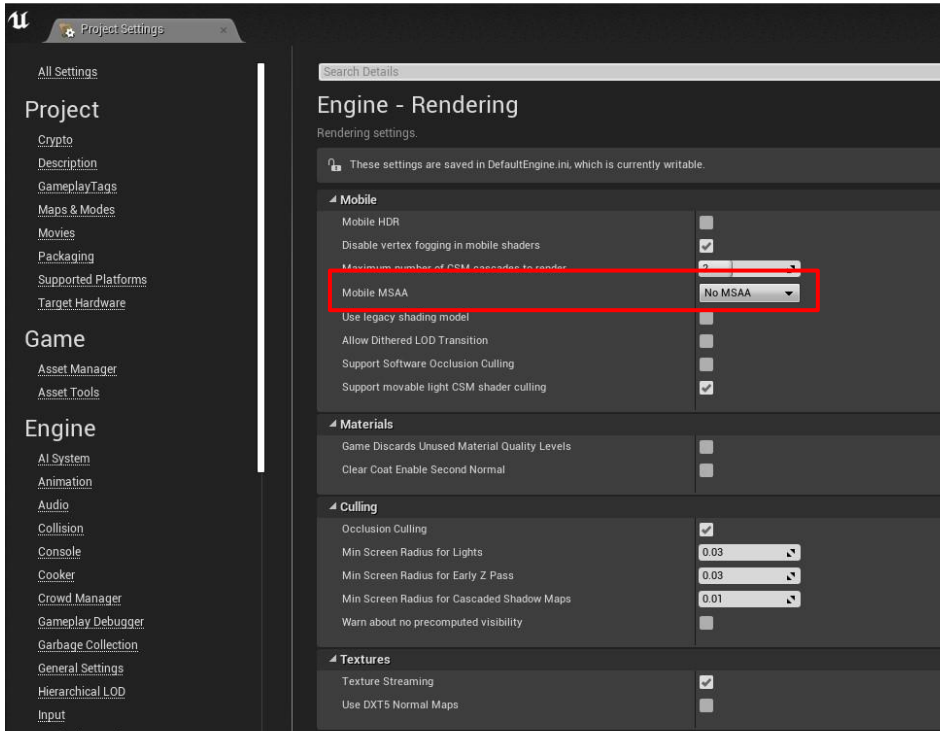


Unreal

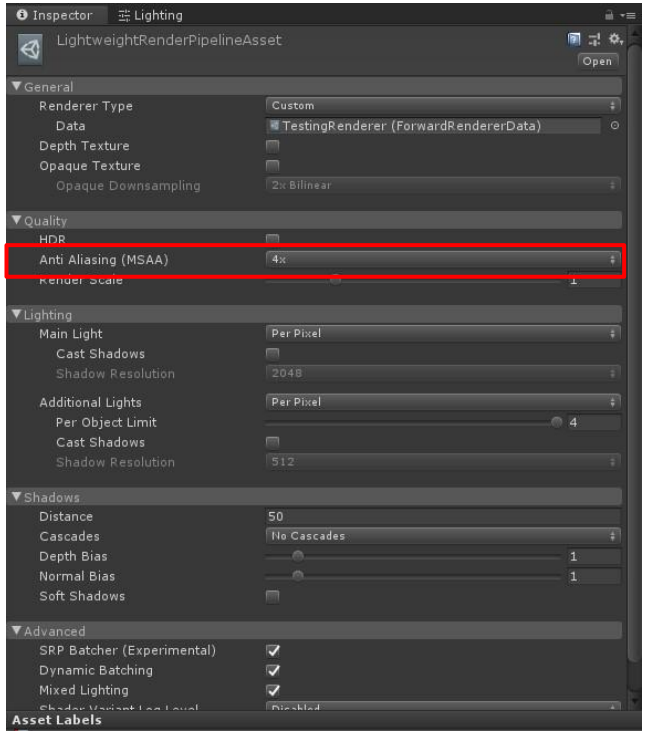


Unity

Anti-Aliasing

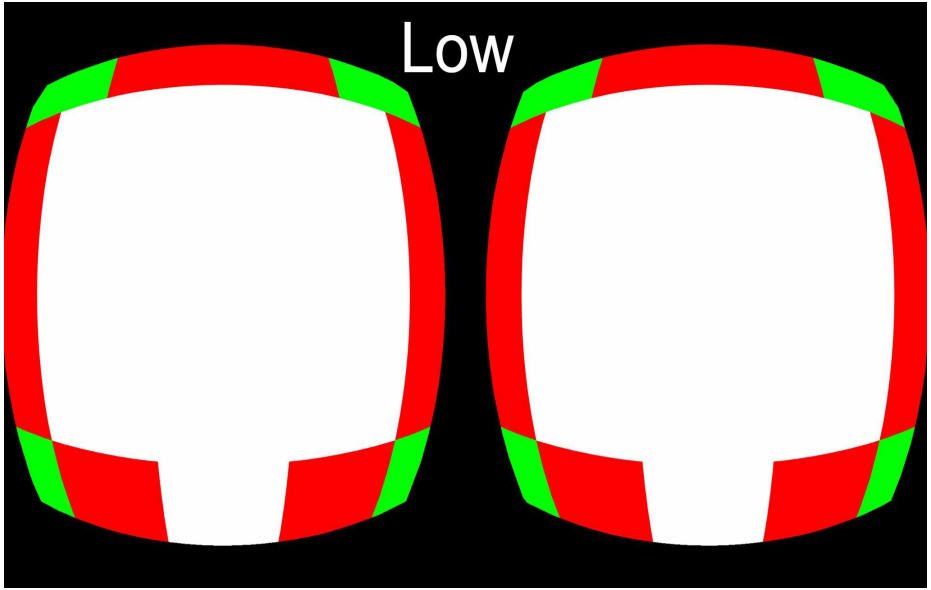


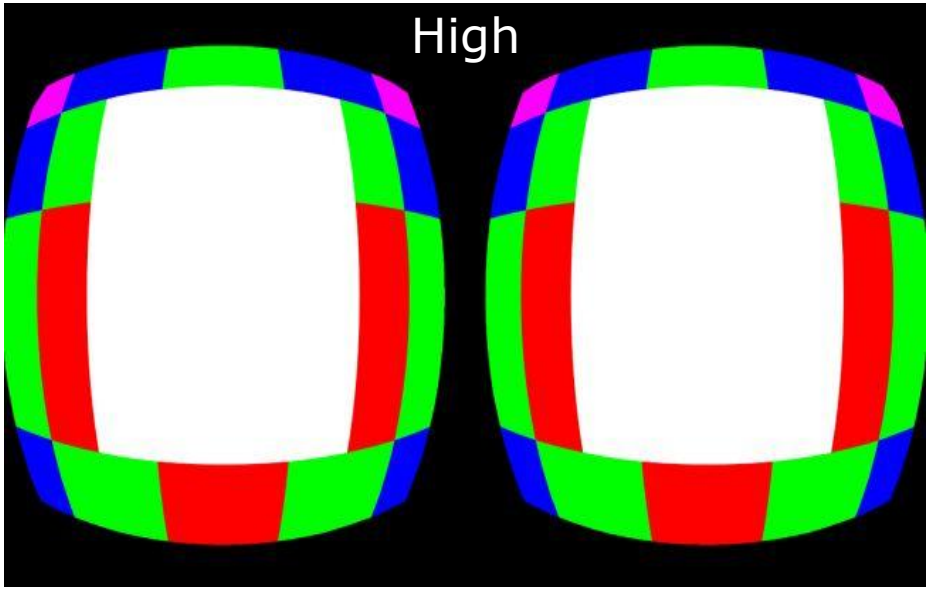
Unreal



Unity

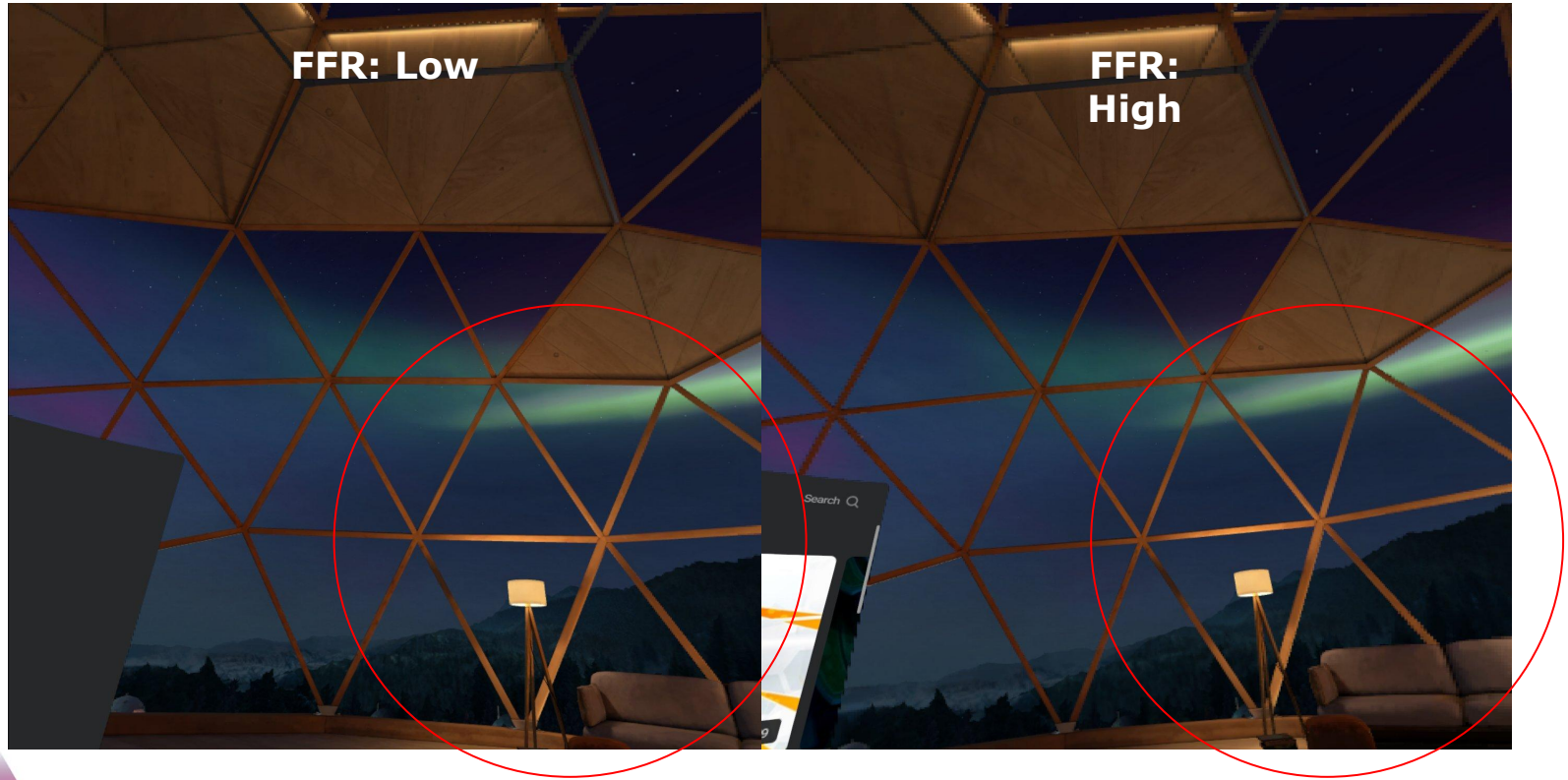
Fixed Foveated Rendering





Lower Resolution

Fixed Foveated Rendering



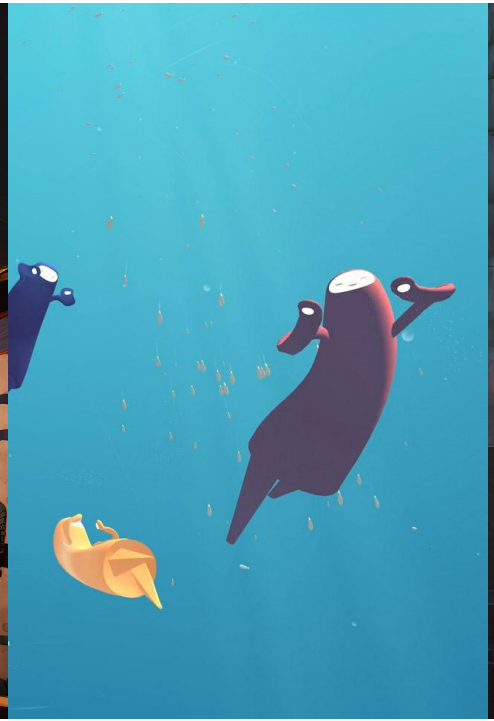
Lights and Shadows

Lights and Shadows

Vader Immortal
ILMxLAB

Moss
Polyarc

Fisherman's Tale
Vertogo Games



Half & Half
Normal

Lights and Shadows

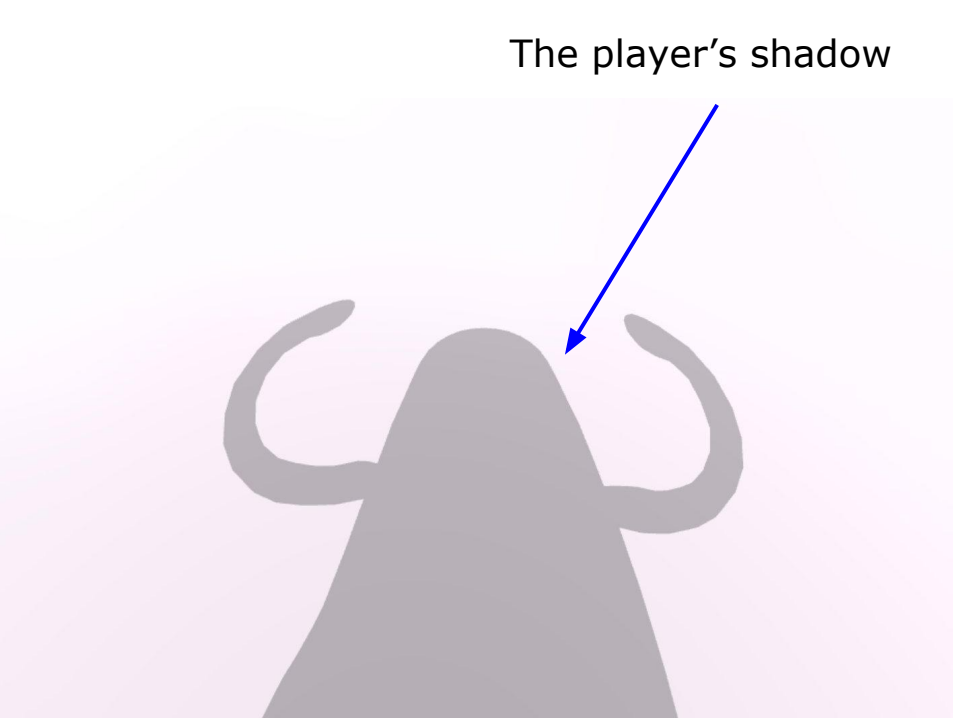
