Vulkan Forward Plus Renderer

Milestone 2

Team Member: Ruoyu Fan, Xueyin Wan

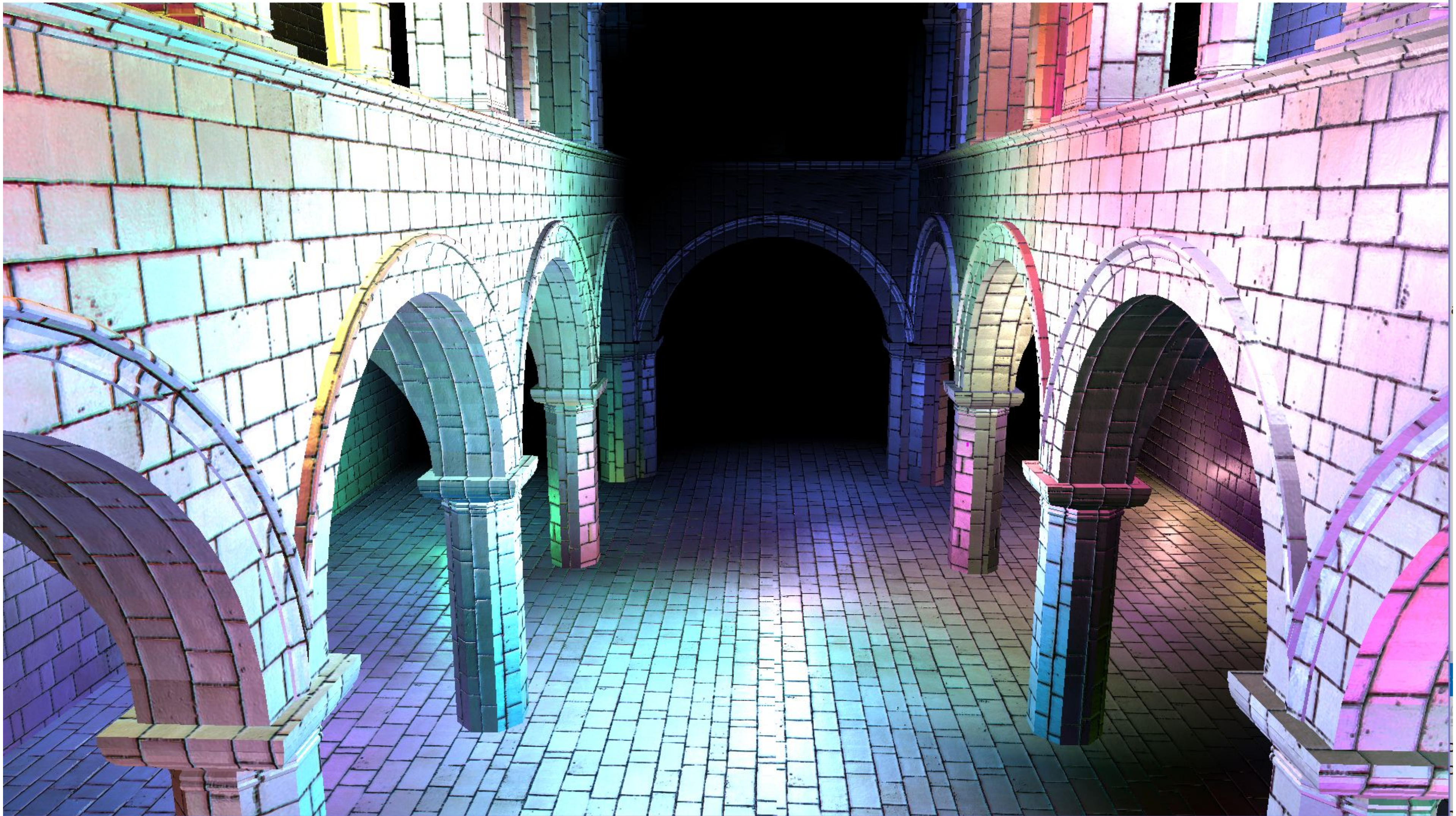
Nov. 28, 2016

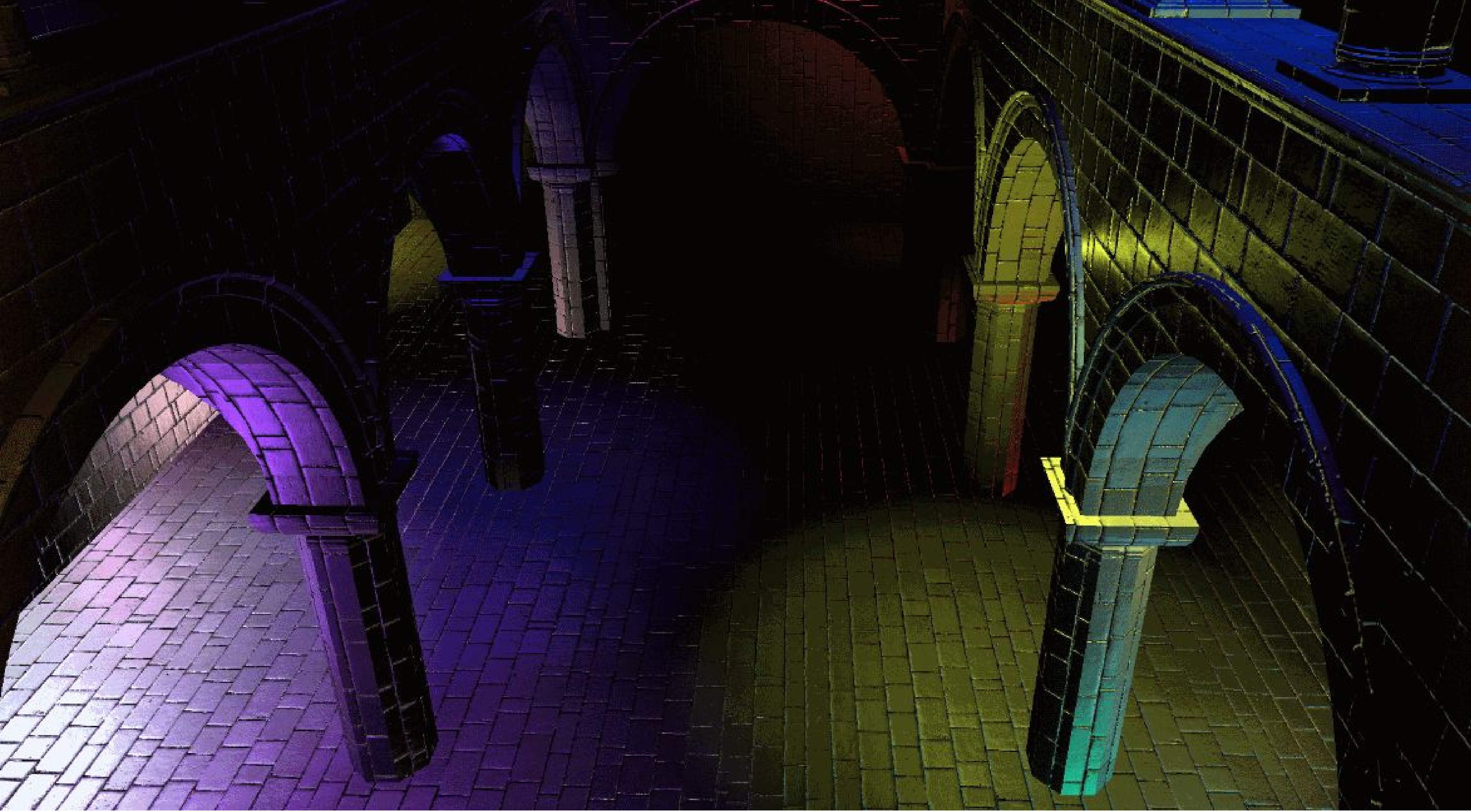
What we have done till Milestone1

- Basic Vulkan Framework including:
    a. texture mapping
    b. mesh loading
    c. rendering pipeline
    d. lambert shading
- Cross Platform using GLFW & CMake (Windows, Linux)

