

Intel® Open Image Denoise in Unity*

Open Source Denoising for Lightmaps



GAME DEVELOPERS
CONFERENCE

Carson Brownlee (Intel), Jesper Mortensen (Unity)

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 [\[intel.com\]](http://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

© Intel Corporation.

Agenda

Intel Rendering Framework

Open Image Denoise (OIDN) Overview

Source Code Examples

Unity Overview

Lightmaps in Unity Overview

Unity Lightmap Baking Results



Unity*

- See future OIDN specific paper/talk for specific details of training algorithm and denoiser



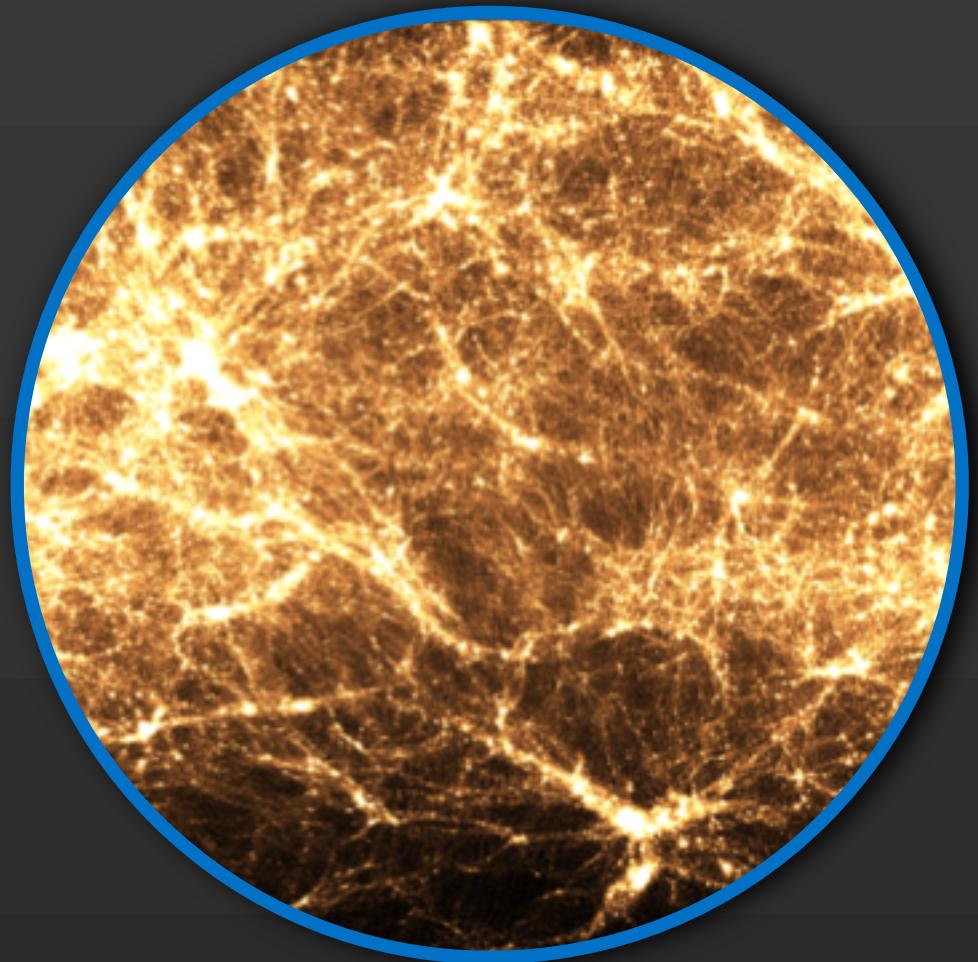
#IntelGameDev

*OTHER NAMES AND BRANDS MAY BE CLAIMED AS THE PROPERTY OF OTHERS.