- Bake lights to get more performant lit looks
  - But remember that baking lightmaps takes time!
  - Balance time against team size and number of environments
- Only one dynamic light at a time
  - Consider baking in static areas anyway

Lights and Shadows

- Dynamic shadows are really expensive
  - Only one shadow casting light at a time
  - Only hard shadows
  - Avoid if possible, unless game is rendering VERY lightweight

Lights and Shadows

Bonfire
Baobab Studios



Half + Half
Normal

Lights and Shadows

Virtual Virtual Reality
Tender Claws

GNOG
KO_OP

Lights and Shadows

- Unlit shaders perform really well
  - Eliminate time spent lighting and baking lightmaps
  - Can require less textures overall
- Lit, but cel-shaded as an alternative
  - Less to compute without going fully unlit

Shaders, Materials, and Textures

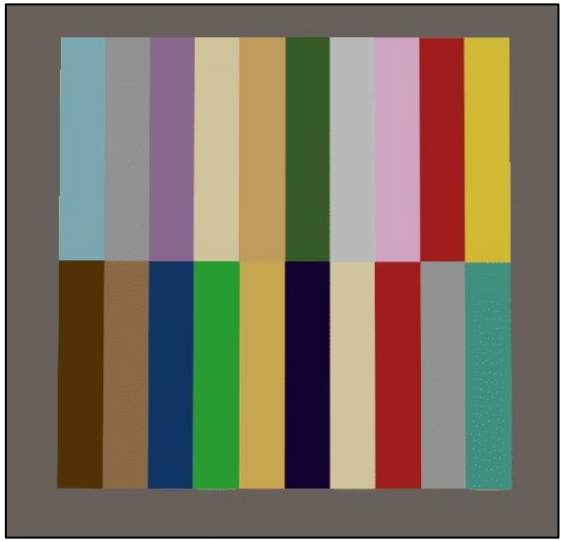
Textures

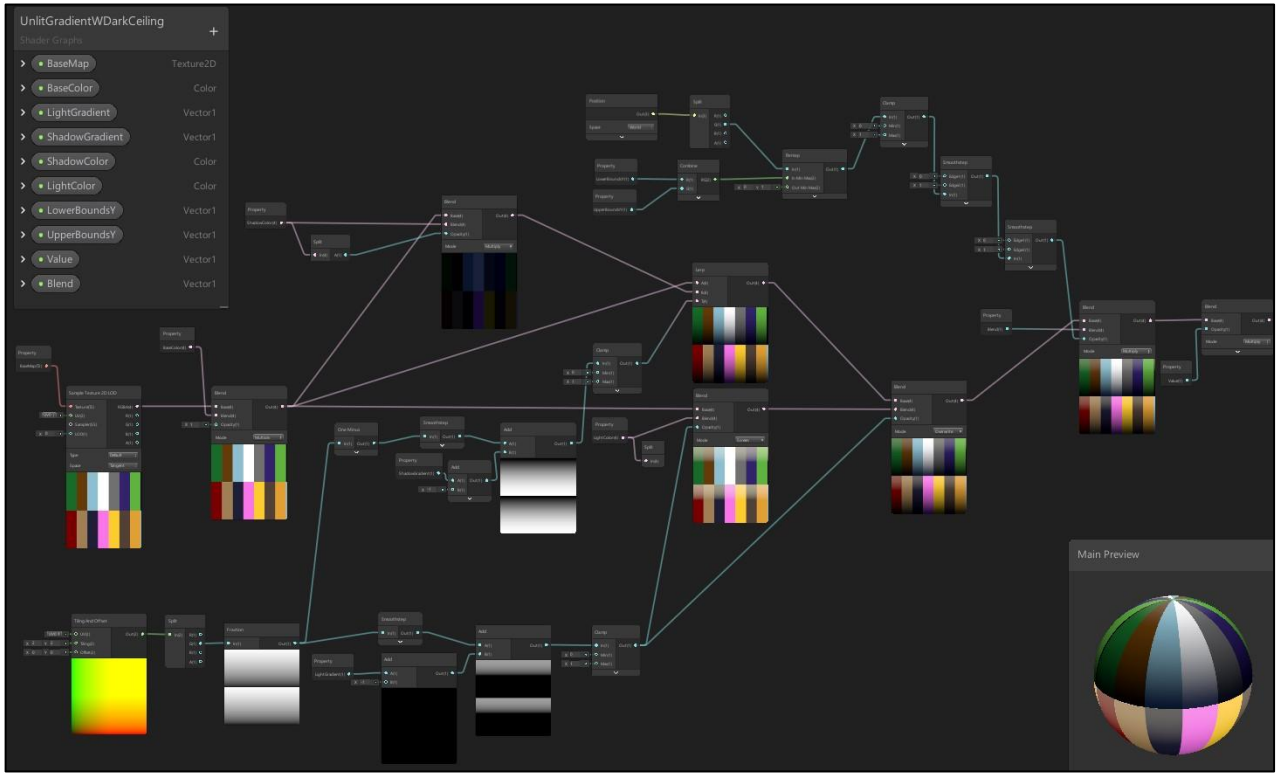
- Keep texture resolutions low
- Use as few maps as possible
- Perhaps try going without certain maps, or packing into RGBA channels
- Reuse and tile what you can
- Don’t forget your mip levels!