What we have done till Milestone 2

- Forward+ framework in progress:
    a. Fancy Camera Control
        i. Support keyboard(W/A/S/D/Q/E) navigation & mouse control
    b. Huge amount of point light
        i. Currently test with 200 lights
    c. normal mapping
    d. Blinn-Phong Shading
    e. Debug views
    f. light culling (in progress)





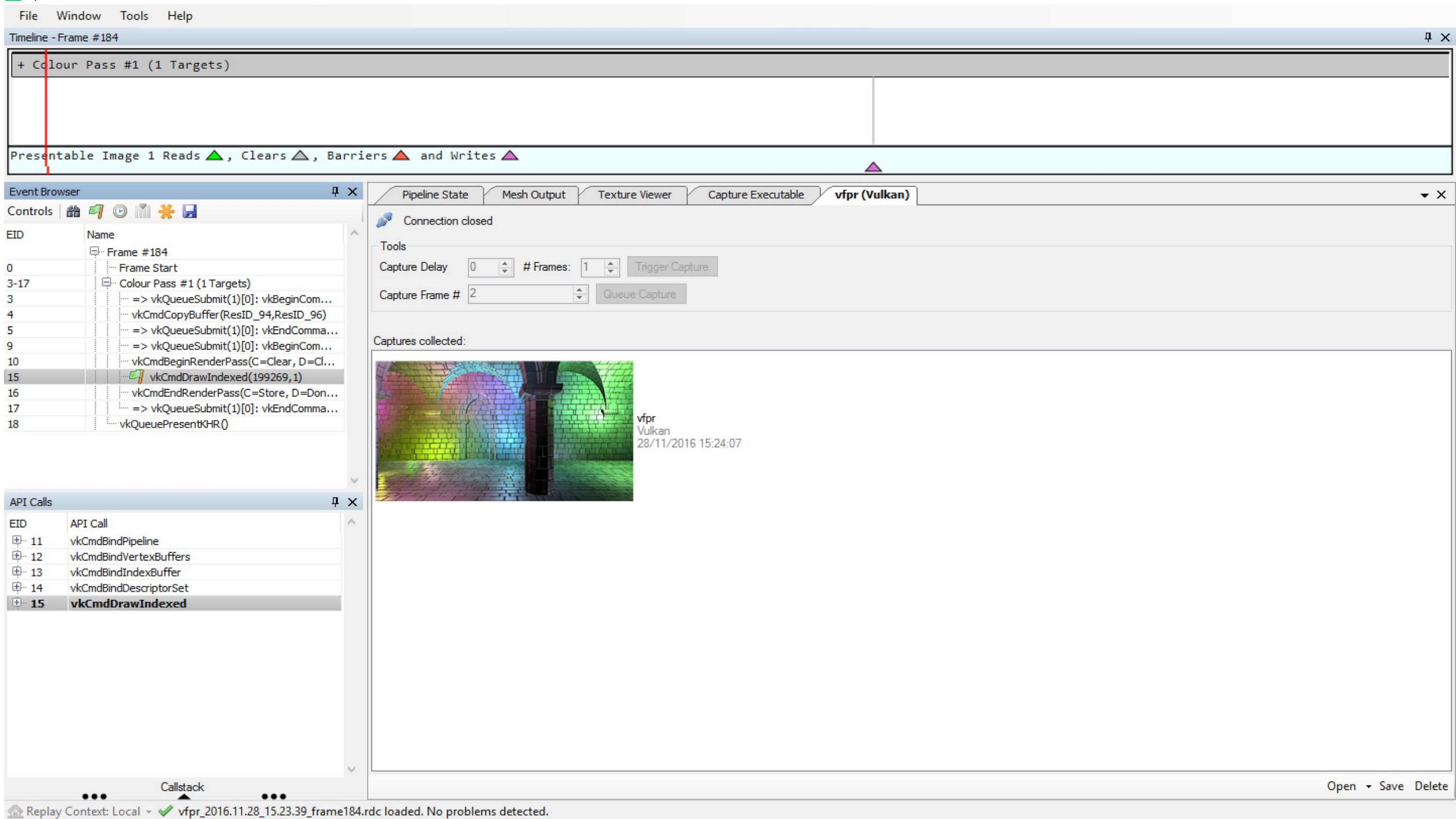
Debug Process

- We use Renderdoc for debugging

    eg: uniform buffer alighment problem (all dark)

```
struct PointLight {
public:
        glm::vec3 pos;
        float radius = { 5.0f };
        glm::vec3 intensity = { 1.0f, 1.0f, 1.0f };
        float padding;    …

};
```



