For the Alliance! World of Warcraft and Intel Discuss an Optimized Azeroth

Max Rahm, World of Warcraft Engine Team
Laura Reznikov, Intel

Legal Notices and Disclaimers

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at [intel.com].

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit www.intel.com/benchmarks.

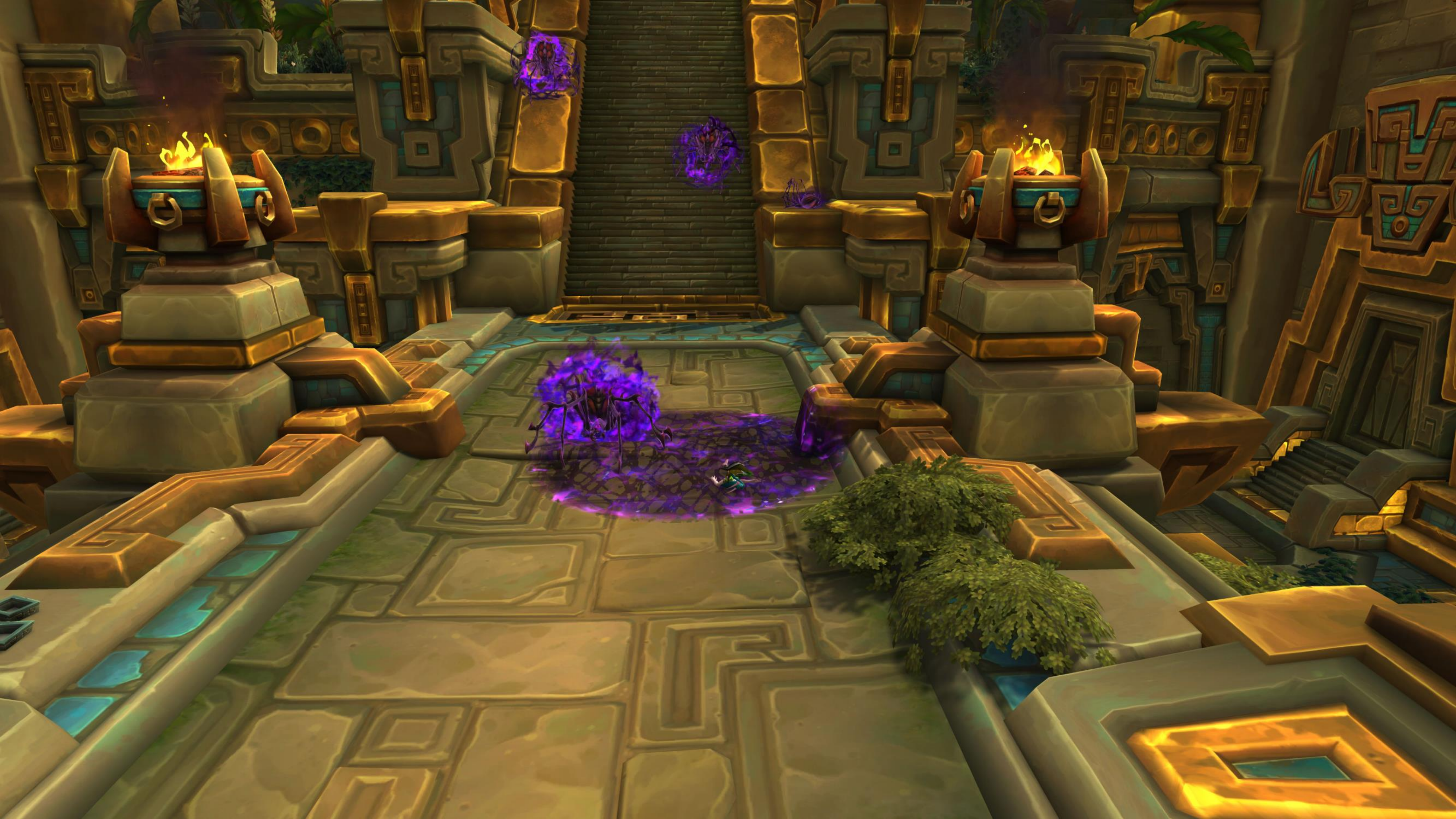
Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Intel, Core and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others

Toxic Saurid



```
HRESULT hr;
ID3D12PipelineState* pso = nullptr;
if (FAILED(hr = m_device->CreateGraphicsPipelineState(&psoDesc, IID_PPV_ARGS(&pso))))
{
    FatalDeviceError(hr, "CreateGraphicsPipelineState failed");
}
```

What we’ve done with Direct X 12 (and Metal) so far

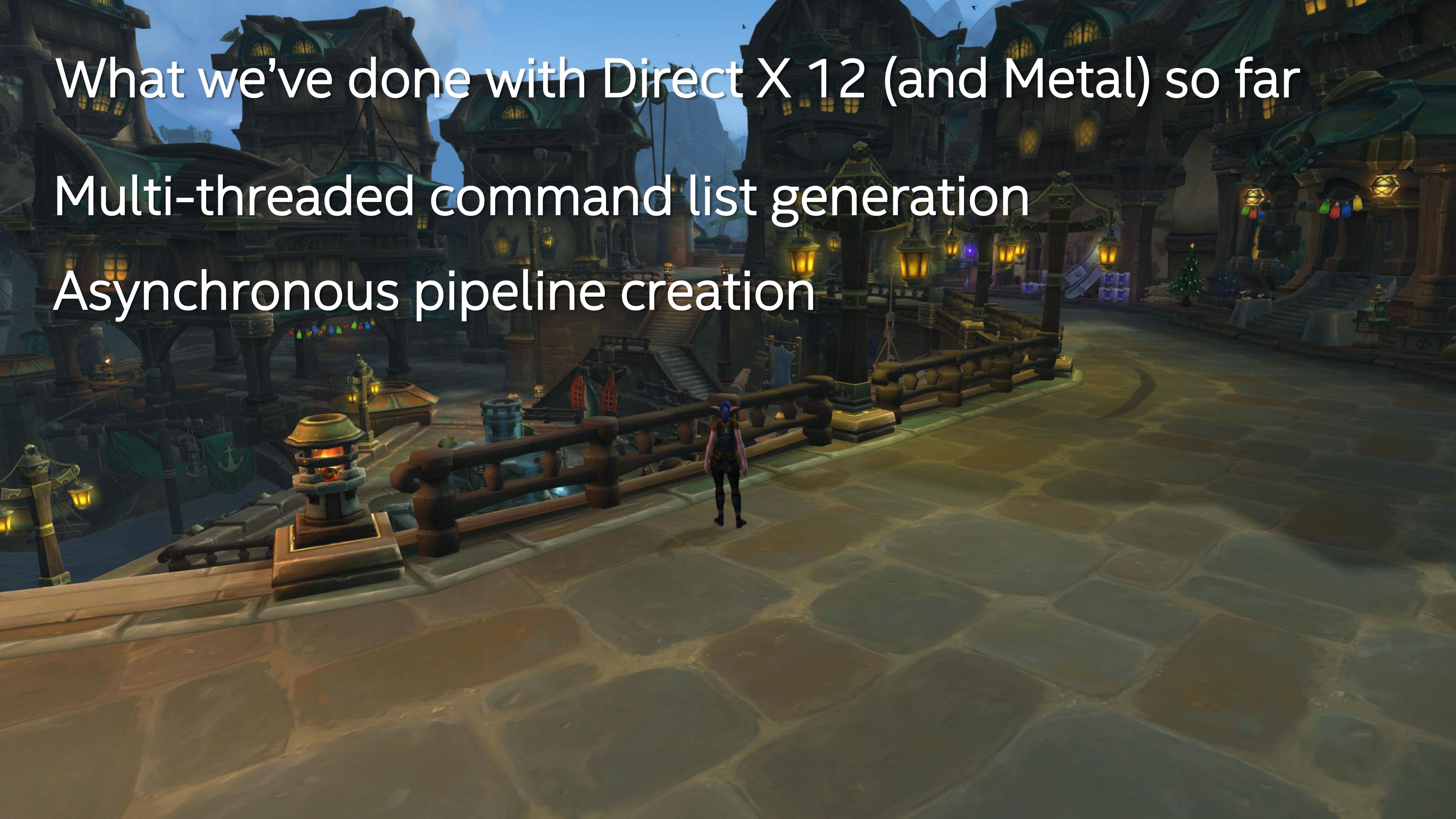
What we've done with Direct X 12 (and Metal) so far

Multi-threaded command list generation

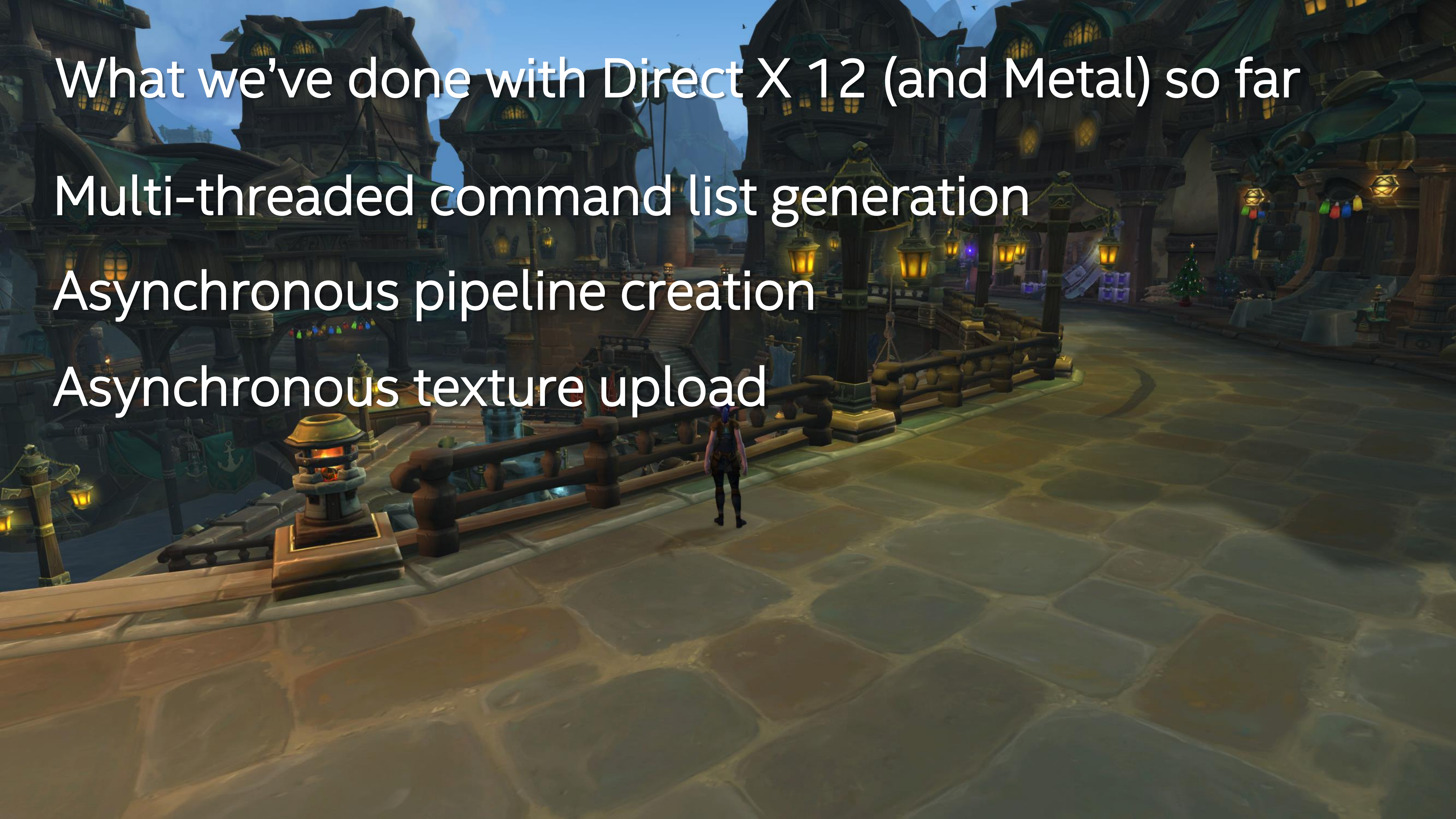
What we’ve done with Direct X 12 (and Metal) so far

Multi-threaded command list generation

Asynchronous pipeline creation



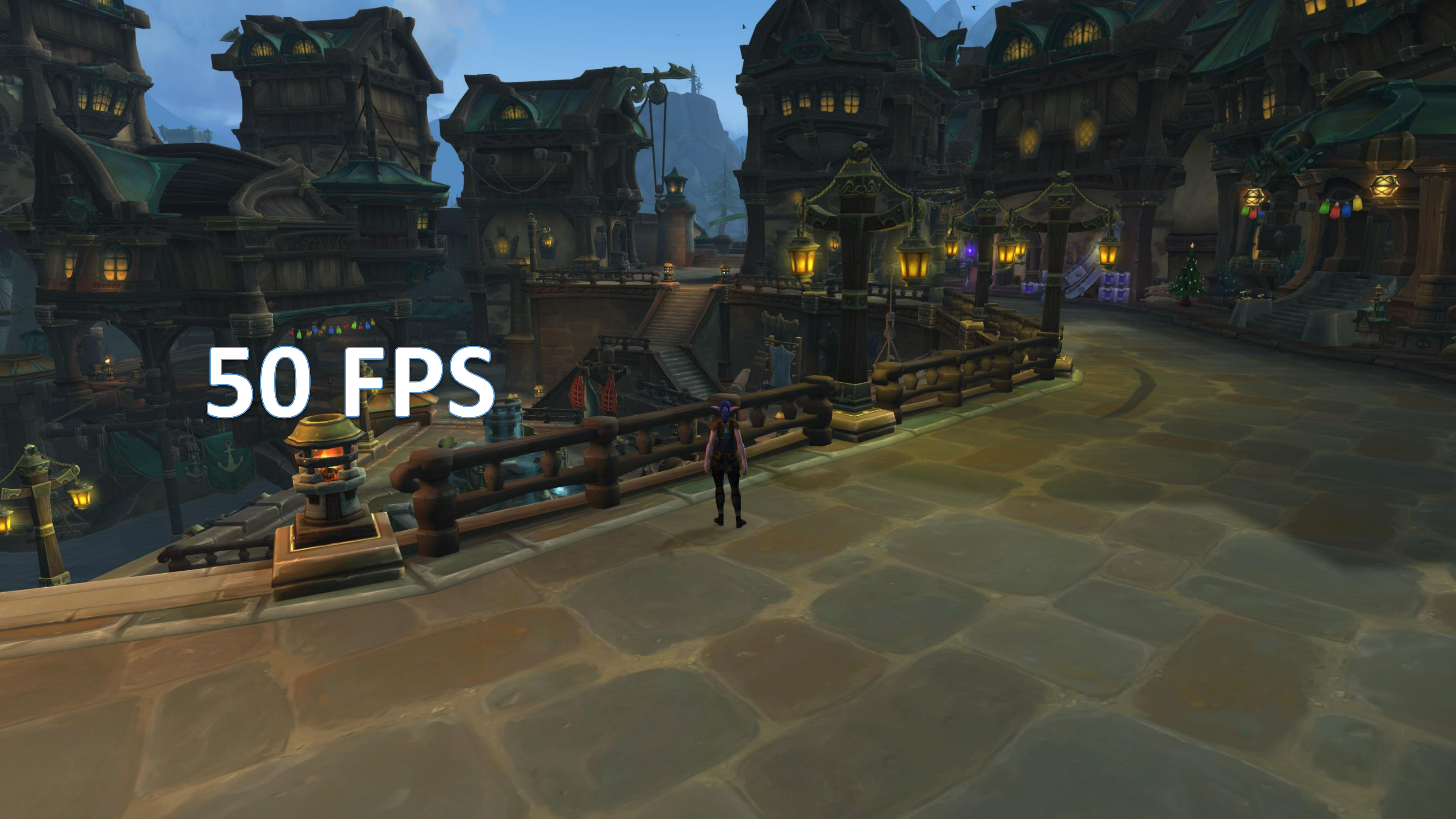
What we’ve done with Direct X 12 (and Metal) so far



Multi-threaded command list generation

Asynchronous pipeline creation

Asynchronous texture upload



50 FPS

50 FPS → 80 FPS

Variable Rate Shading

What is Variable Rate Shading (VRS)?

A hardware feature to reduce pixel shader work.