Shaders

- Number of instructions impact performance
- Number of textures impact performance, especially if they tile across the screen
- But shaders can also be really helpful to create beautiful and unique looks
- Experiment with lightweight shaders and see what unexpected work they can do

Shaders




UnlitGradientWDarkCeiling
Shader Graphs
BaseMap
Texture2D
BaseColor
Color
LightGradient
Vector1
ShadowGradient
Vector1
ShadowColor
Color
LightColor
Color
LowerBoundsY
Vector1
UpperBoundsY
Vector1
Value
Vector1
Blend
Vector1
Main Preview

Materials

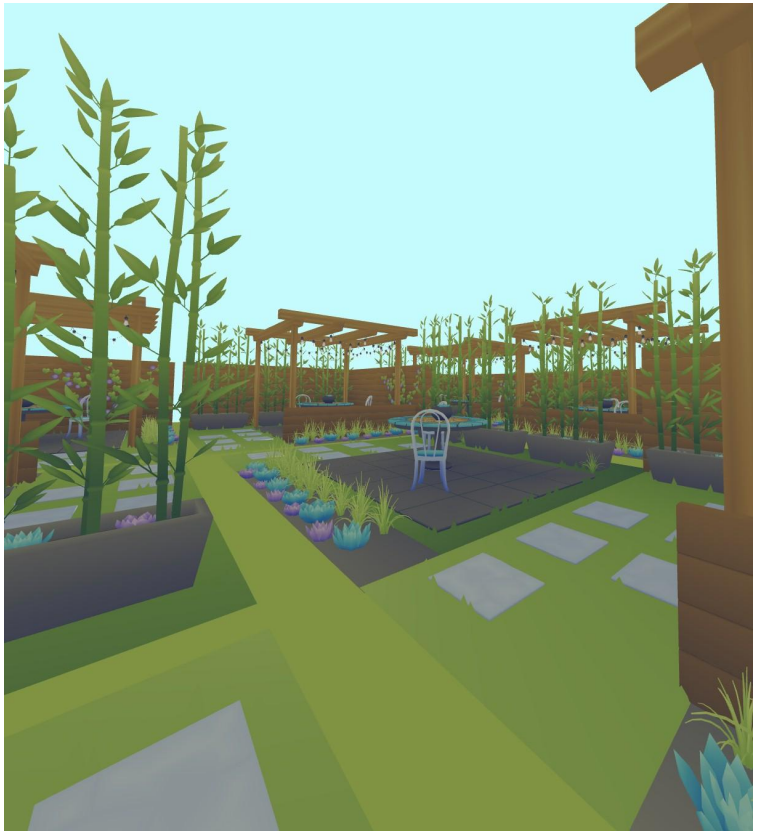
- Switching materials and shaders has a slight performance hit per draw call
- Atlas what you can to reduce number of unique materials, even if it won’t reduce draw calls
- Consolidate shaders if you can to limit the number of unique ones

Poly Count

Poly Count

FPS: **63**

Poly Count:
500K Verts



Poly Count

- Instancing in memory only does not reduce draw calls
- *Some* kinds of instances DO combine/batch draw calls
- LODs are still there, and still work!
- Some kinds of instances also use LODs *and* batch draw calls

Poly Count

- Use good industry practices, especially when working from high to low poly
- Explore new styles!
- Integrate useful and applicable low-poly stylizations into your game as solutions for difficult problems
    - Example: Look at how trees or foliage are handled in low poly styles, if you’re having trouble with performance using traditional methods.

Job Simulator
Owlchemy Labs

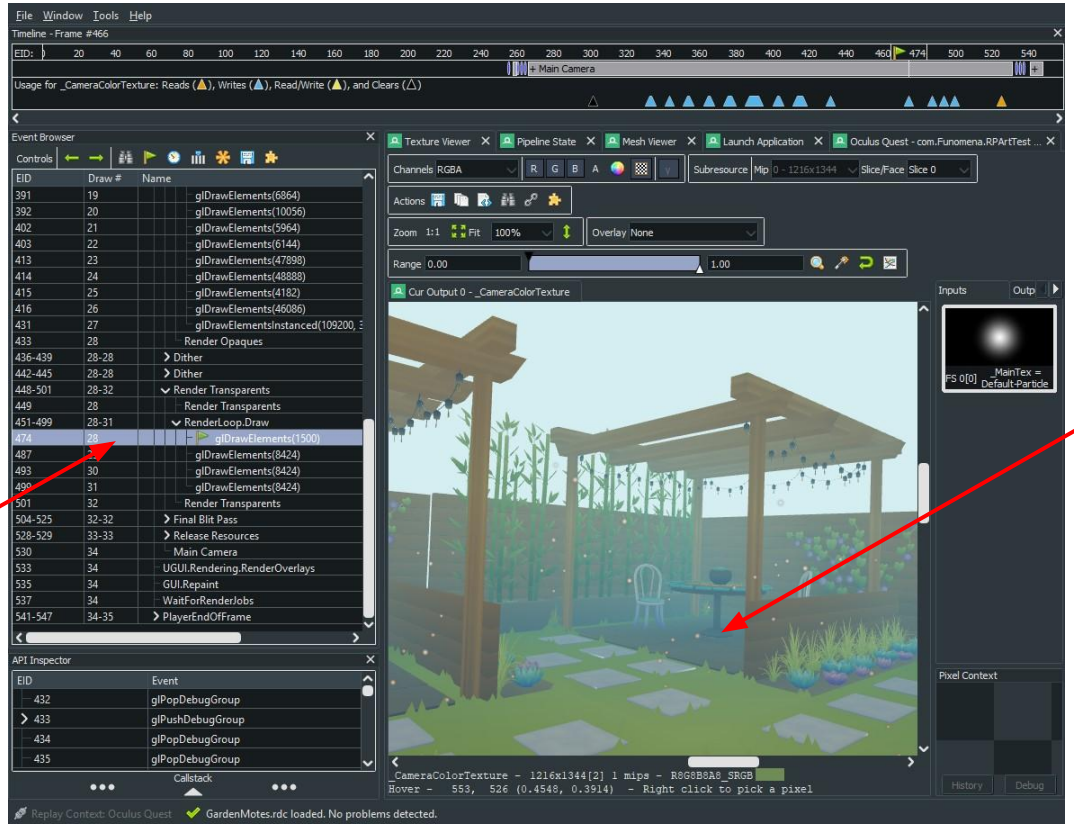
Virtual Virtual Reality
Tender Claws

SUPERHOT VR
SUPERHOT

Batching

Batching

All the cards in the particle systems are drawn on this call



Dust motes

Batching

- Draw multiple objects on one call!
- Different types and methods in different engines
- All come with a little overhead you should keep in mind
- Figure out if its cheaper to batch or use another solutions like merging

Unity Batching

- Dynamic Batching
  - Same mesh with same material under 300 verts
  - Some overhead, but pretty good for small duplicated objects
- Static Batching
  - Higher poly models, but uses more memory