Timeline - Frame #184

+ Colour Pass #1 (1 Targets)

Presentable Image 1 Reads ▲, Clears ▲, Barriers ▲ and Writes ▲

Event Browser

Controls

EID	Name
	⊟ Frame #184
0	Frame Start
3-17	⊟ Colour Pass #1 (1 Targets)
3	=> vkQueueSubmit(1)[0]: vkBeginCom...
4	vkCmdCopyBuffer(ResID_94,ResID_96)
5	=> vkQueueSubmit(1)[0]: vkEndComma...
9	=> vkQueueSubmit(1)[0]: vkBeginCom...
10	vkCmdBeginRenderPass(C=Clear, D=Cl...
15	vkCmdDrawIndexed(199269,1)
16	vkCmdEndRenderPass(C=Store, D=Don...
17	=> vkQueueSubmit(1)[0]: vkEndComma...
18	vkQueuePresentKHR()

API Calls

EID	API Call
⊞ 11	vkCmdBindPipeline
⊞ 12	vkCmdBindVertexBuffers
⊞ 13	vkCmdBindIndexBuffer
⊞ 14	vkCmdBindDescriptorSet
⊞ **15**	**vkCmdDrawIndexed**

Connection closed

Tools

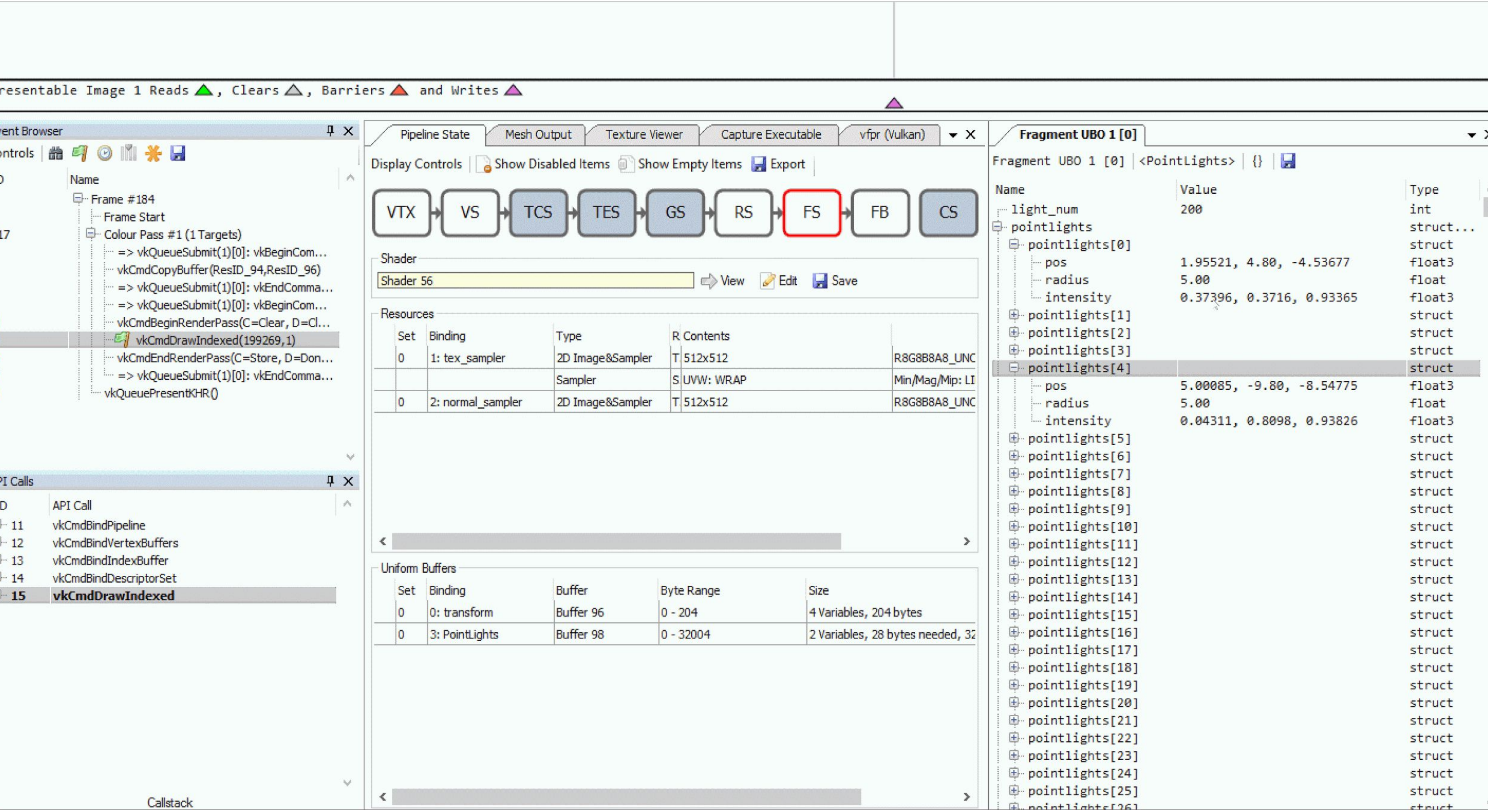
Capture Delay: 0 # Frames: 1

Capture Frame #: 2

Captures collected:

vfpr
Vulkan
28/11/2016 15:24:07

✓ vfpr_2016.11.28_15.23.39_frame184.rdc loaded. No problems detected.



D	Name
	Frame #184
	Frame Start
17	Colour Pass #1 (1 Targets)
	=> vkQueueSubmit(1)[0]: vkBeginCom...
	vkCmdCopyBuffer(ResID_94,ResID_96)
	=> vkQueueSubmit(1)[0]: vkEndComma...
	=> vkQueueSubmit(1)[0]: vkBeginCom...
	vkCmdBeginRenderPass(C=Clear, D=Cl...
	vkCmdDrawIndexed(199269,1)
	vkCmdEndRenderPass(C=Store, D=Don...
	=> vkQueueSubmit(1)[0]: vkEndComma...
	vkQueuePresentKHR()

D	API Call
11	vkCmdBindPipeline
12	vkCmdBindVertexBuffers
13	vkCmdBindIndexBuffer
14	vkCmdBindDescriptorSet
15	**vkCmdDrawIndexed**