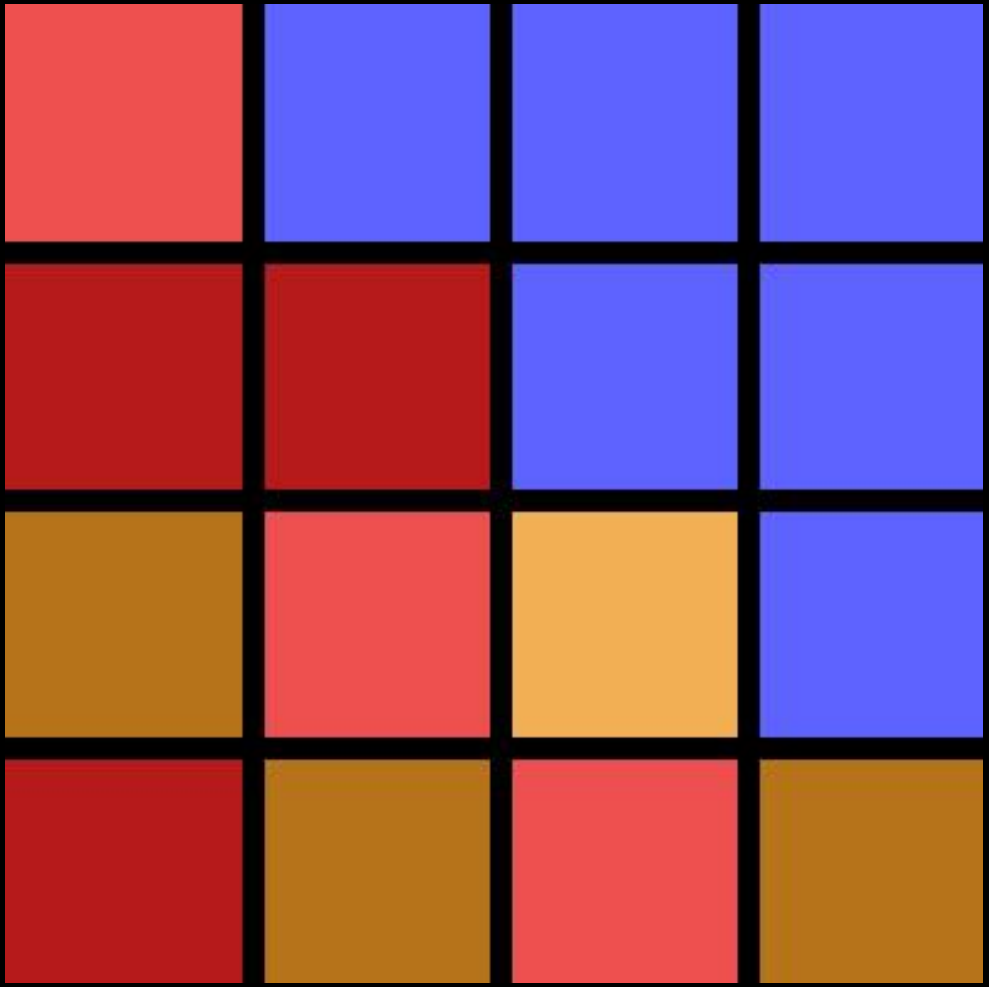
It works by doing pixel shader invocation over pixel groups rather than per pixel.

It can be thought as an extension to MSAA and like an LOD at shading rate

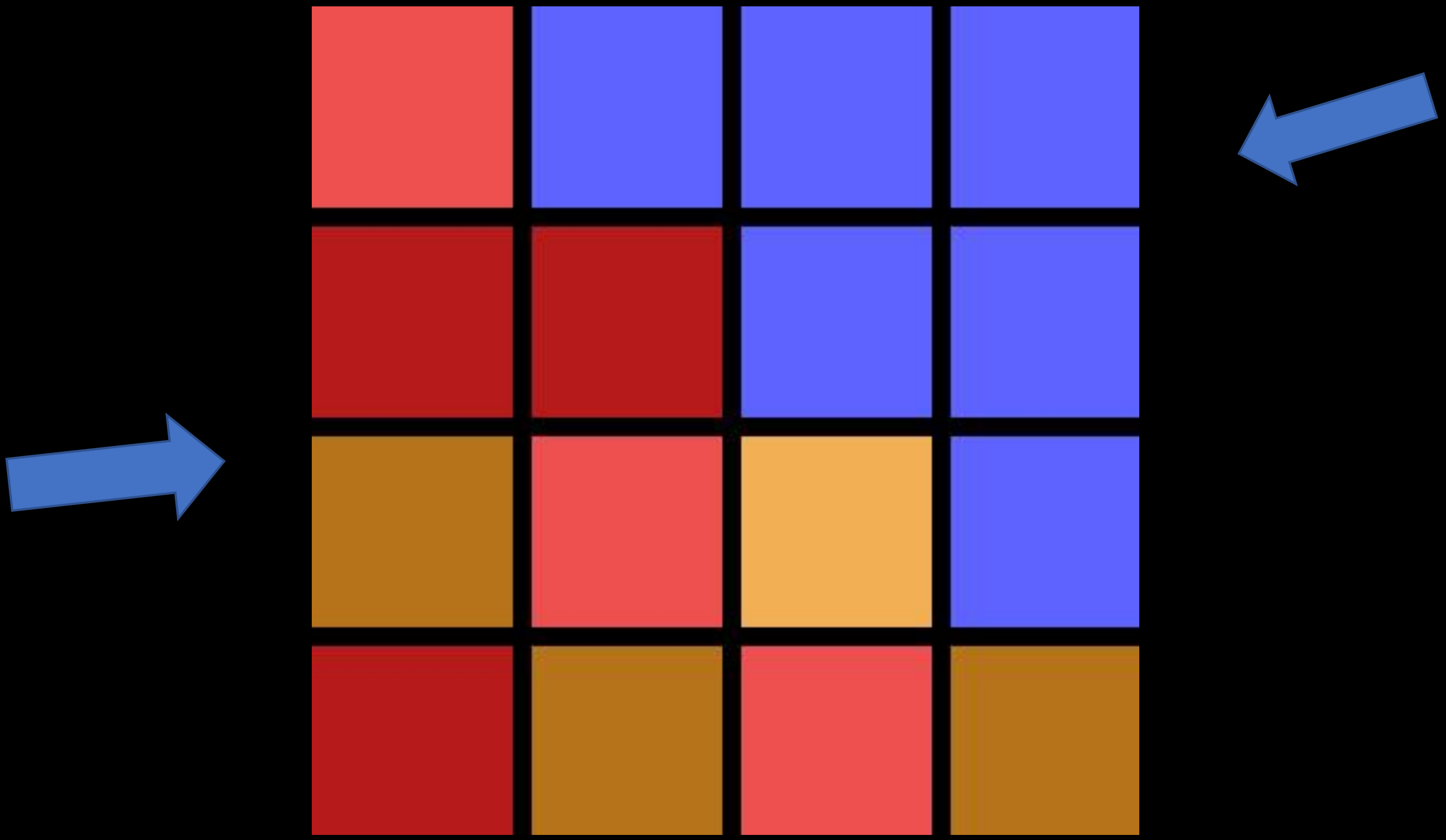
Controlling VRS

- Per draw call
- By screen space mask
- From vertex/geometry shader

This is an apple



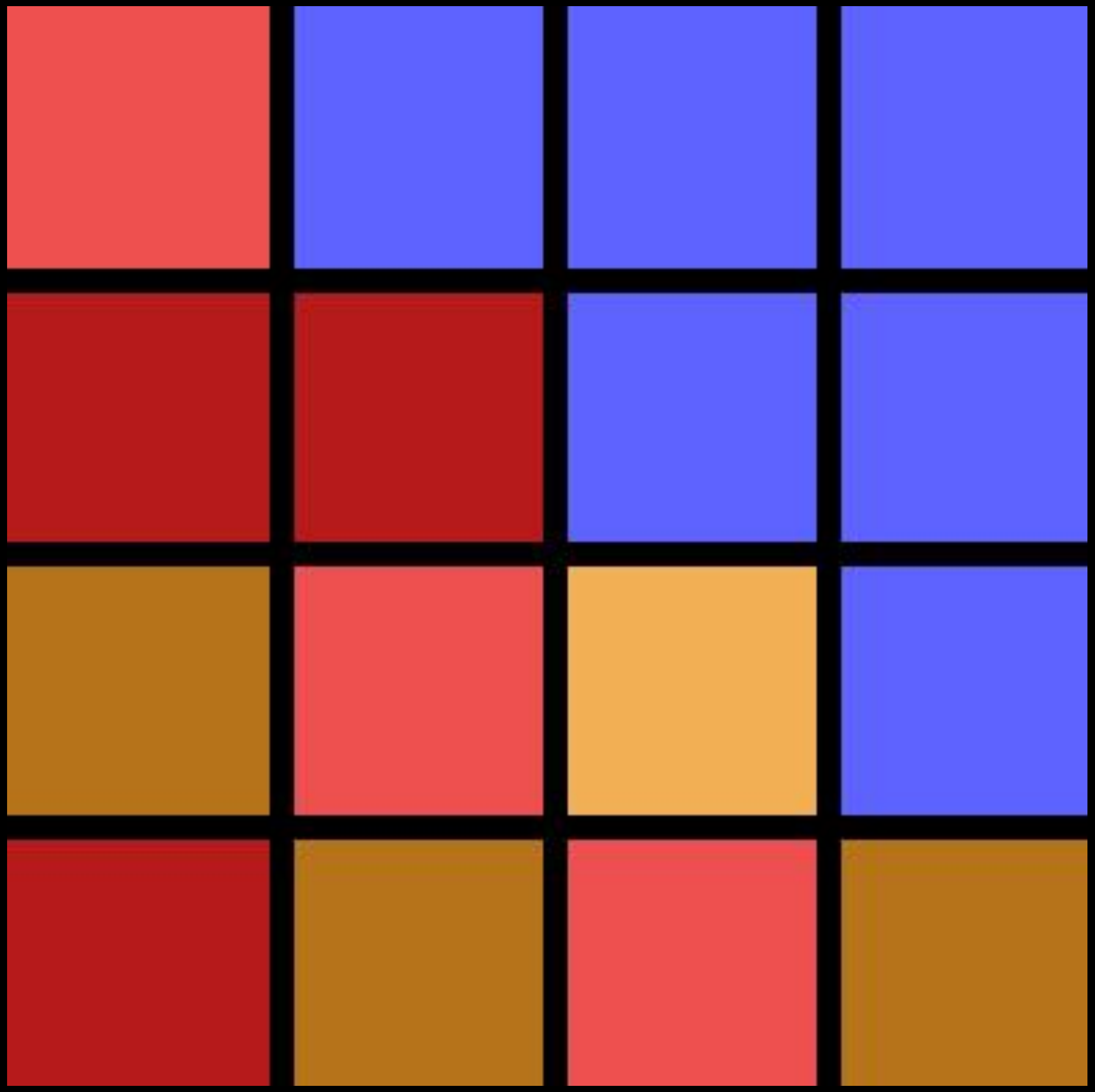
This is an apple



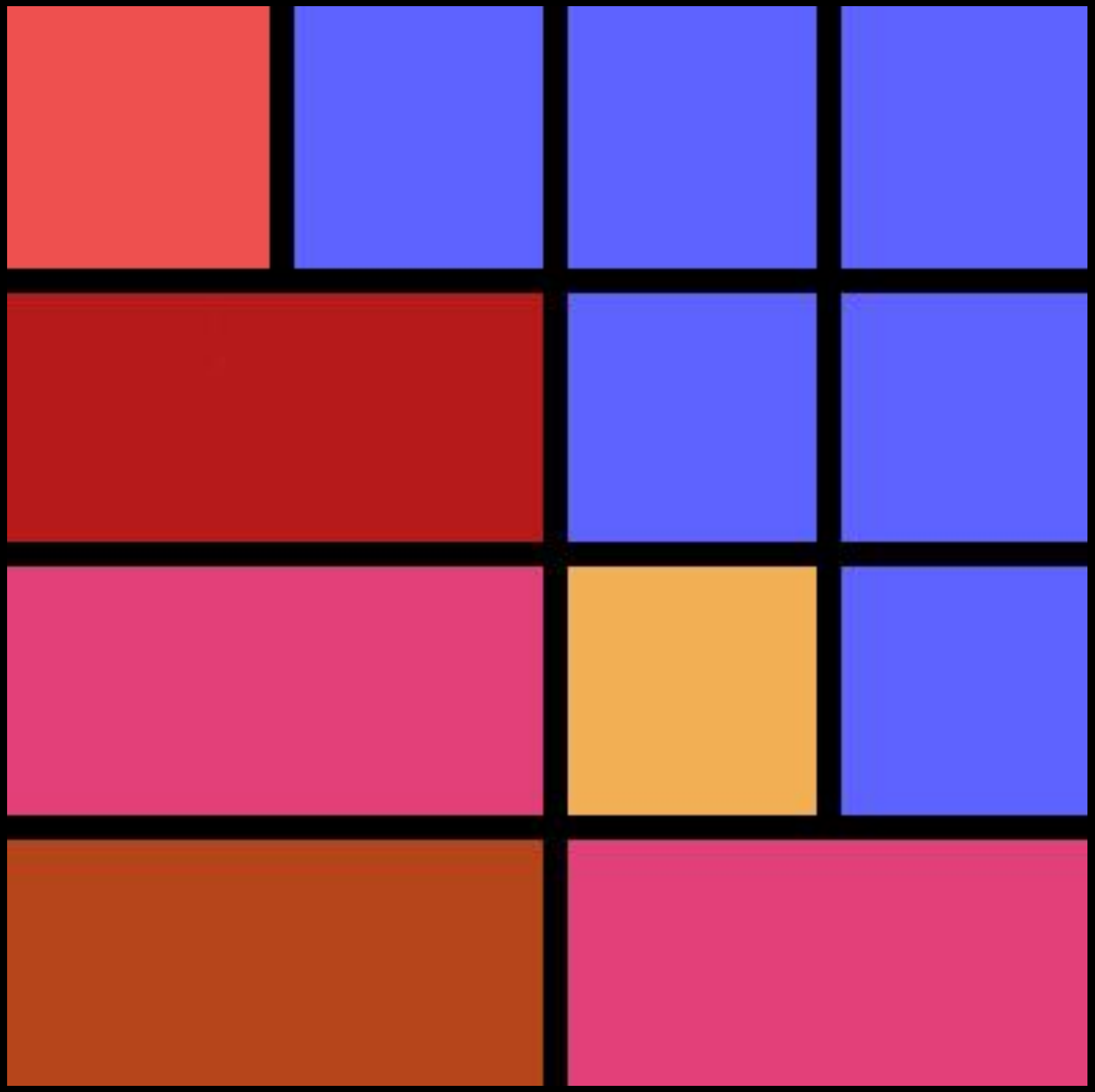
Sky

Apple

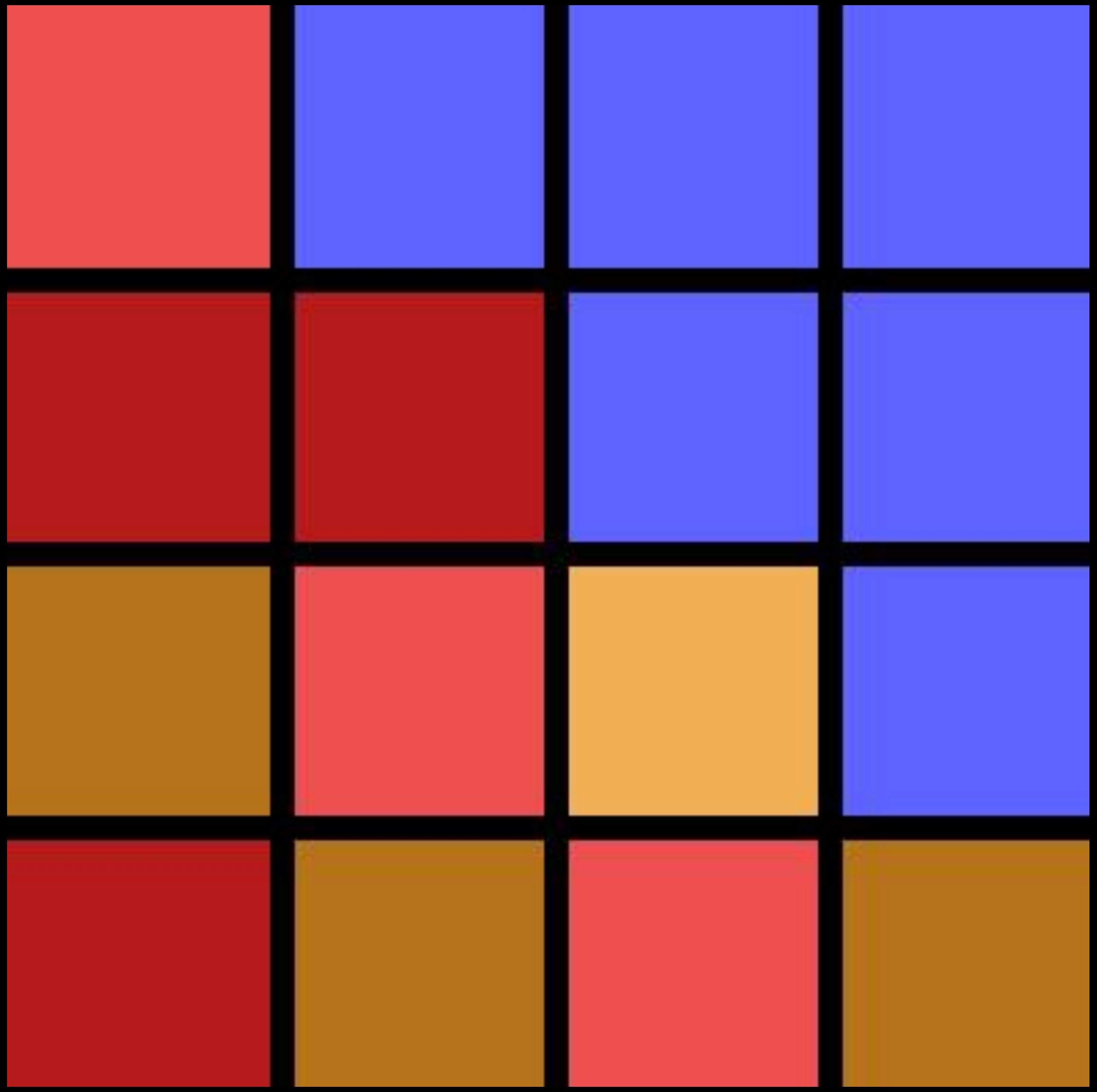
1x1 (normal pixel shading)