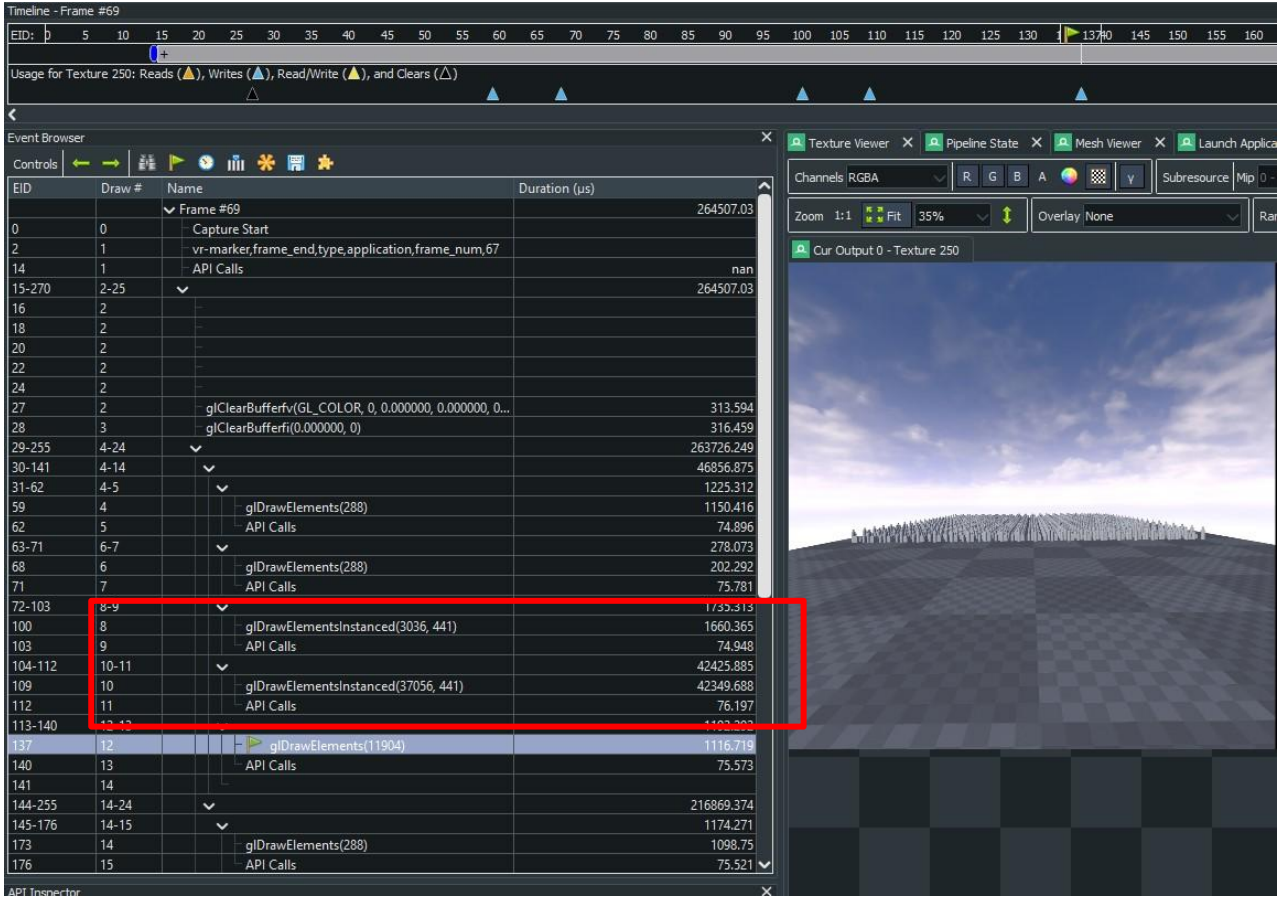
Unreal Batching

- Instanced Static Meshes
  - One draw call, but not much performance saving otherwise

Unreal - HISM

FPS: 5

Draw Calls: 5

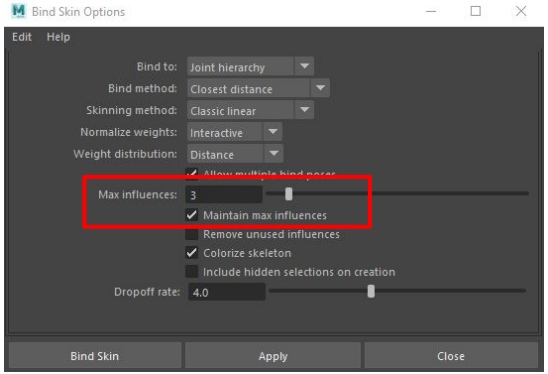


Unreal Batching

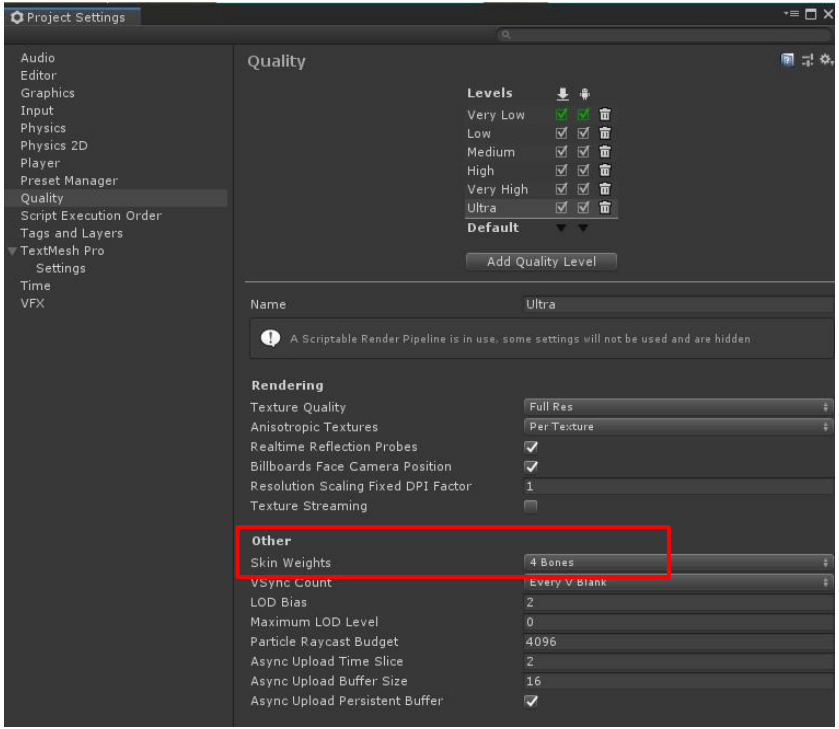
- Instanced Static Meshes
  - One draw call, but not much performance saving otherwise
- Hierarchical Instanced Static Meshes
  - LODs and culling works!
  - Hard to work with, so make a tool to help you