Shader

Shader 56

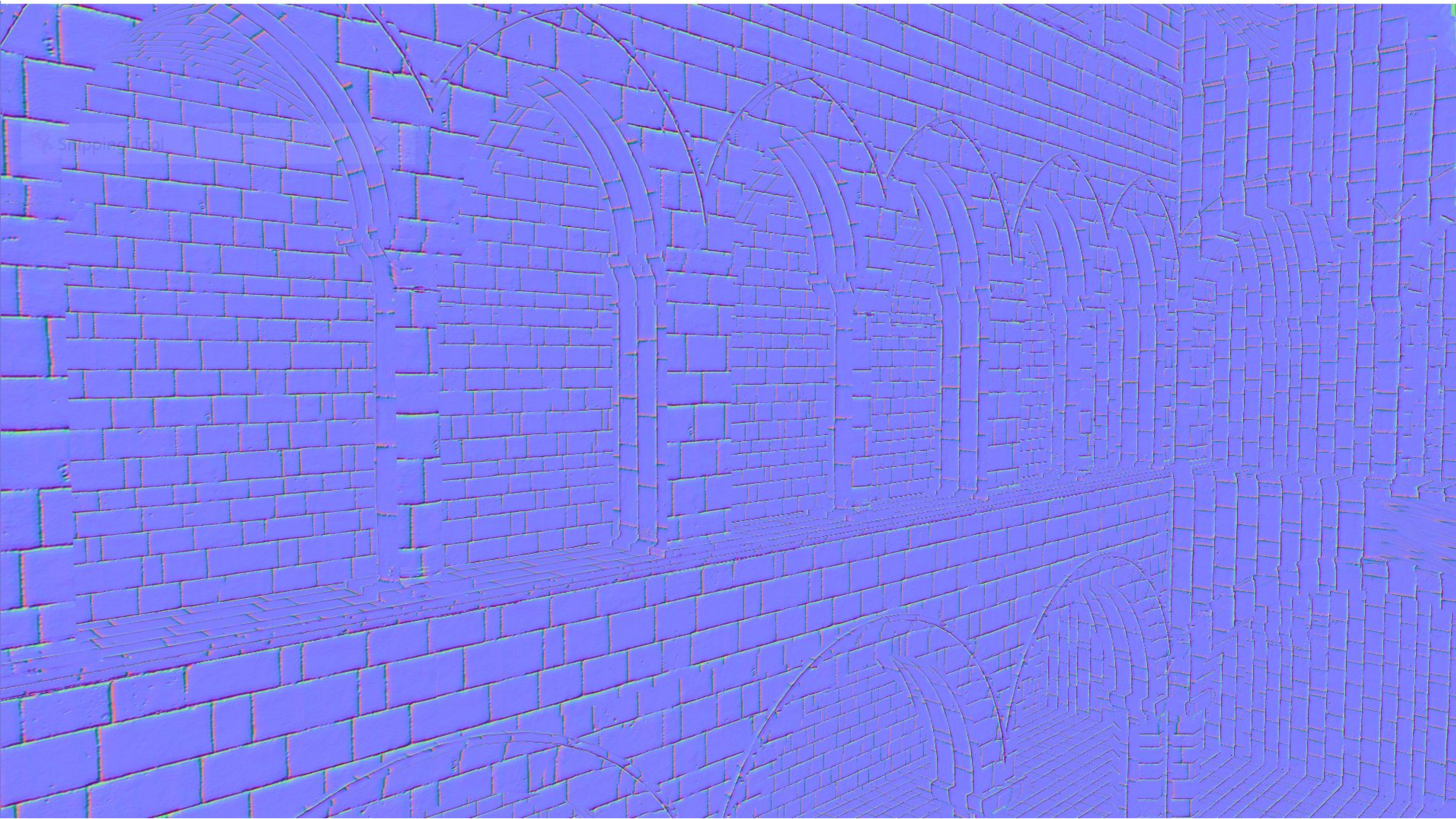
Resources

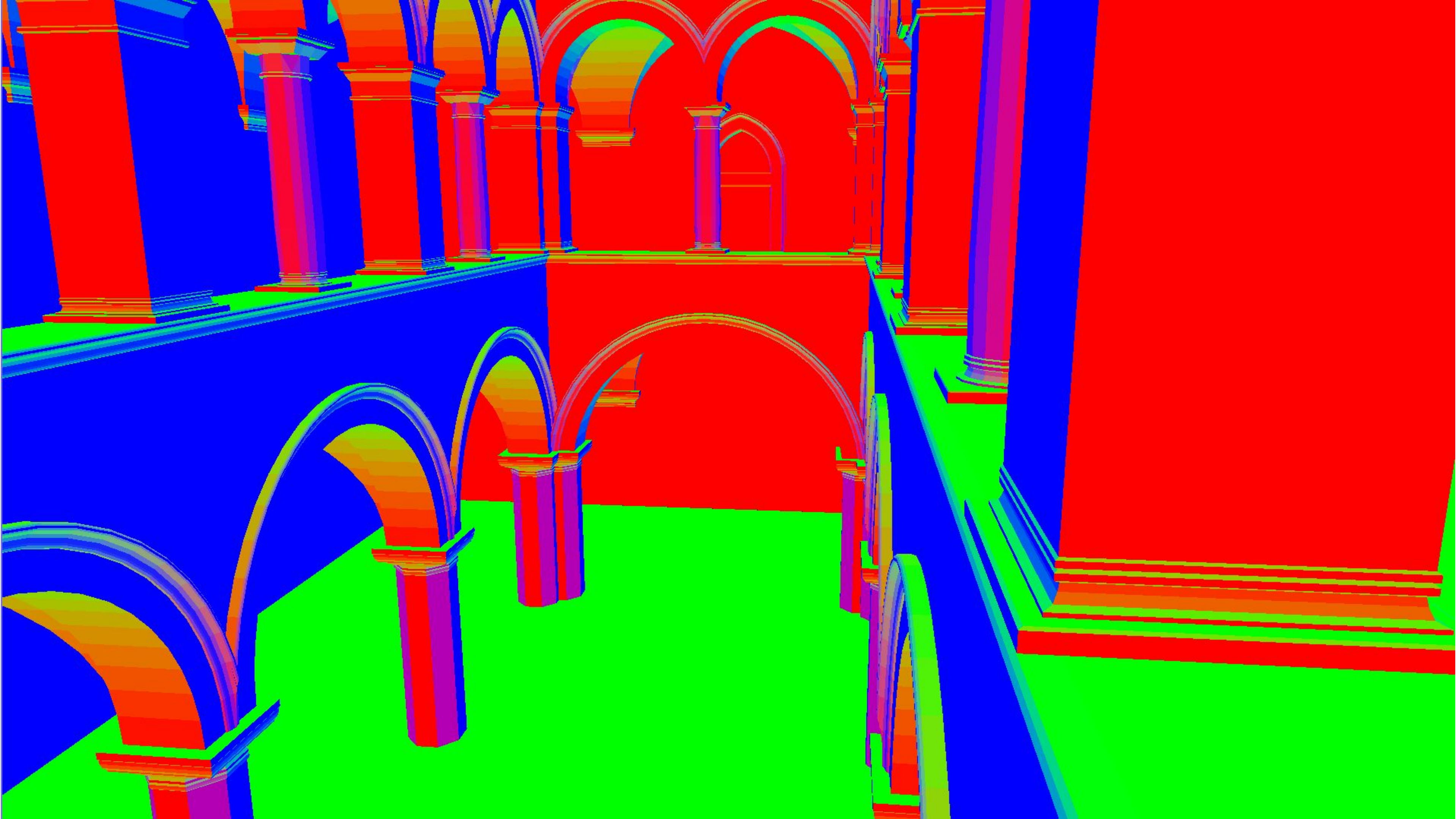
Set	Binding	Type	R	Contents	
0	1: tex_sampler	2D Image&Sampler	T	512x512	R8G8B8A8_UNC
		Sampler	S	UVW: WRAP	Min/Mag/Mip: LI
0	2: normal_sampler	2D Image&Sampler	T	512x512	R8G8B8A8_UNC