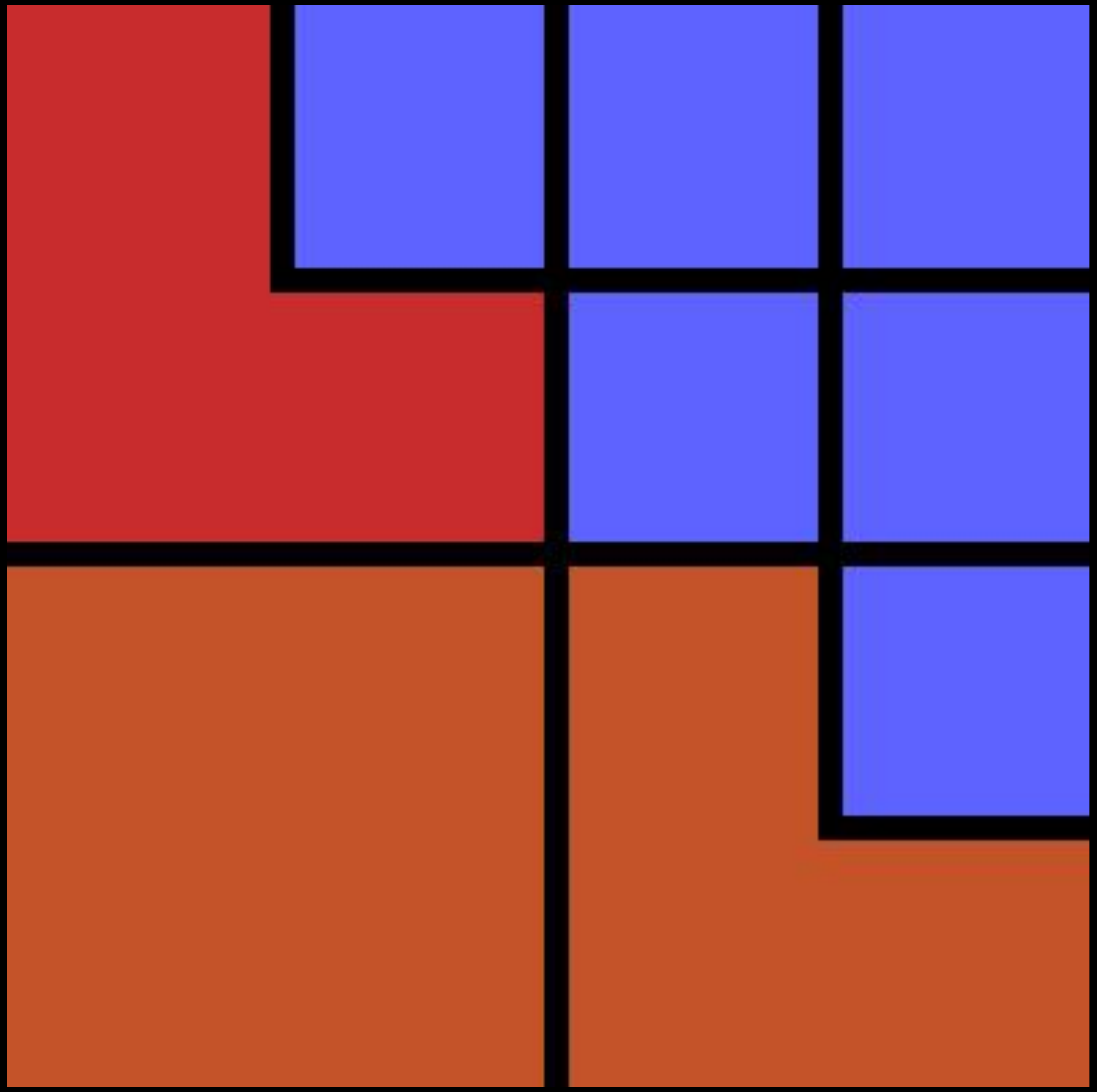
2x1 (variable rate shading)



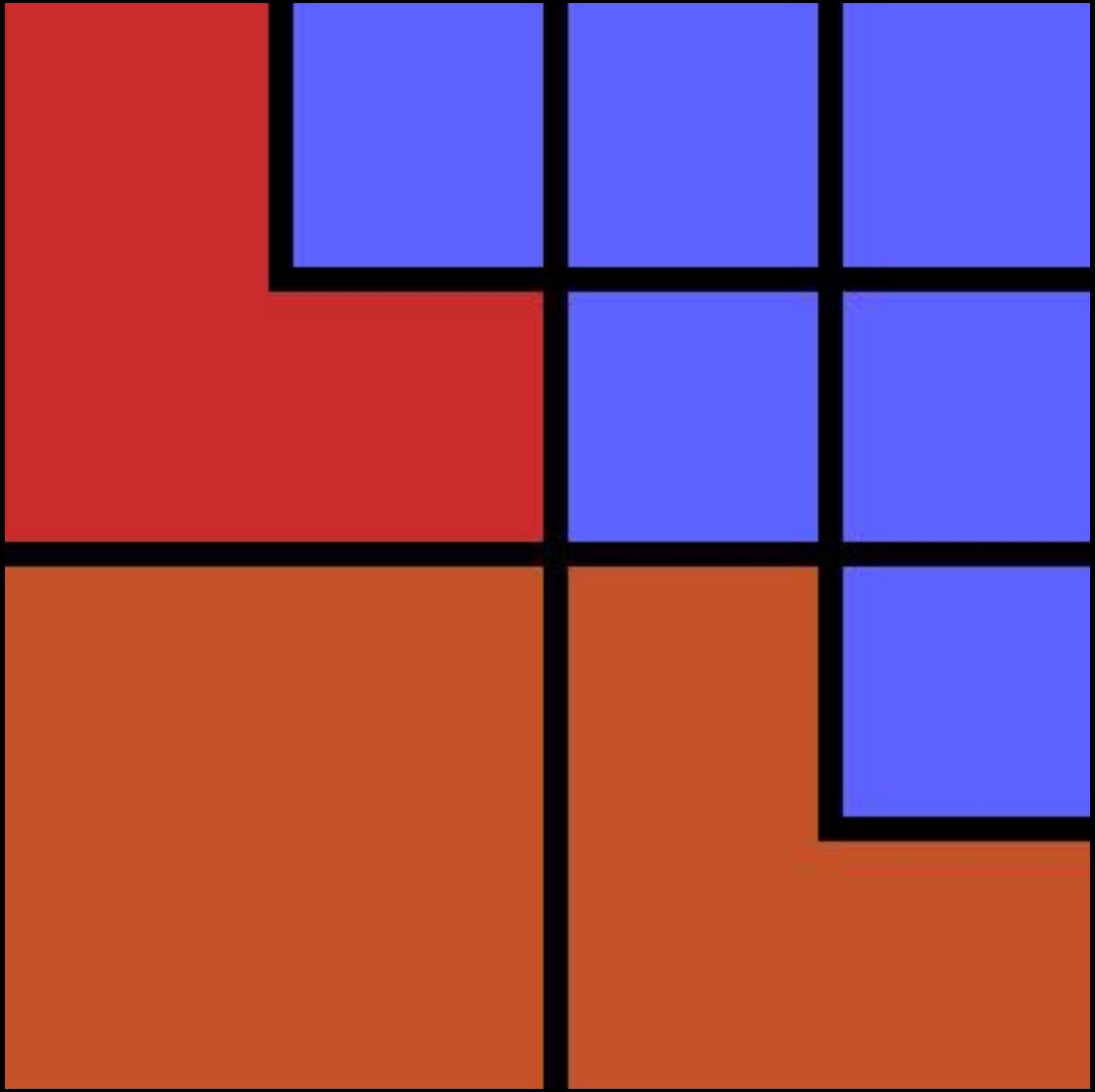
1x1 (normal pixel shading)



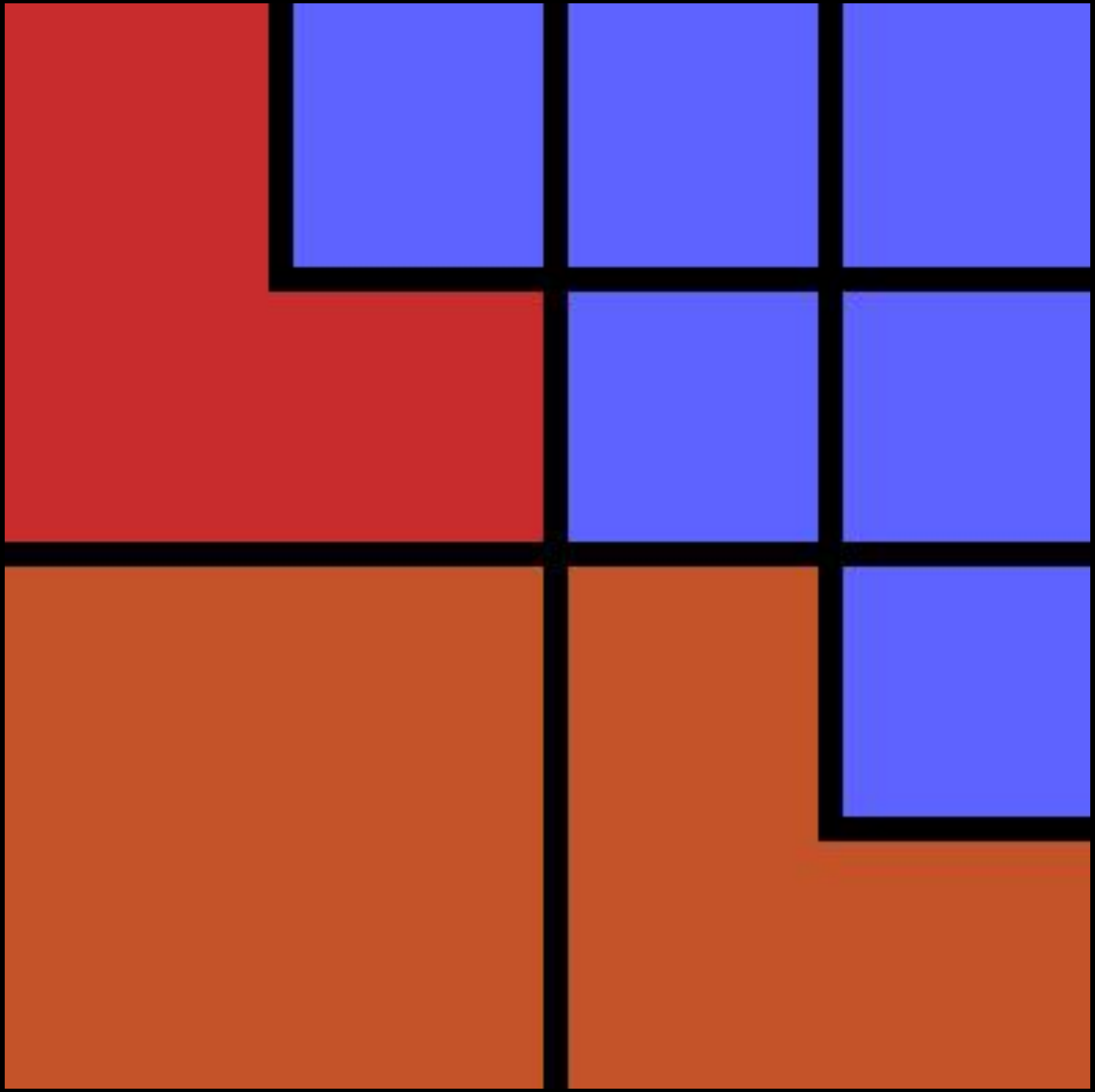
2x2 (variable rate shading)



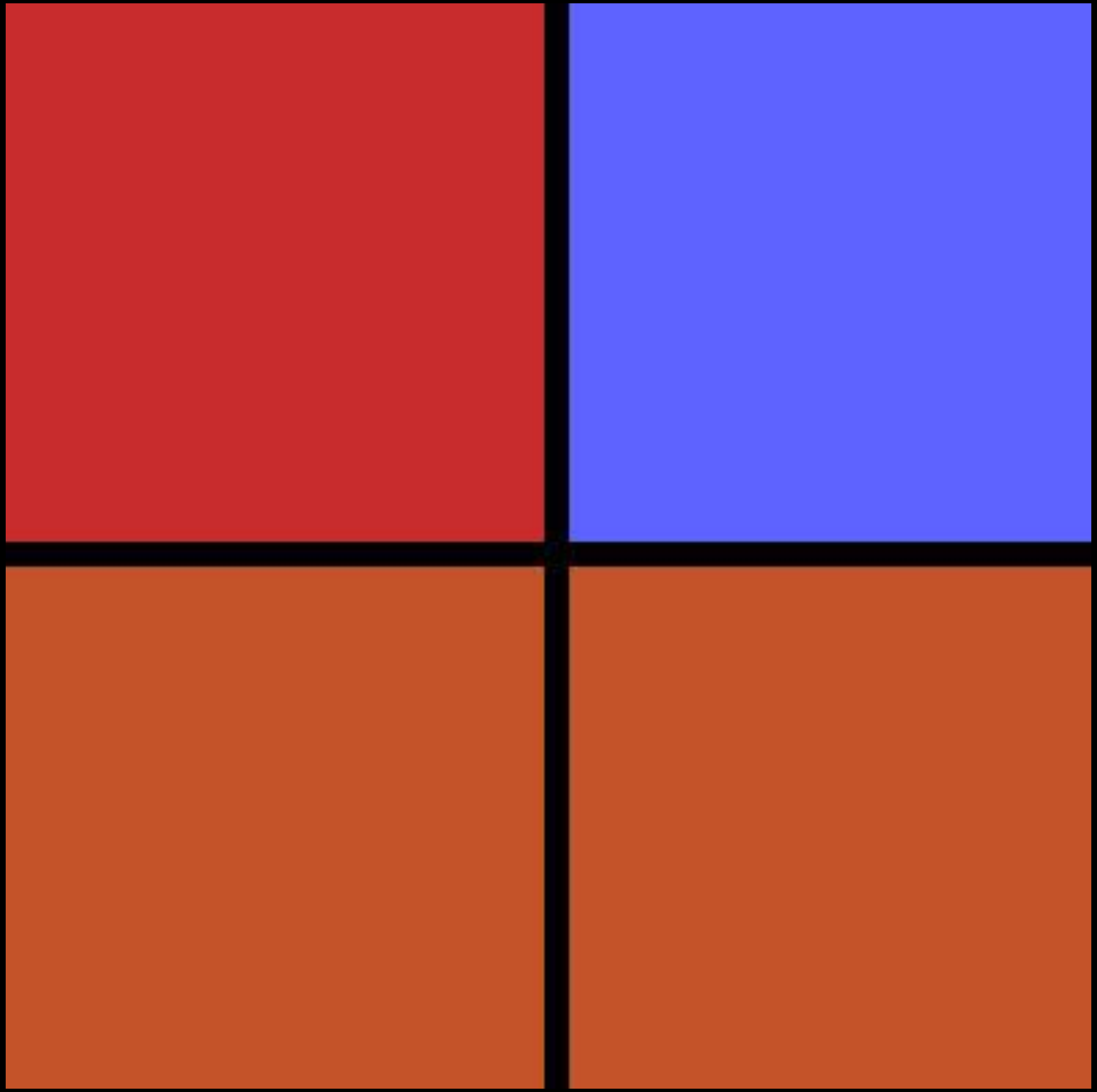
2x2 (variable rate shading)