Skinned Meshes

Skinned Meshes



Maya

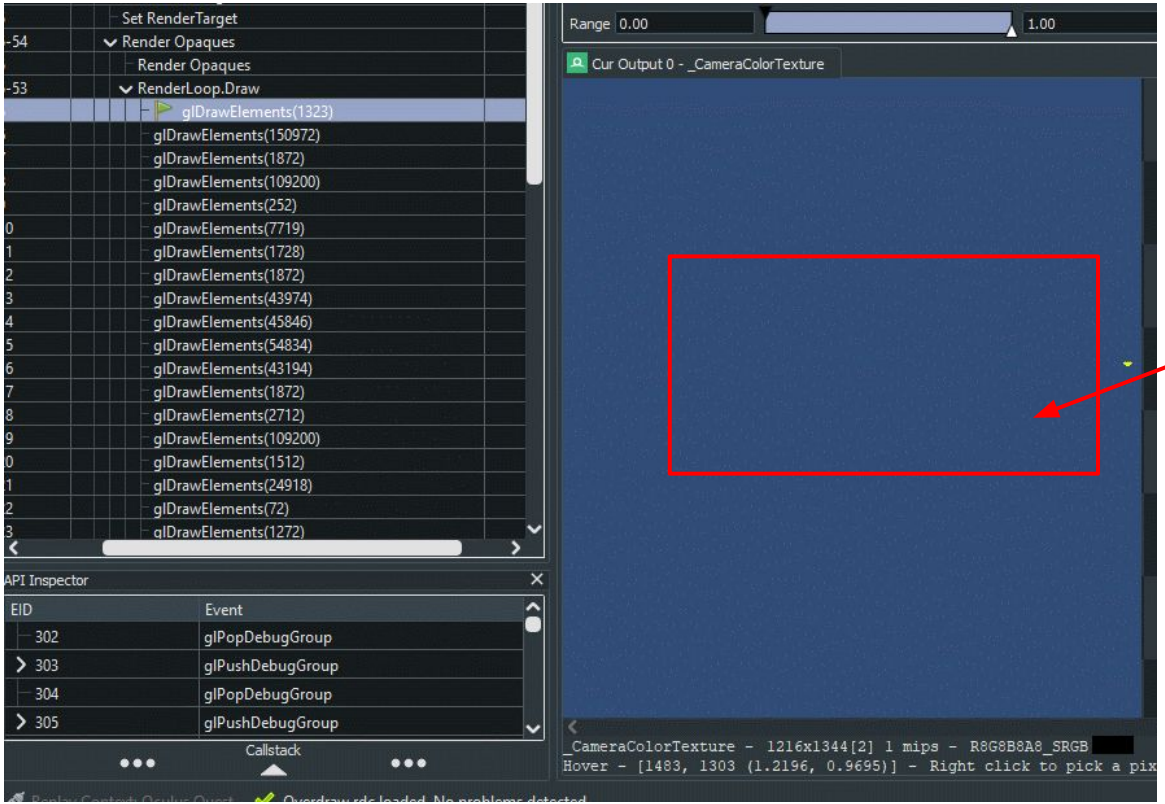


Unity

Skinned Meshes

Transparency

Transparency



Several objects are drawn after this wall, but are completely occluded by it.

Transparency

- Small objects with transparency perform pretty well
- Large transparent objects which overlap are the worst for performance
- Keep use of transparency small and infrequent
- Test on the device!!

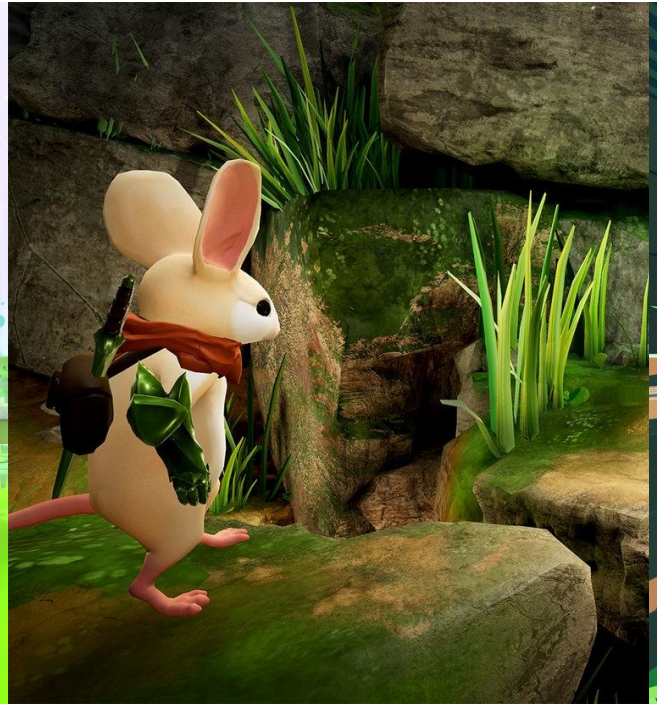
Transparency

Soft alpha card effects have their own issues anyway.

Vertex Fog is an old method which still works.