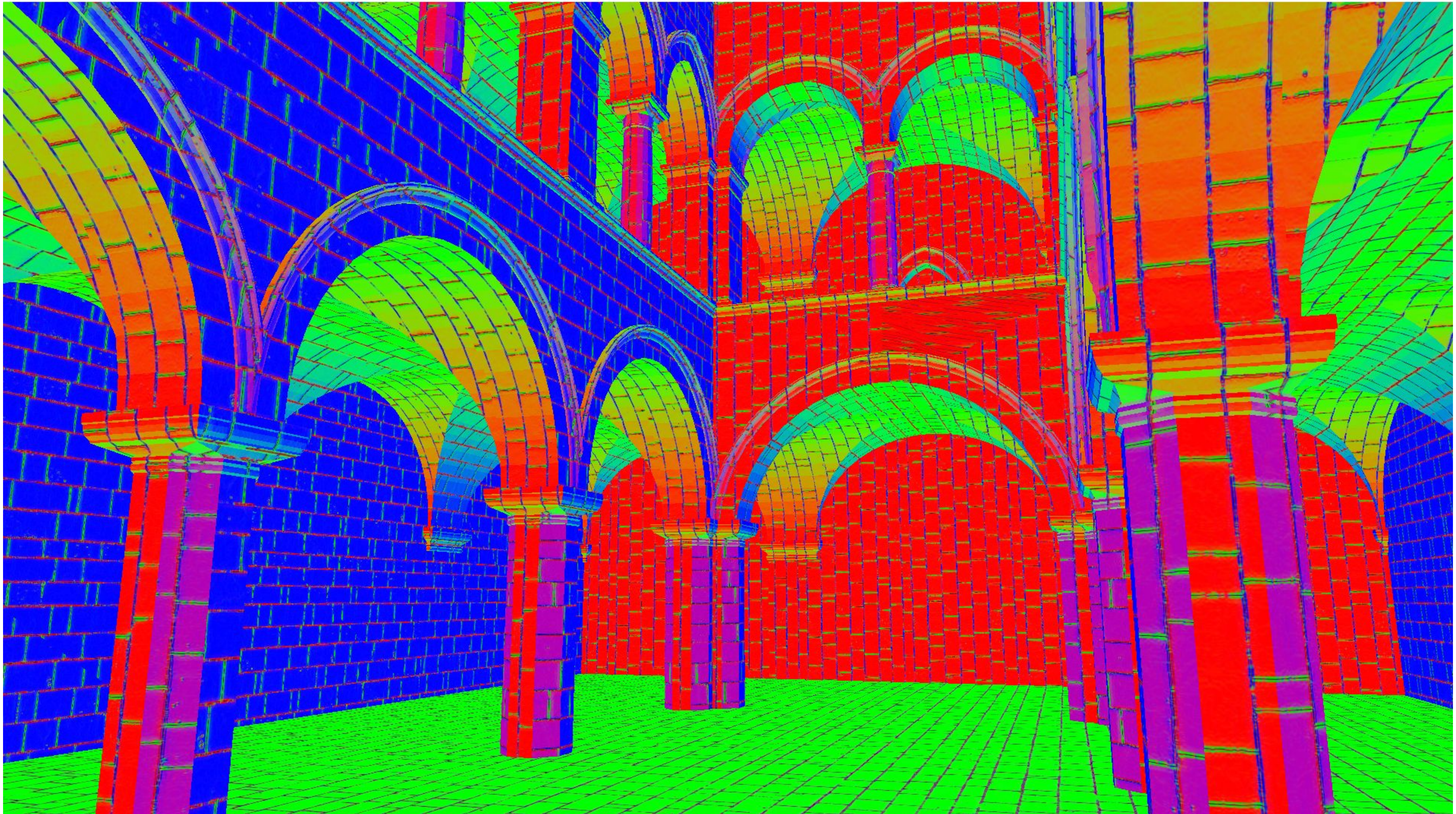
Uniform Buffers

Set	Binding	Buffer	Byte Range	Size
0	0: transform	Buffer 96	0 - 204	4 Variables, 204 bytes
0	3: PointLights	Buffer 98	0 - 32004	2 Variables, 28 bytes needed, 32

Name	Value	Type
light_num	200	int
pointlights		struct...
pointlights[0]		struct
pos	1.95521, 4.80, -4.53677	float3
radius	5.00	float
intensity	0.37396, 0.3716, 0.93365	float3
pointlights[1]		struct
pointlights[2]		struct
pointlights[3]		struct
pointlights[4]		struct
pos	5.00085, -9.80, -8.54775	float3
radius	5.00	float
intensity	0.04311, 0.8098, 0.93826	float3
pointlights[5]		struct
pointlights[6]		struct
pointlights[7]		struct
pointlights[8]		struct
pointlights[9]		struct
pointlights[10]		struct
pointlights[11]		struct
pointlights[12]		struct
pointlights[13]		struct
pointlights[14]		struct
pointlights[15]		struct
pointlights[16]		struct
pointlights[17]		struct
pointlights[18]		struct
pointlights[19]		struct
pointlights[20]		struct
pointlights[21]		struct
pointlights[22]		struct
pointlights[23]		struct
pointlights[24]		struct
pointlights[25]		struct
pointlights[26]		struct


Snipping Tool





Future Milestones

11/28 - 12/2: Tile based + Light Culling Done

12/2 - 12/11: all other works.

- Optimization & comparision;
- Possibly add more fancy features to our renderer: material variety, bloom, SSAO, MSAA, TAA, etc.

References

Vulkan Tutorial: https://vulkan-tutorial.com/

Vulkan examples and demos from Sascha Willems:
https://github.com/SaschaWillems/Vulkan

Forward vs Deferred vs Forward+ Rendering: http://www.3dgep.com/forward-plus/

Forward+: Bringing Deferred Lighting to the Next Level:
https://takahiroharada.files.wordpress.com/2015/04/forward_plus.pdf