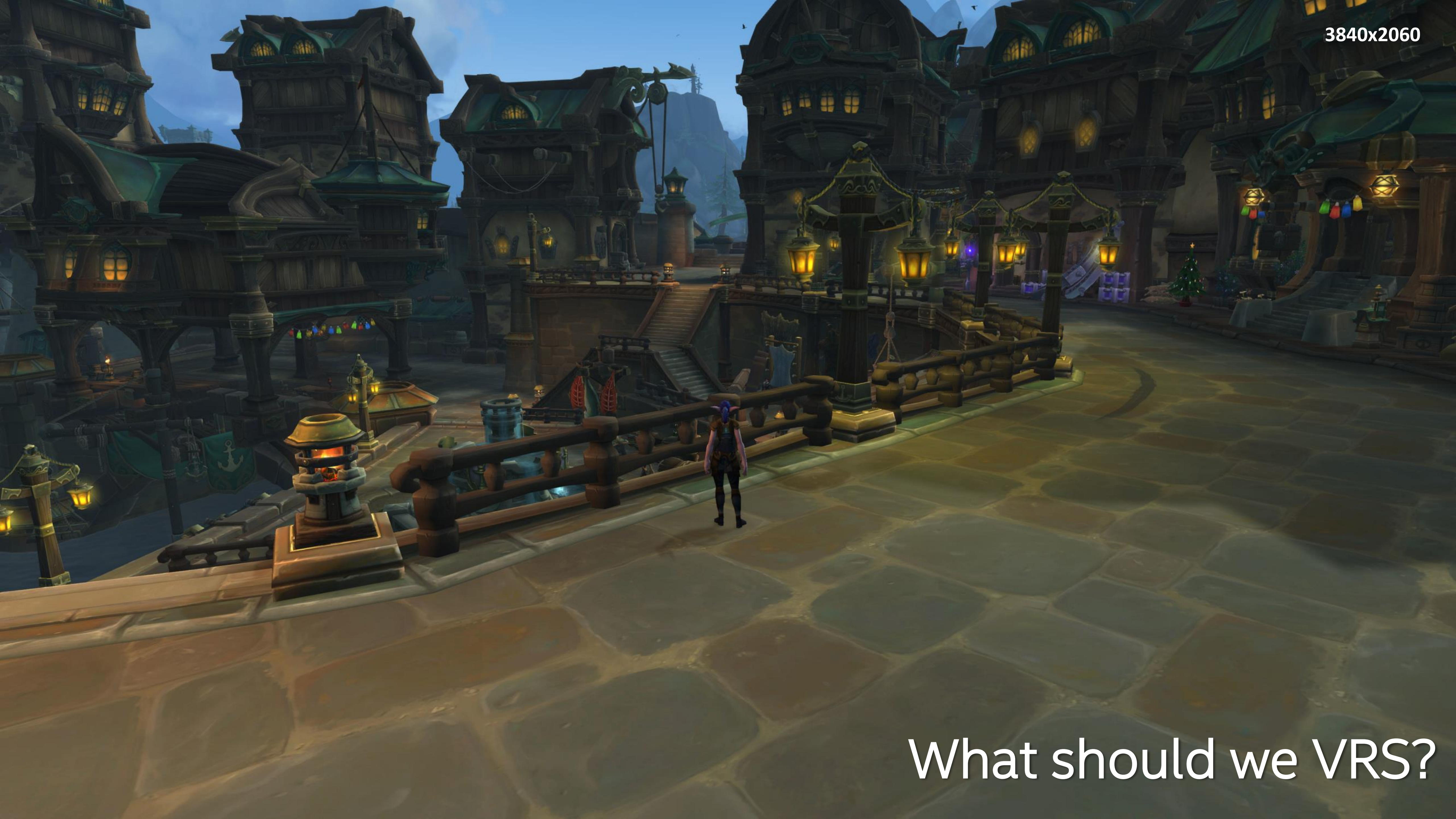


2x2 (variable rate shading)



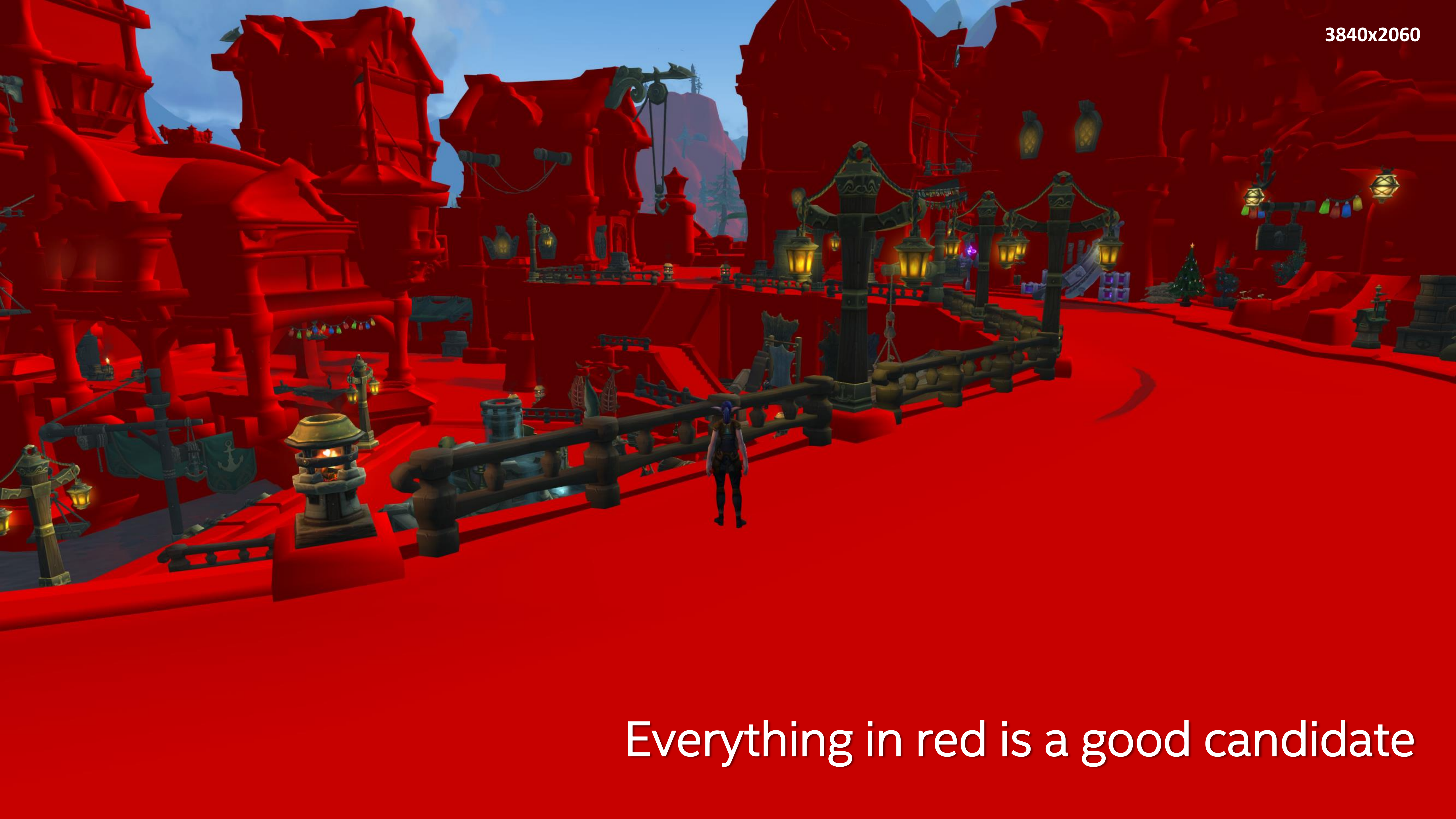
50% render scale





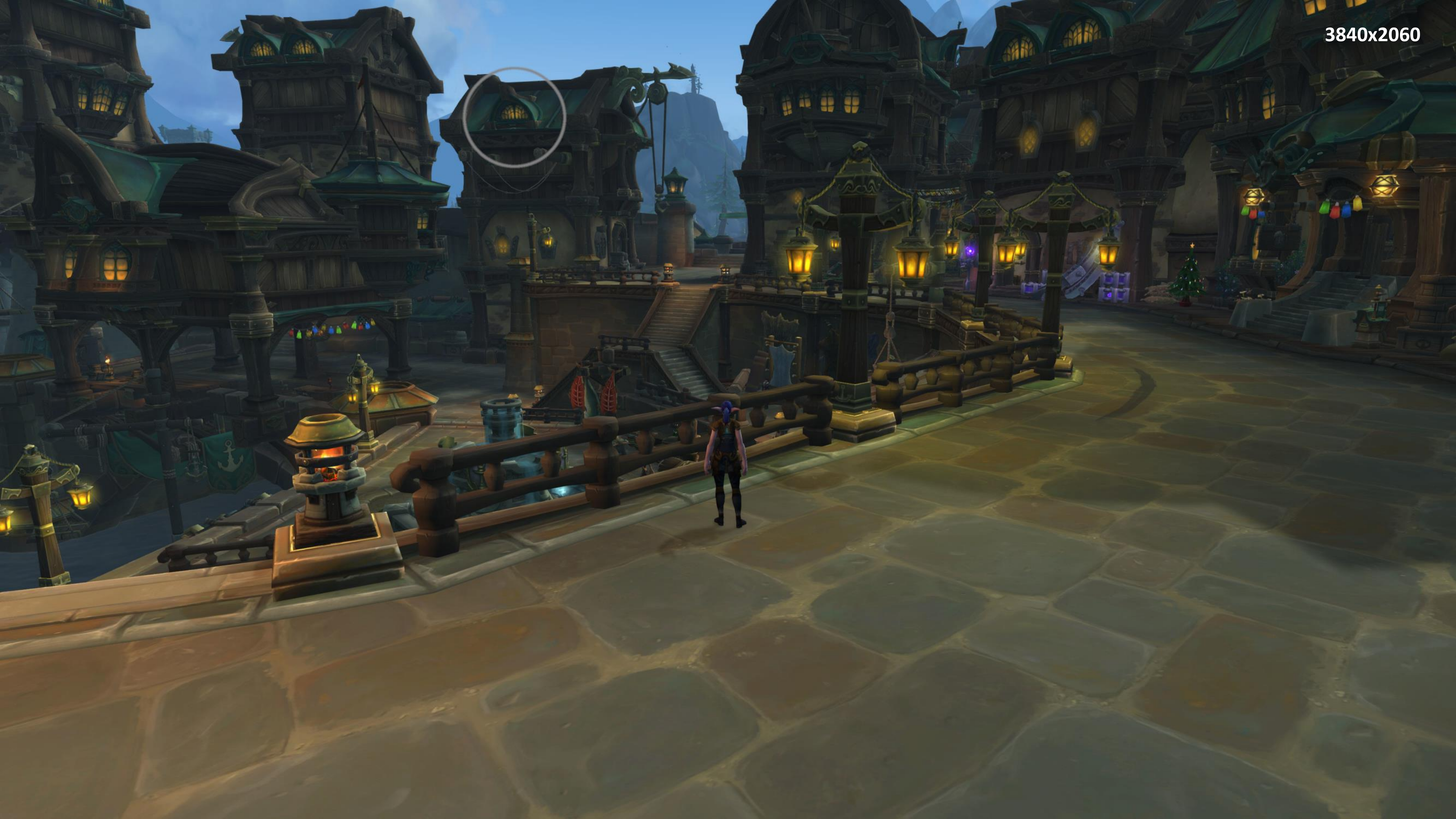
3840x2060

What should we VRS?