Transparency

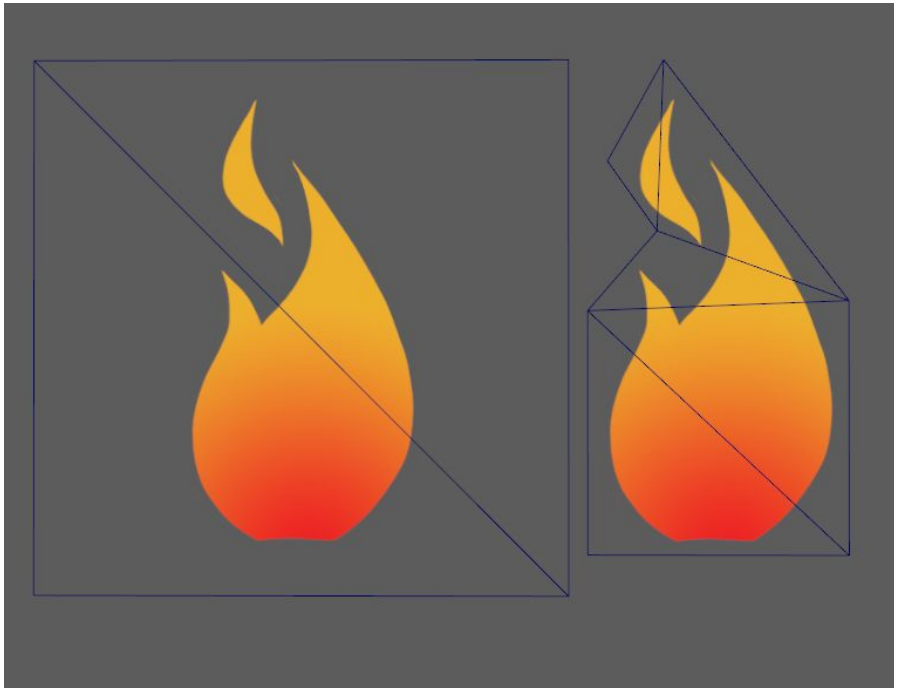
Fuji
Funktronic Labs

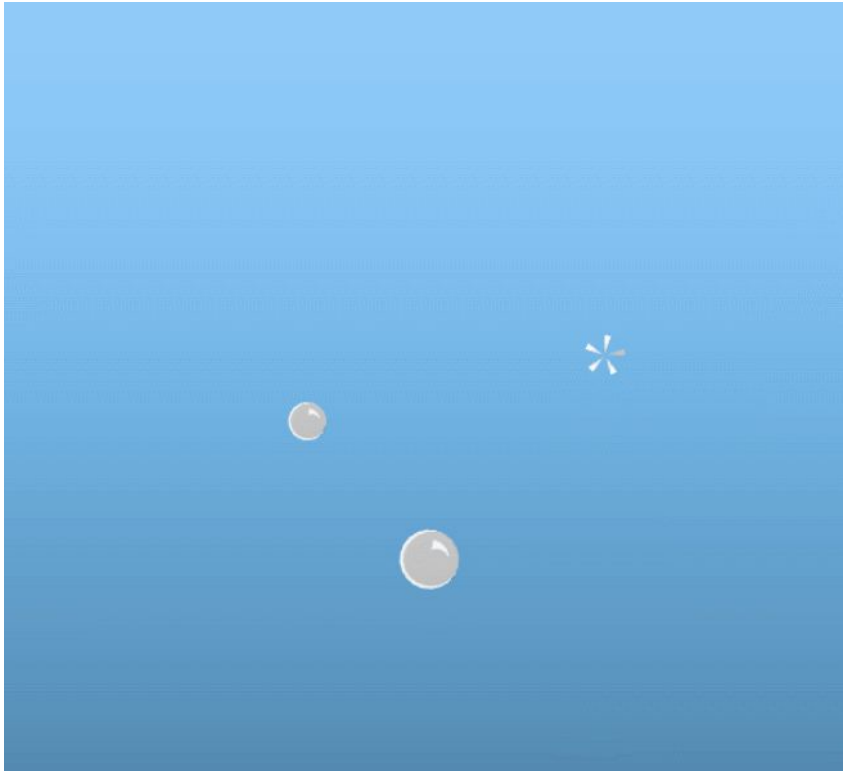


Moss
Polyarc

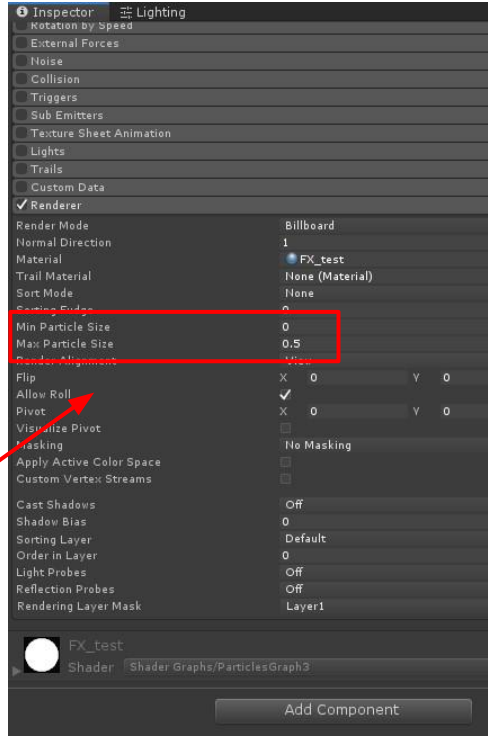
Virtual Virtual Reality
Tender Claws

Transparency

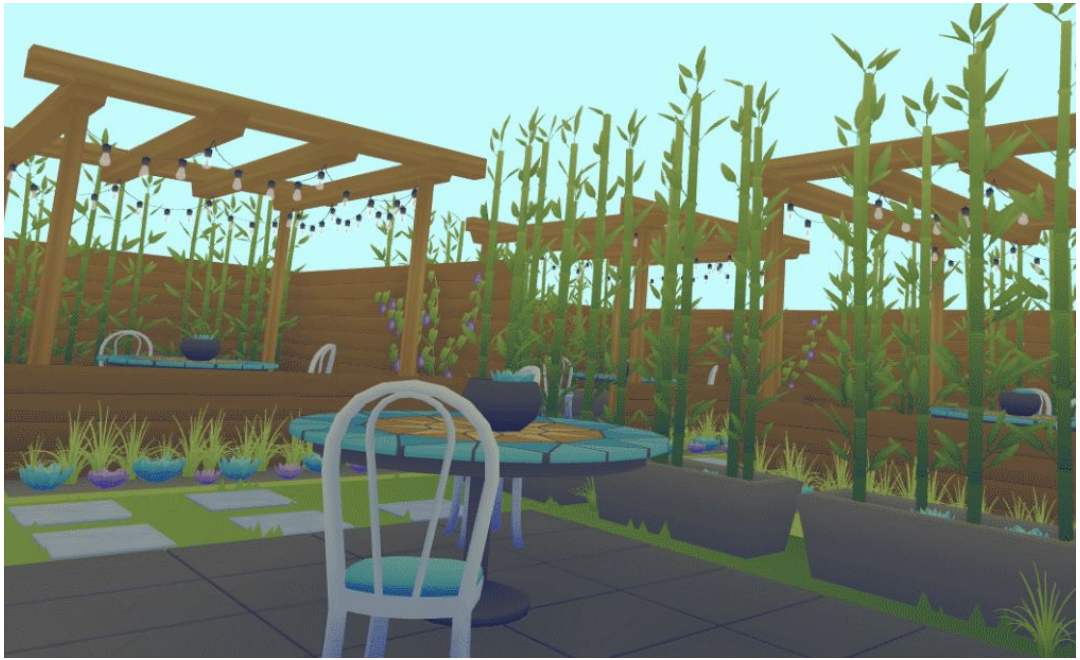




Transparency



Min and max particle size refer to size on-screen



Get creative with transitions! Not everything needs to be a fade

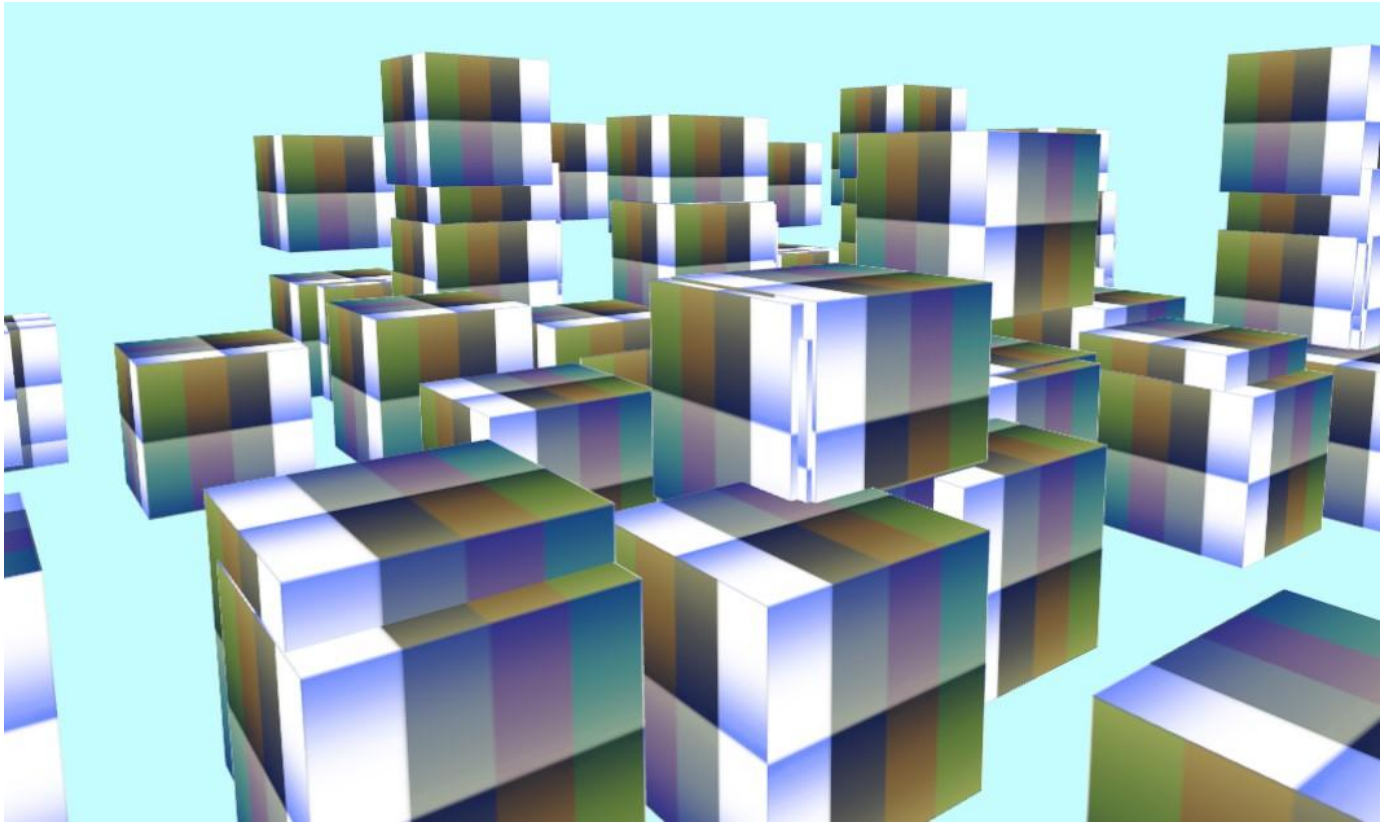
Post Processing

Post Processing

- Color correction can be done in shader