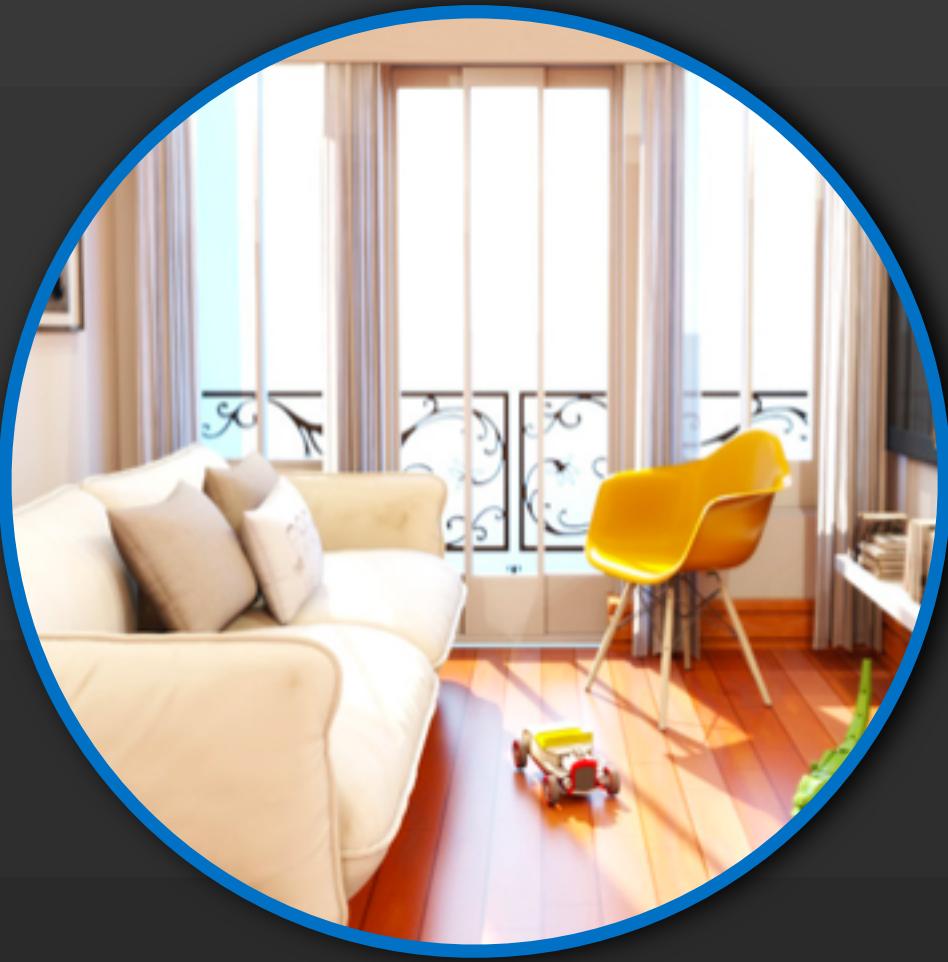


INTEL® Rendering Framework

INTEL®



OpenSWR



Embree



OSPRay



Open Image
Denoise



#IntelGameDev



INTEL® Embree

Optimized Ray Tracing Kernels
Acceleration Structures
Ray Traversals
Ray Intersections

Industry Leading CPU Performance
1.5-6x speedups reported by users



Many Use Cases
Real-time applications
Photorealism
Simulation

Open Source
Apache* 2.0 License
<http://embree.github.com>

Embree: A Kernel Framework for Efficient CPU Ray Tracing
Wald et al. (SIGGRAPH '14)

#IntelGameDev

*OTHER NAMES AND BRANDS MAY BE CLAIMED AS THE PROPERTY OF OTHERS.