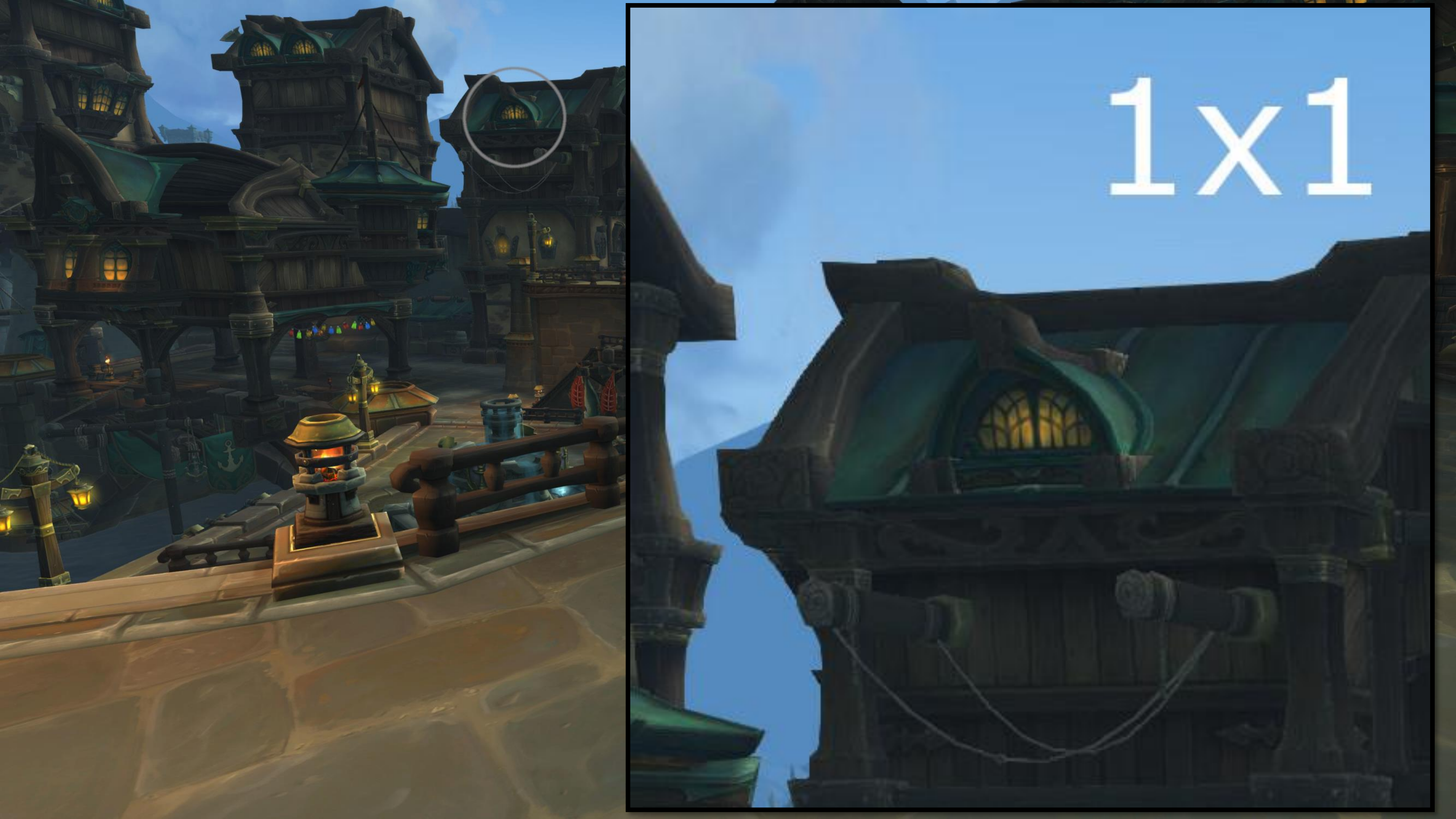


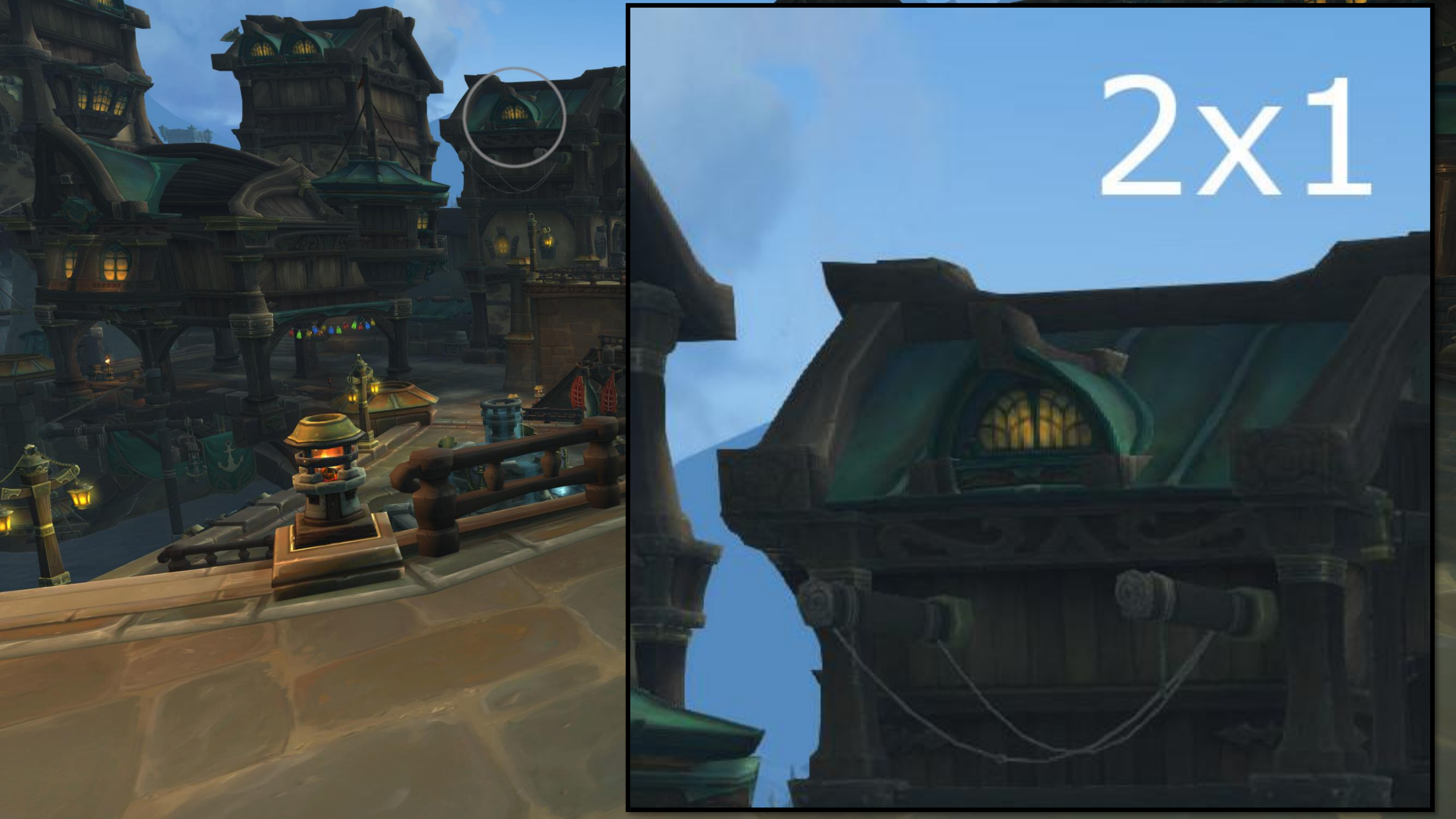
3840x2060

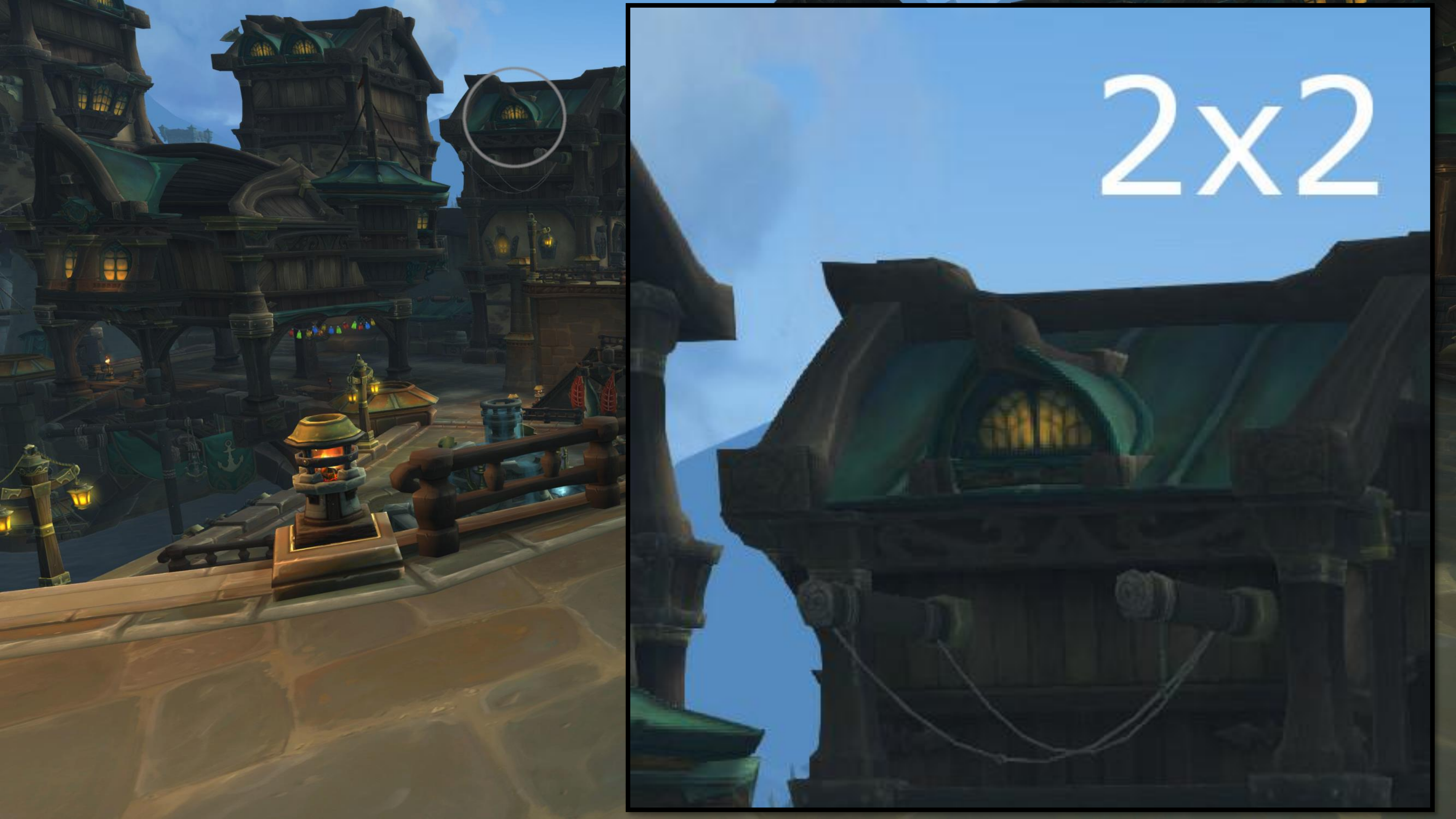
Everything in red is a good candidate



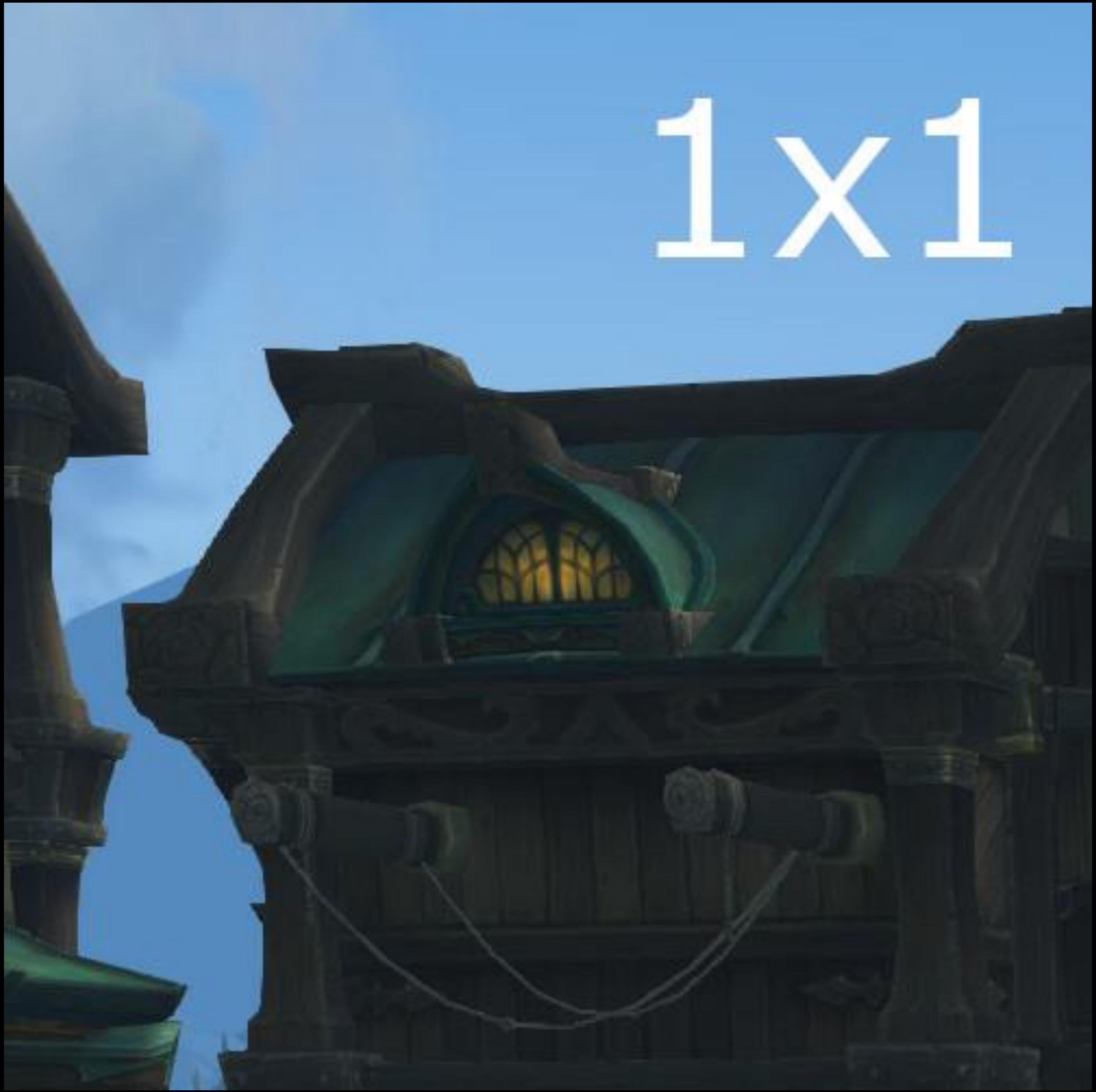
3840x2060

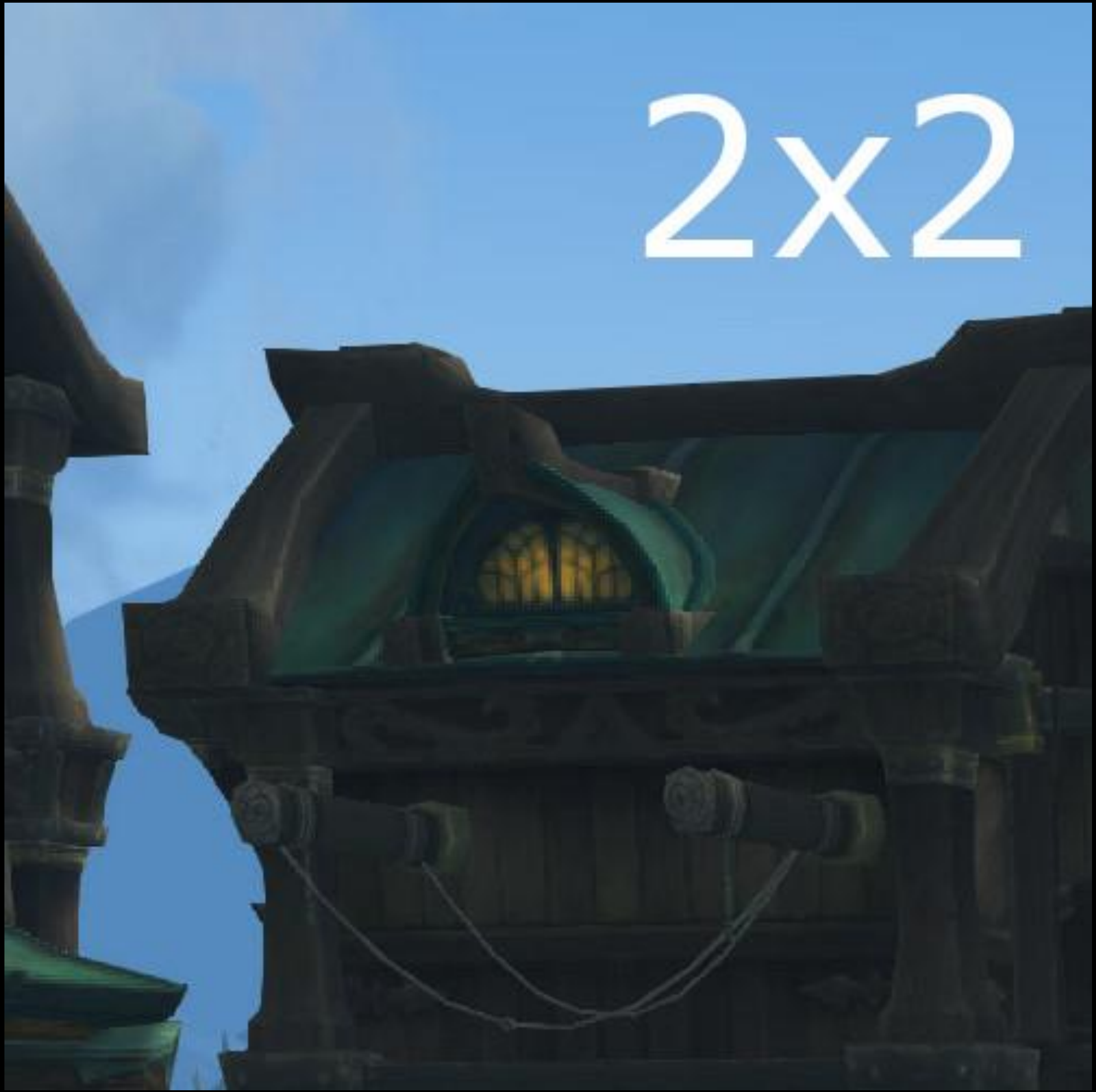

1x1

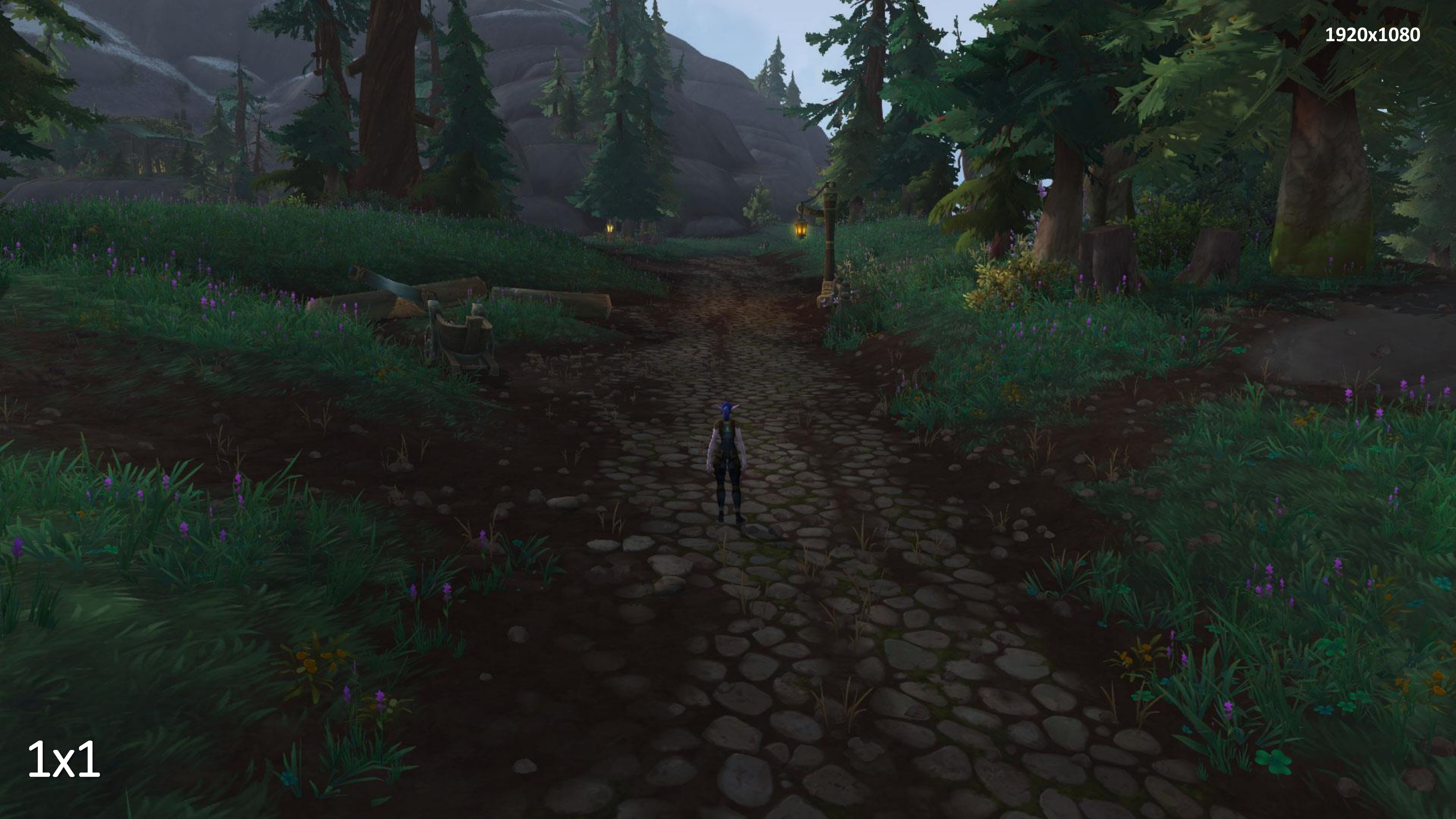

2x1


2x2

Silhouette stays the same

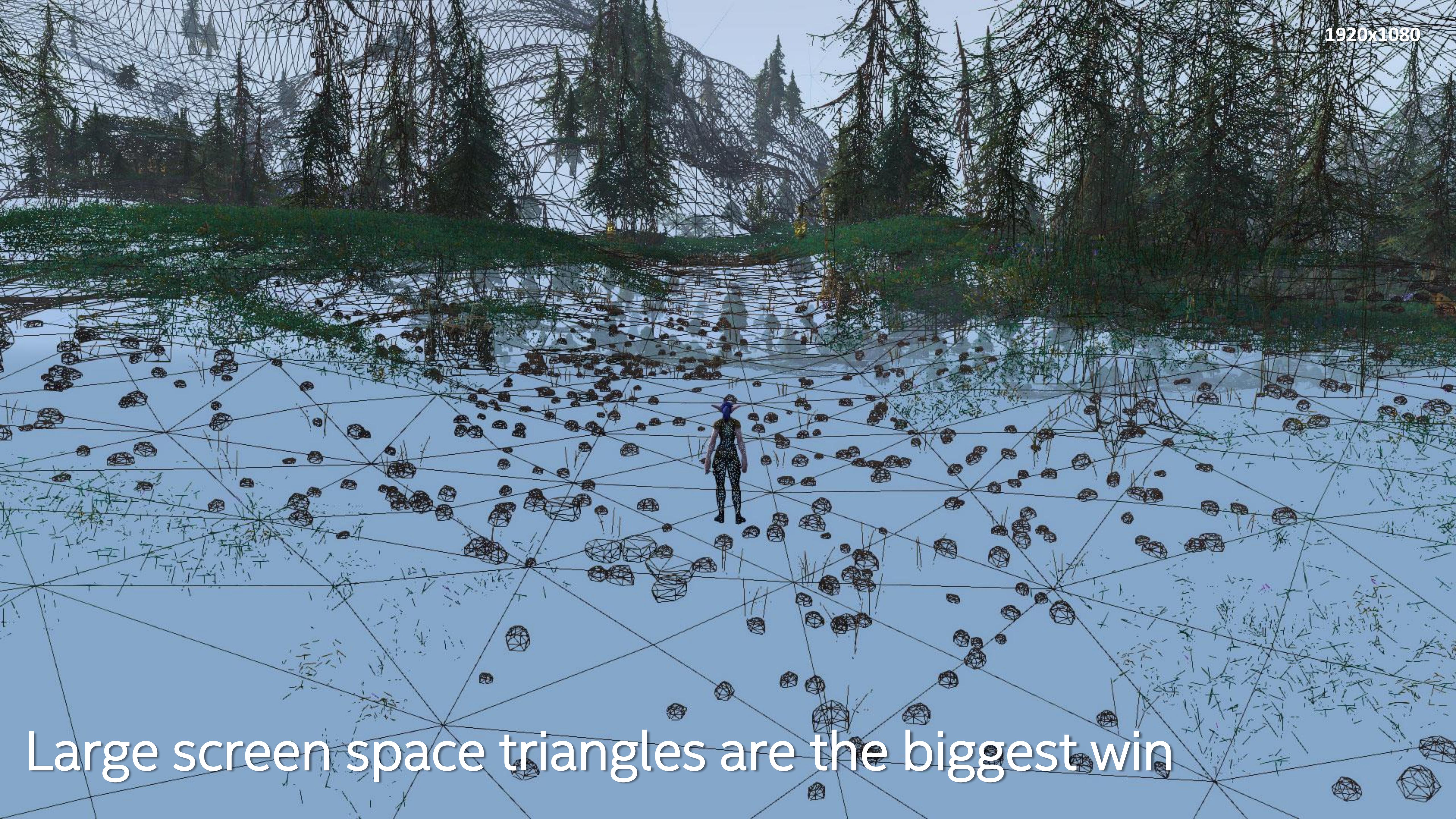




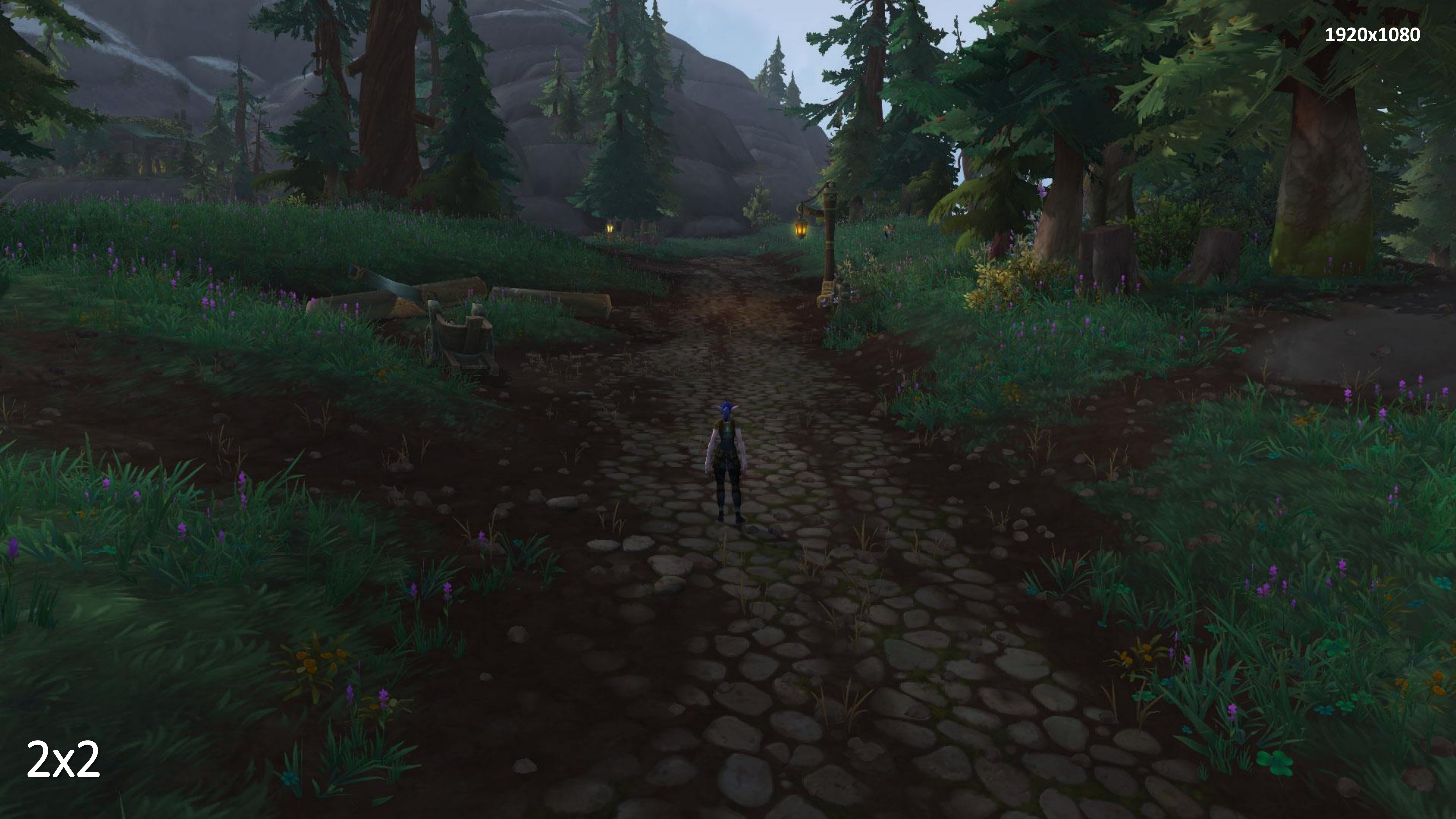