- If you really need bloom in a few places you can use some cards to fake it
- Probably can’t use depth of field, screen overlays and fancy post shaders
- Think about what you would need from post-processing and try to implement in other ways

Test Test Test

Tests

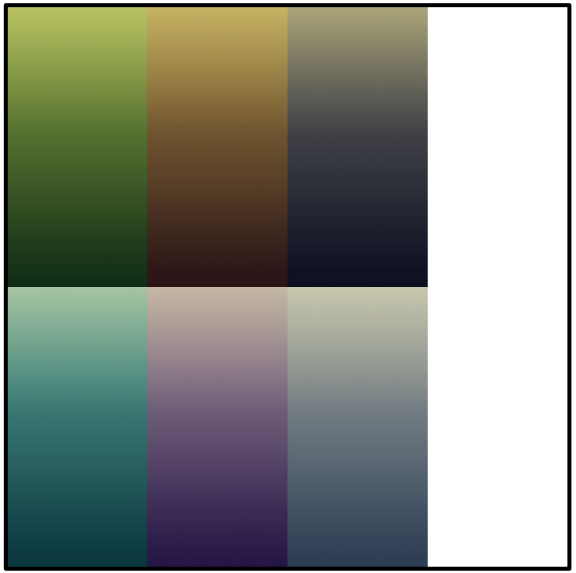


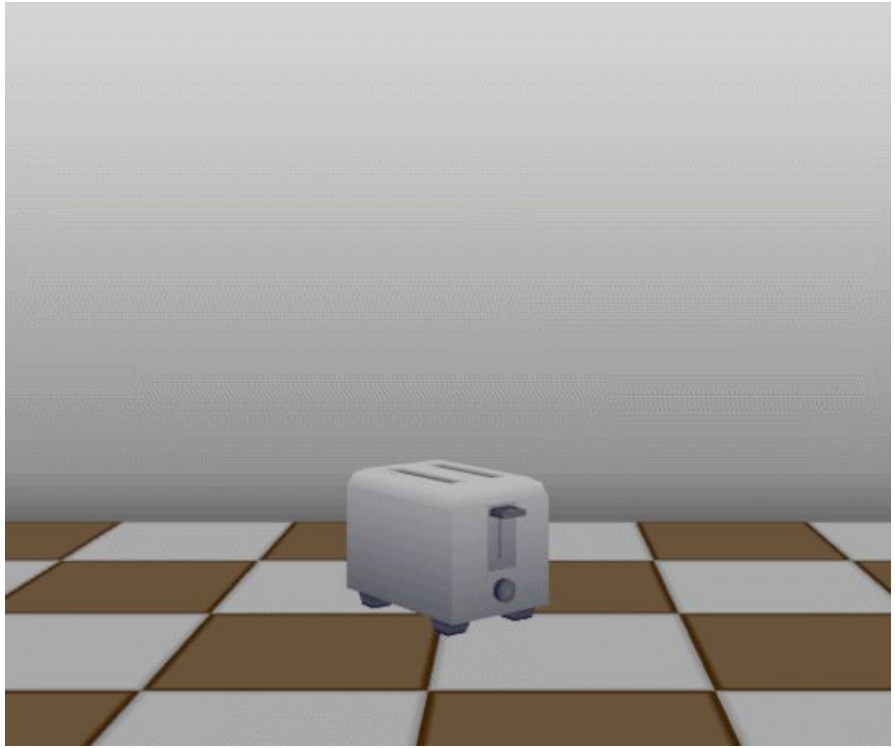
Tests

FPS: **72**

Well, yeah! Almost nothing else is going on in this scene!

Budget Your Art
&
Stay Sane





Summary

Thanks!

- www.allenahail.com
- Twitter: @LenasTeapot