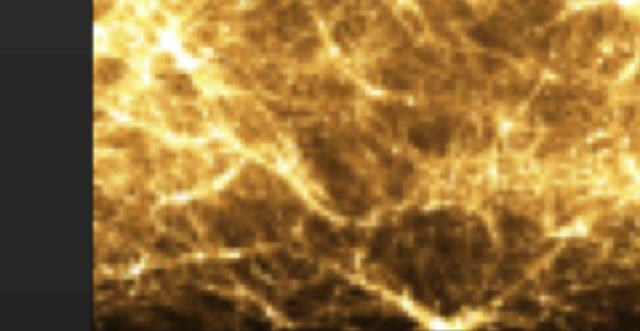
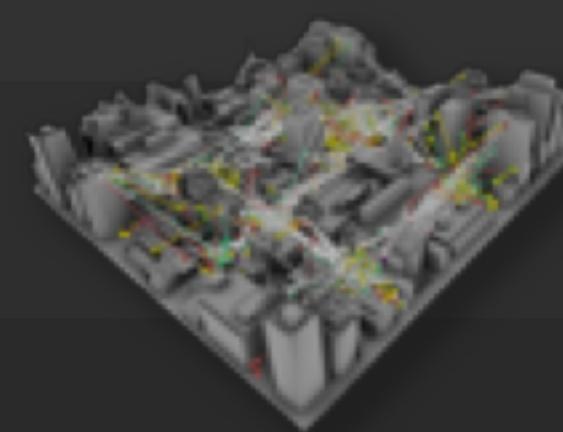
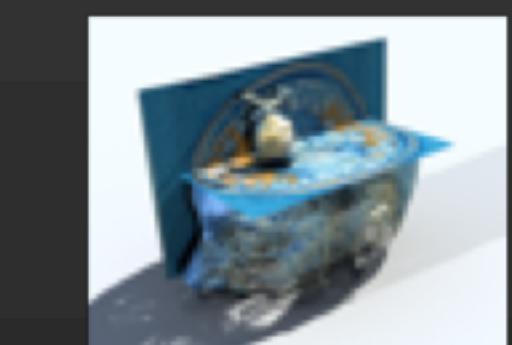








Wide Adoption - 60+ Apps



And...