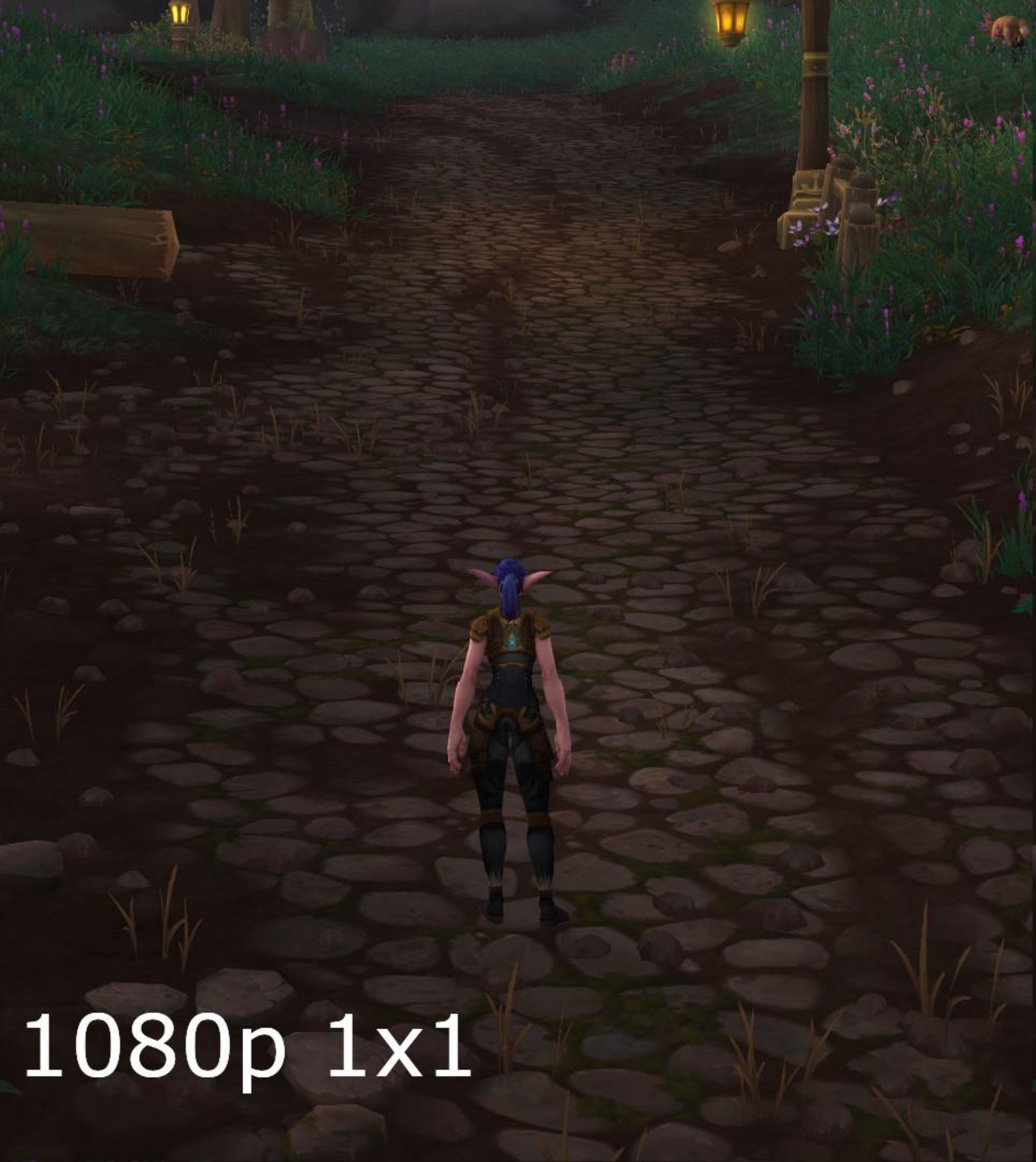
1920x1080

1x1

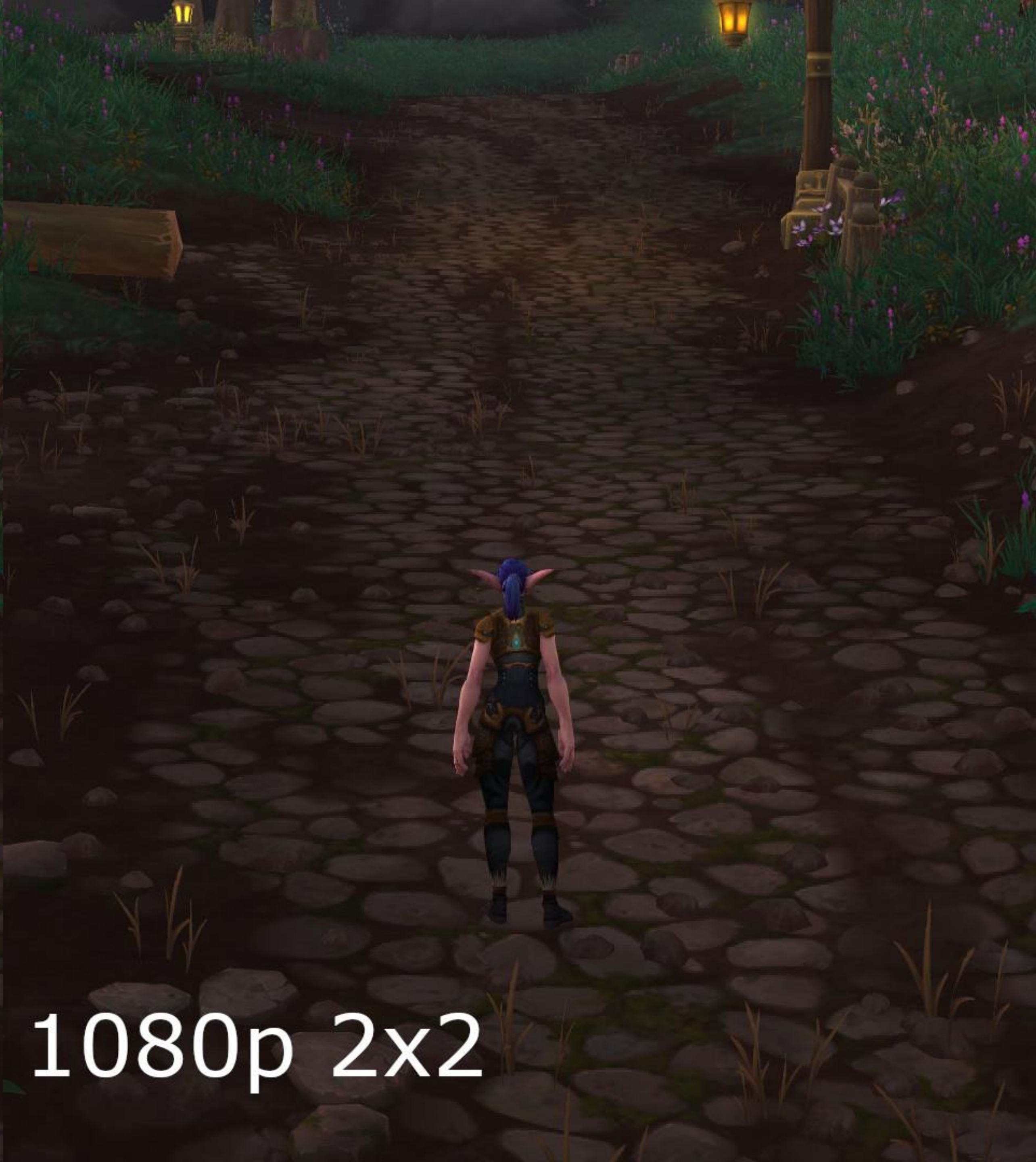


Large screen space triangles are the biggest win

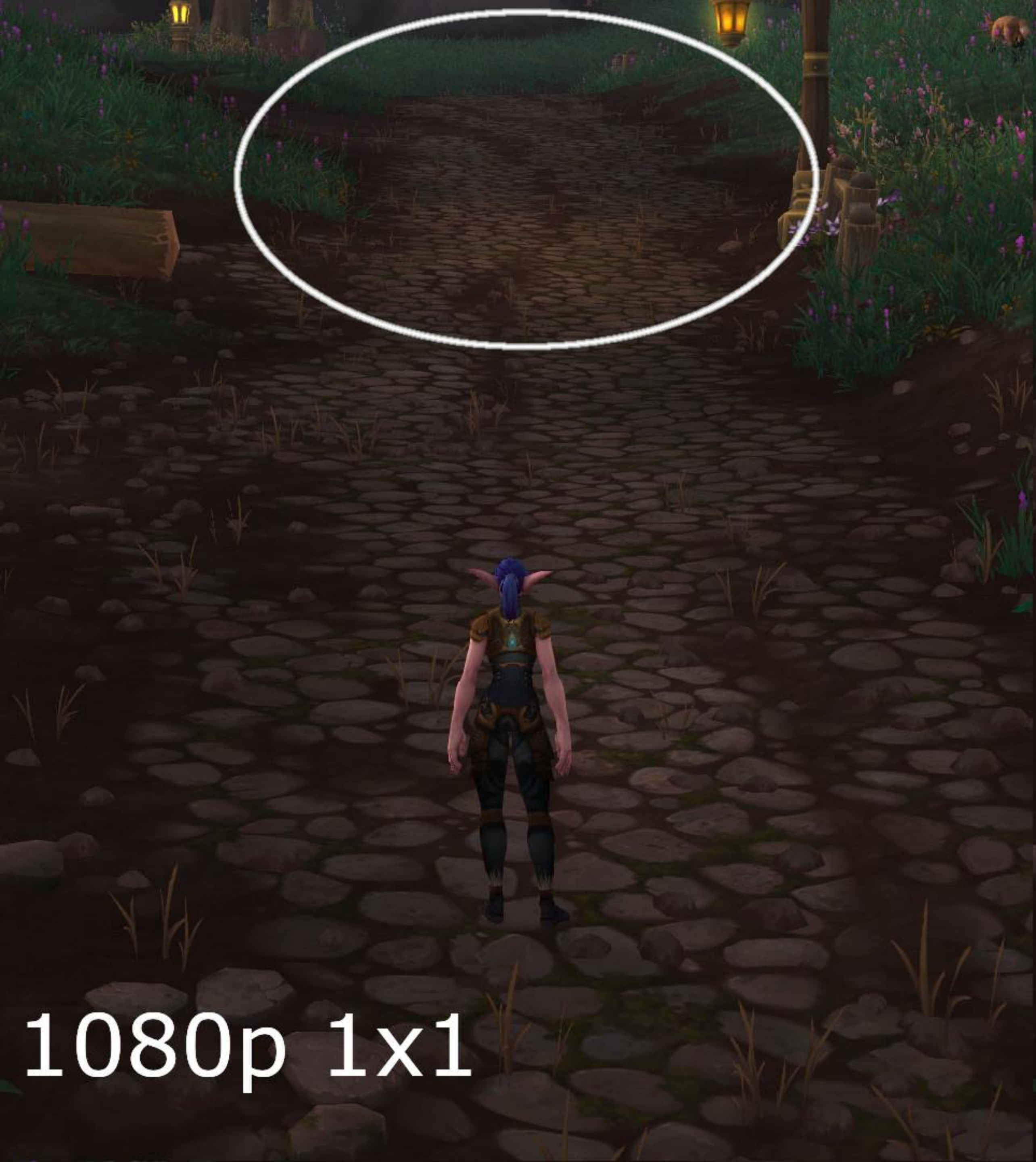

1920x1080
2x2



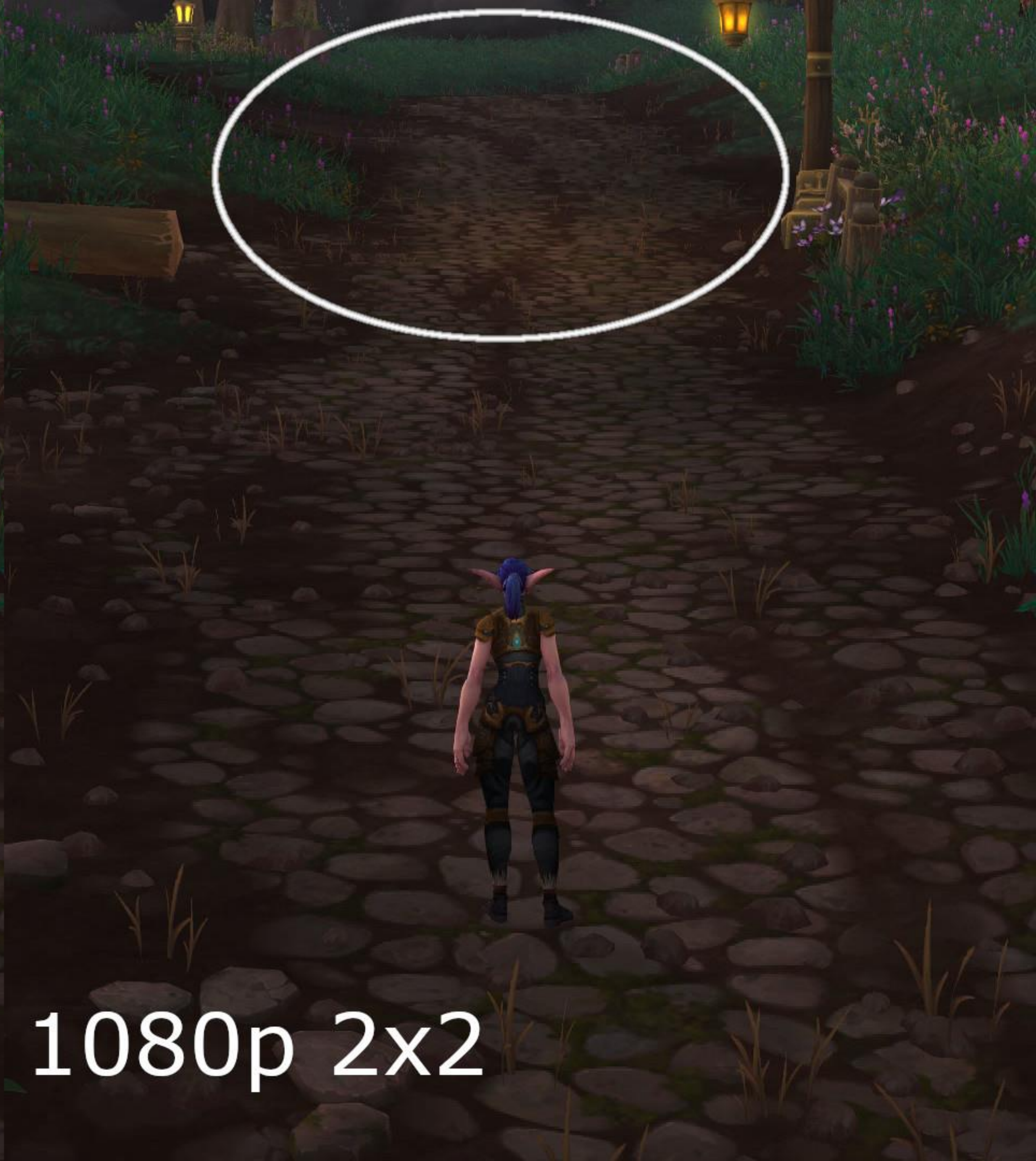
1080p 1x1



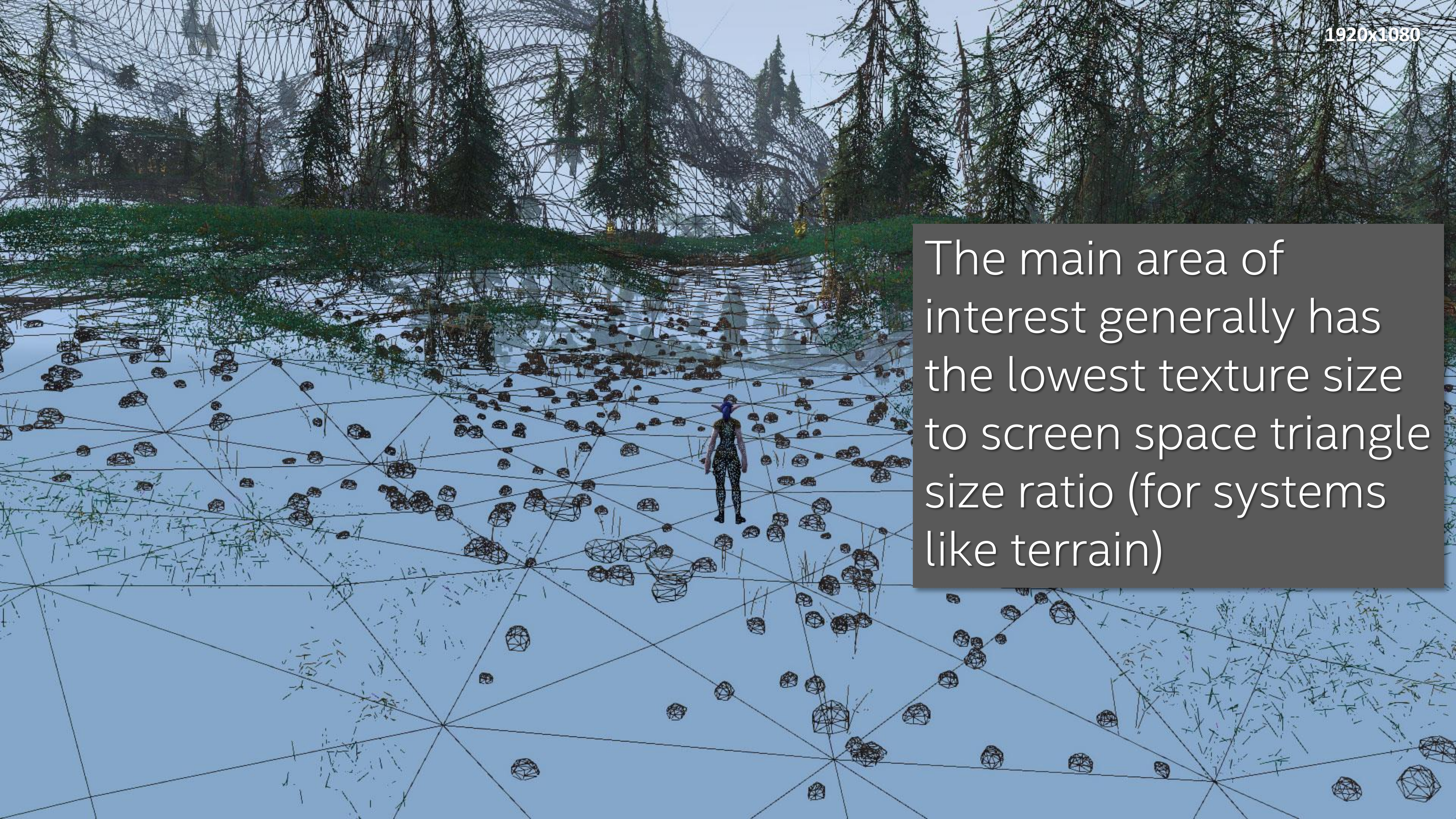
1080p 2x2



1080p 1x1

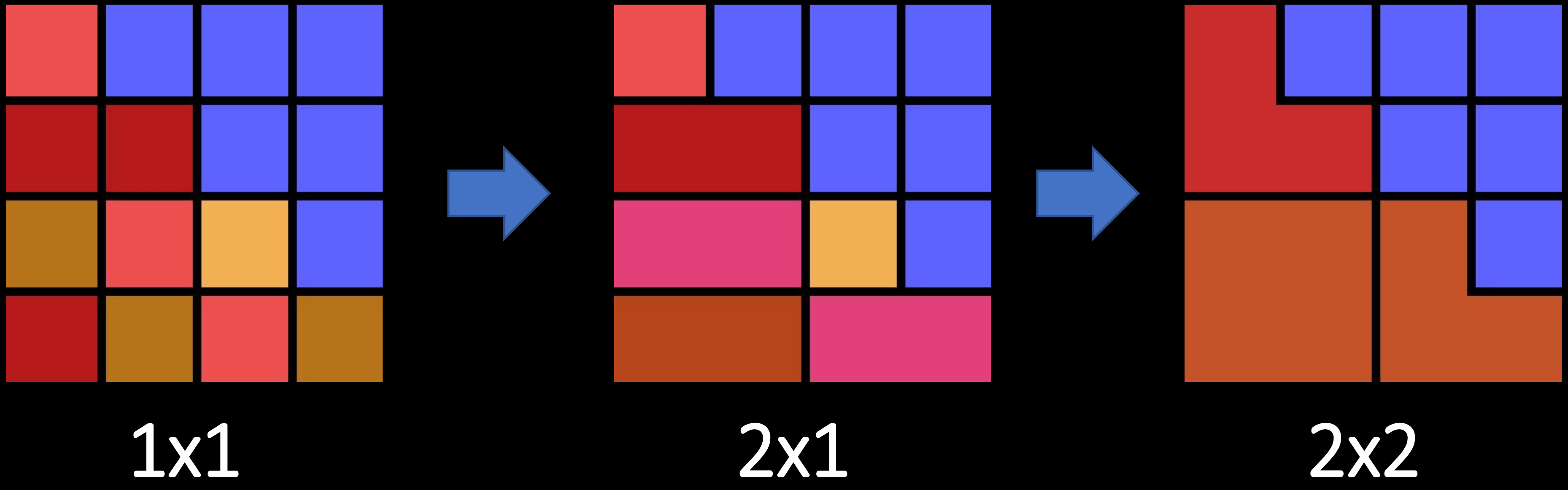


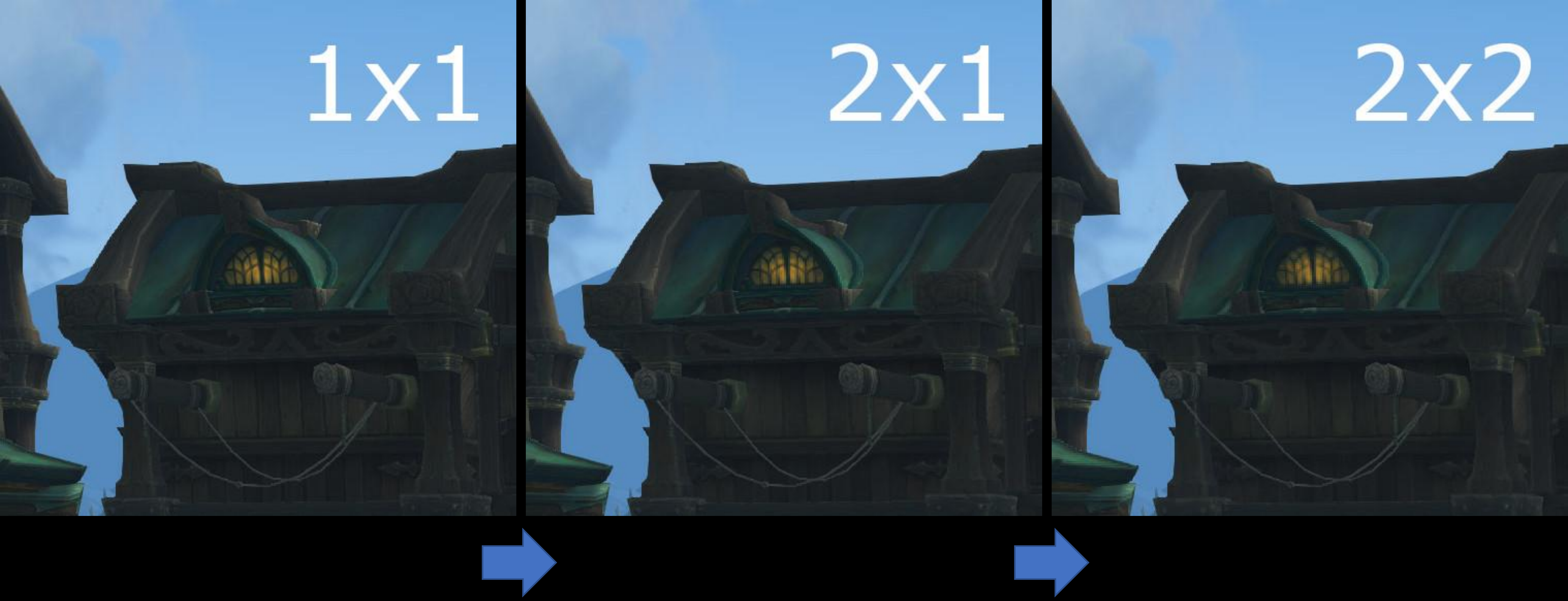
1080p 2x2



The main area of interest generally has the lowest texture size to screen space triangle size ratio (for systems like terrain)

VRS transitions can be fairly unnoticeable





2x1 makes a smoother transition by leveraging an assumed 16:9 like screen ratio

Best places to use VRS

Look for your heavy pixel shaders and cases where lower visual quality may not be perceptually noticeable

- Terrain
- Motion blur
- DOF
- Distant objects, etc.

Comparing VRS with Render Scale

VRS Pros:

- Preserves edges/silhouettes
  - Works well with edge based AA (such as CMAA)
- Control over which systems have VRS applied
  - Especially good for depth based text effects
- No upscale pass required

Comparing VRS with Render Scale

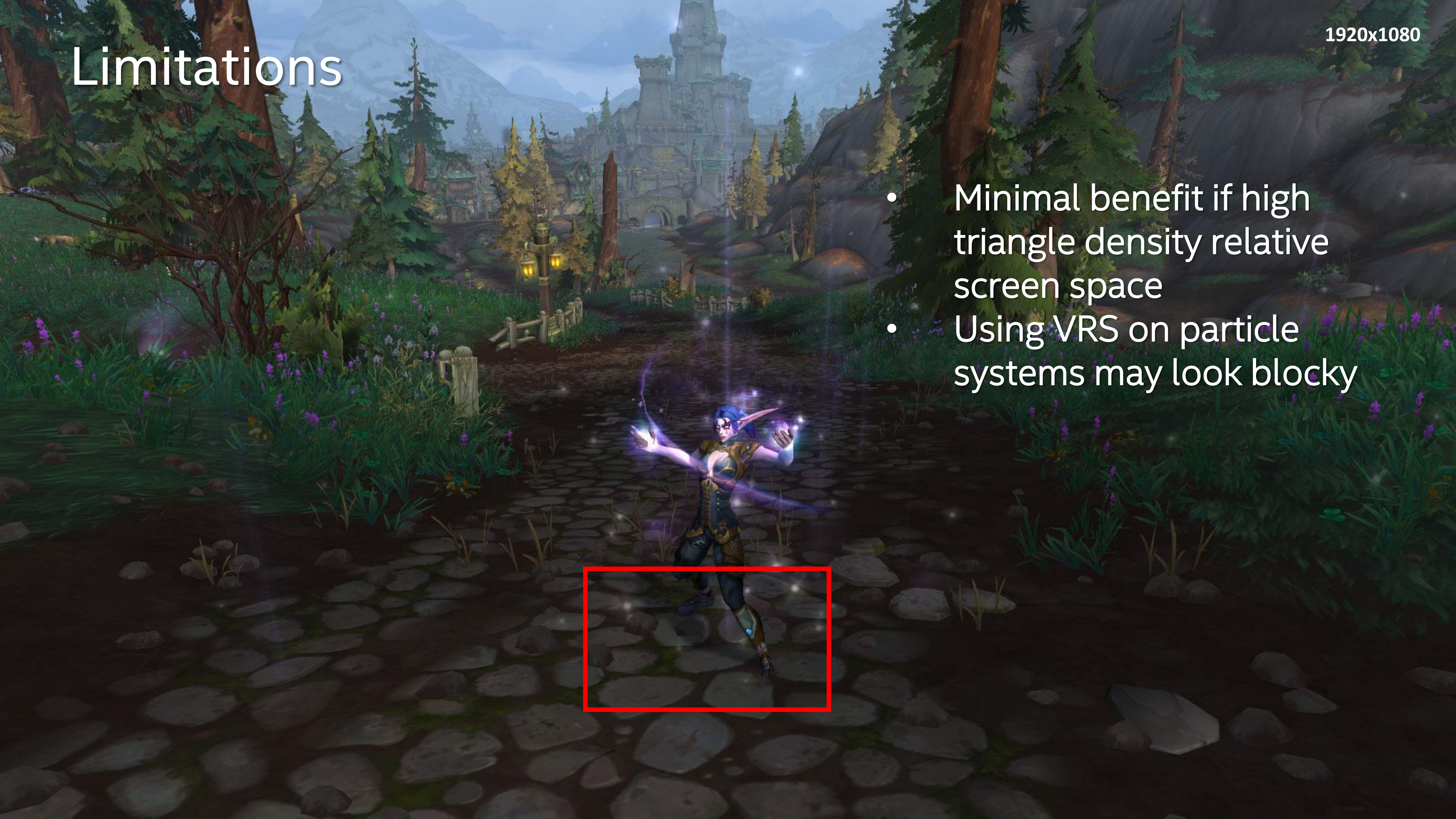
Render Scale Pros:

- Also reduces vertex and bandwidth costs
- Smooth range of intensity
  - e.g. all the values between 1920 to 3840

Limitations

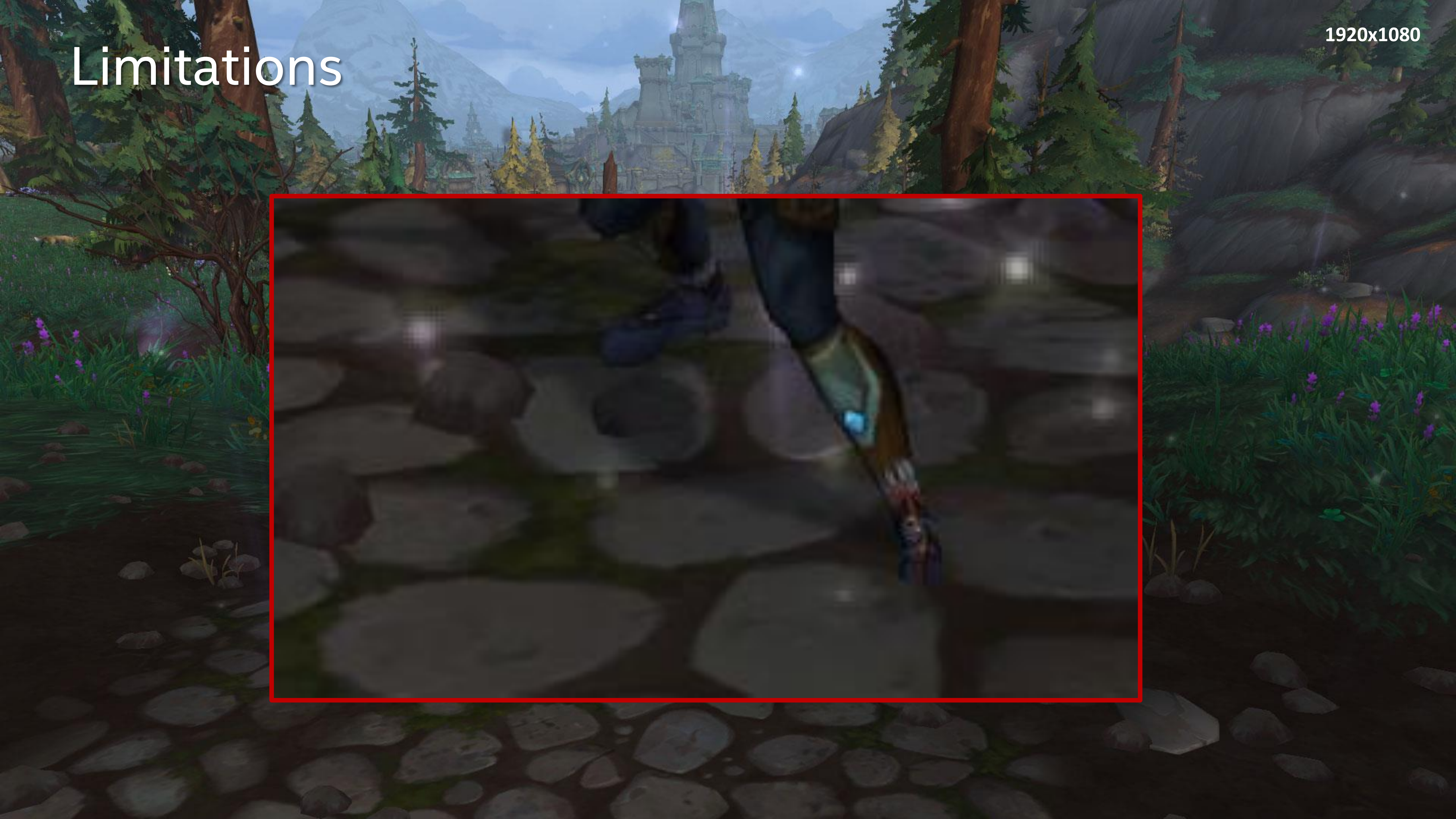
- Minimal benefit if high triangle density relative screen space
- Using VRS on particle systems may look blocky

Limitations



- Minimal benefit if high triangle density relative screen space
- Using VRS on particle systems may look blocky

Limitations

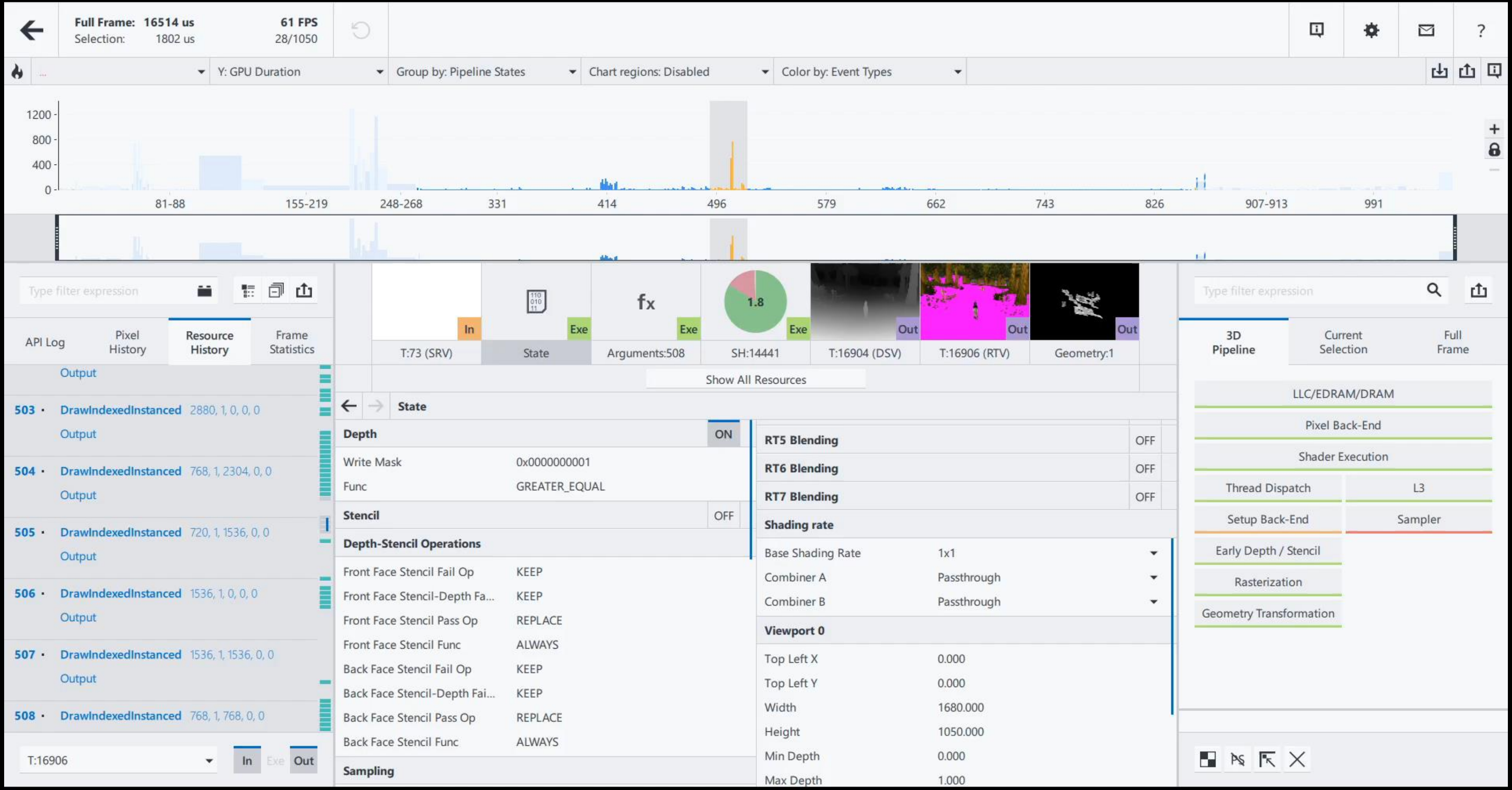


Performance on Icelake

Terrain in Elwynn Forrest example
61FPS @ 1x1
68FPS @ 2x2
76FPS @ 4x4

When balancing visual quality, ~5-10% has been common

VRS in Intel’s Graphics Performance Analyzer



Full Frame: 16514 us 61 FPS
Selection: 1802 us 28/1050

Y: GPU Duration | Group by: Pipeline States | Chart regions: Disabled | Color by: Event Types

```
        Output
503 •   DrawIndexedInstanced  2880, 1, 0, 0, 0
        Output
504 •   DrawIndexedInstanced  768, 1, 2304, 0, 0
        Output
505 •   DrawIndexedInstanced  720, 1, 1536, 0, 0
        Output
506 •   DrawIndexedInstanced  1536, 1, 0, 0, 0
        Output
507 •   DrawIndexedInstanced  1536, 1, 1536, 0, 0
        Output
508 •   DrawIndexedInstanced  768, 1, 768, 0, 0
```

T:16906 In Exe Out

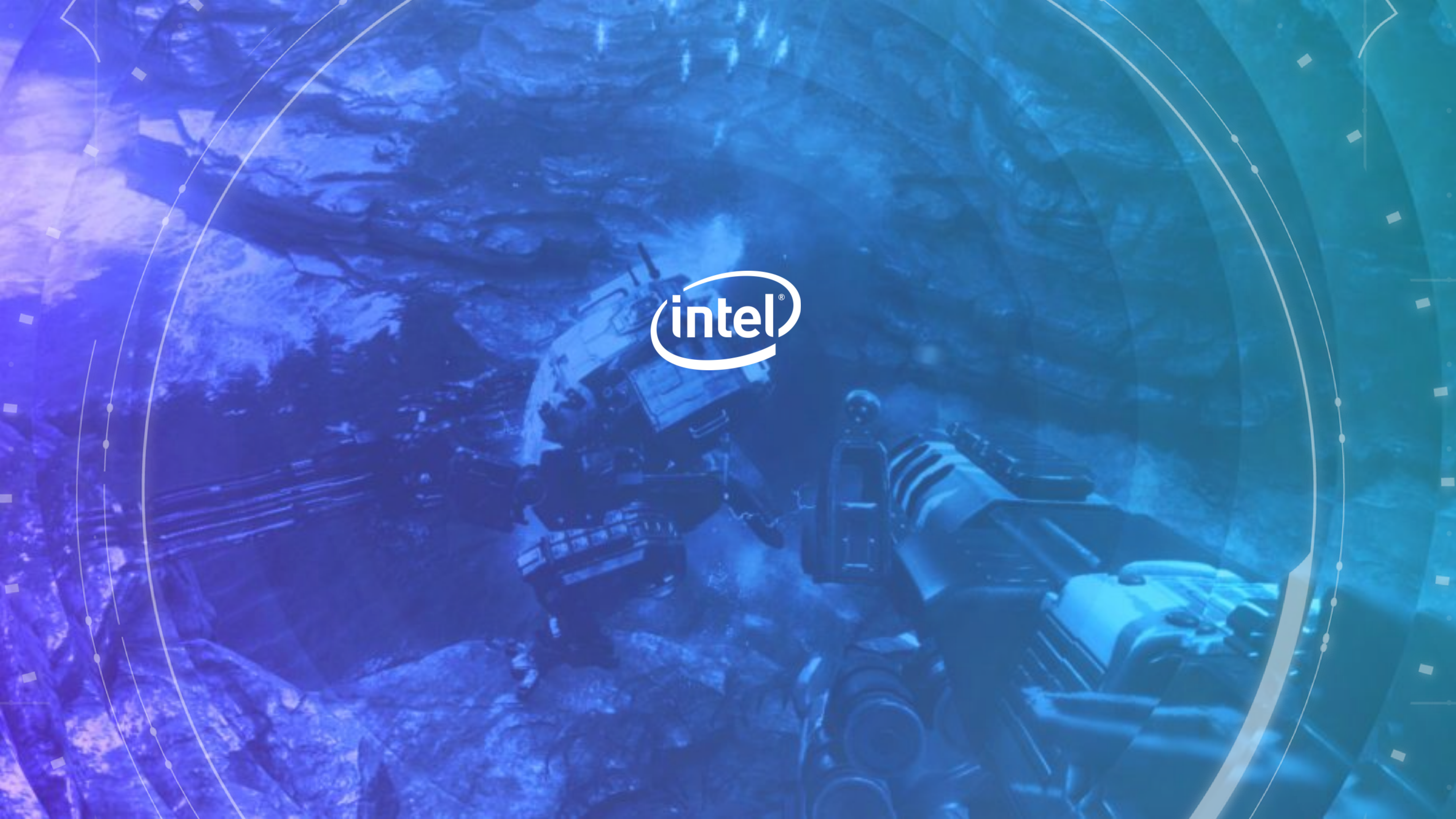
State

Depth	ON
Write Mask	0x0000000001
Func	GREATER_EQUAL
Stencil	OFF
Depth-Stencil Operations	
Front Face Stencil Fail Op	KEEP
Front Face Stencil-Depth Fa...	KEEP
Front Face Stencil Pass Op	REPLACE
Front Face Stencil Func	ALWAYS
Back Face Stencil Fail Op	KEEP
Back Face Stencil-Depth Fai...	KEEP
Back Face Stencil Pass Op	REPLACE
Back Face Stencil Func	ALWAYS
Sampling	

RT5 Blending	OFF
RT6 Blending	OFF
RT7 Blending	OFF
Shading rate	
Base Shading Rate	1x1
Combiner A	Passthrough
Combiner B	Passthrough
Viewport 0	
Top Left X	0.000
Top Left Y	0.000
Width	1680.000
Height	1050.000
Min Depth	0.000
Max Depth	1.000

Takeaways

- Per Draw call VRS is very easy to integrate (into a DX12 engine)
- Can be dynamically turned on, seamlessly
- Can reduce pixel shader cost with minimal reduction in quality



intel