#IntelGameDev

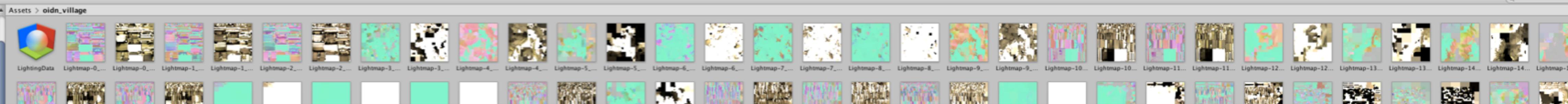


File Edit Assets GameObject Component Window Help



Project Console

Assets



INTEL® OSPRay

Open Scalable Portable Ray Tracing Engine

Utilizes Embree

Optimized for laptops, workstations,
and multi-node clusters

Open Source under Apache* 2.0

Distributed Rendering

Multiple nodes for greater speed

Larger cumulative memory

In Situ applications

Scientific Visualization

Distributed with ParaView binaries

Support for VMD, EnSight*, VisIt*, many others

Portable

Plugin Interface

Modular

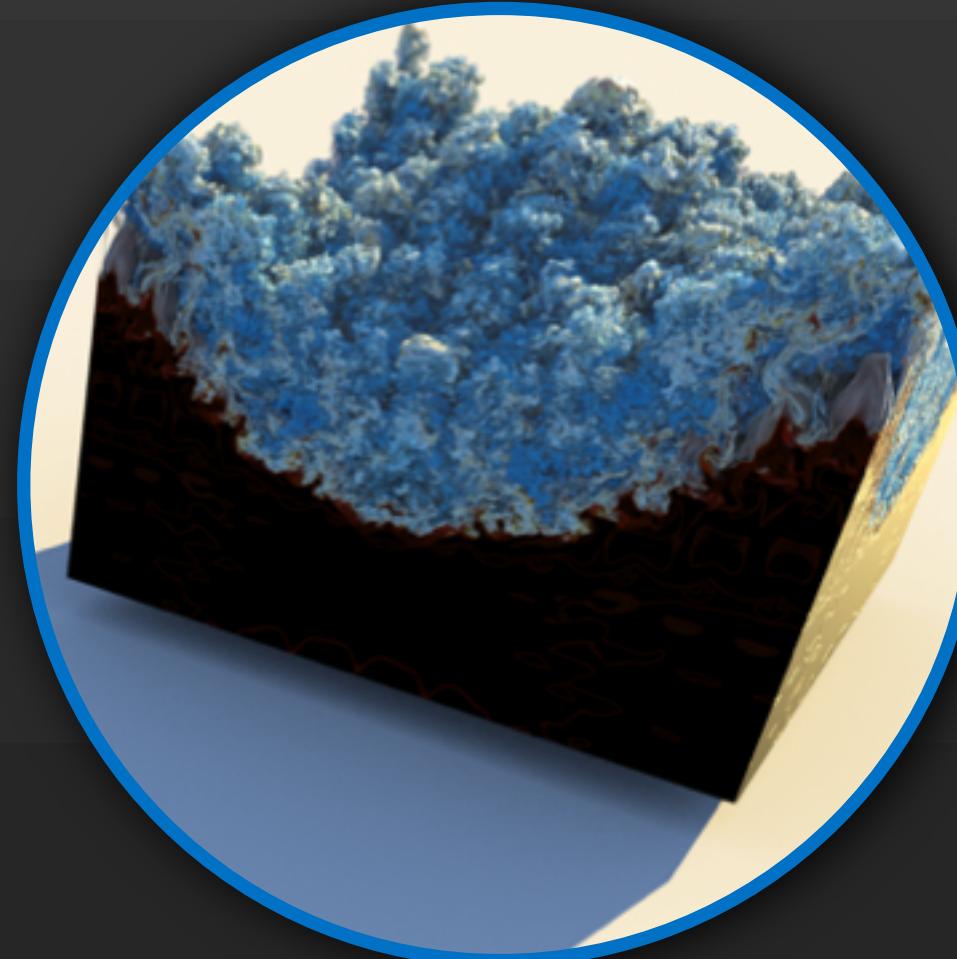
User Defined Geometry

Multiple Use Cases

Real-time rendering

Photorealism

Simulations



ospray.org

OSPRay - A CPU Ray Tracing Framework for Scientific Visualization

Wald et al. (TVCG, IEEE SciVis '16)



#IntelGameDev

*OTHER NAMES AND BRANDS MAY BE CLAIMED AS THE PROPERTY OF OTHERS.



Broad Industry + Academic Support

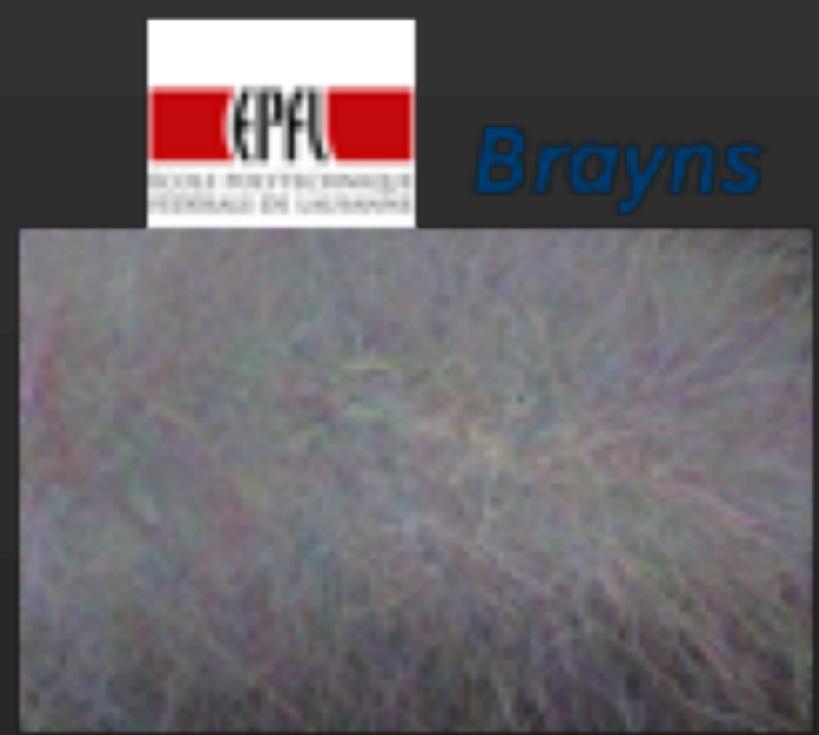
NASA



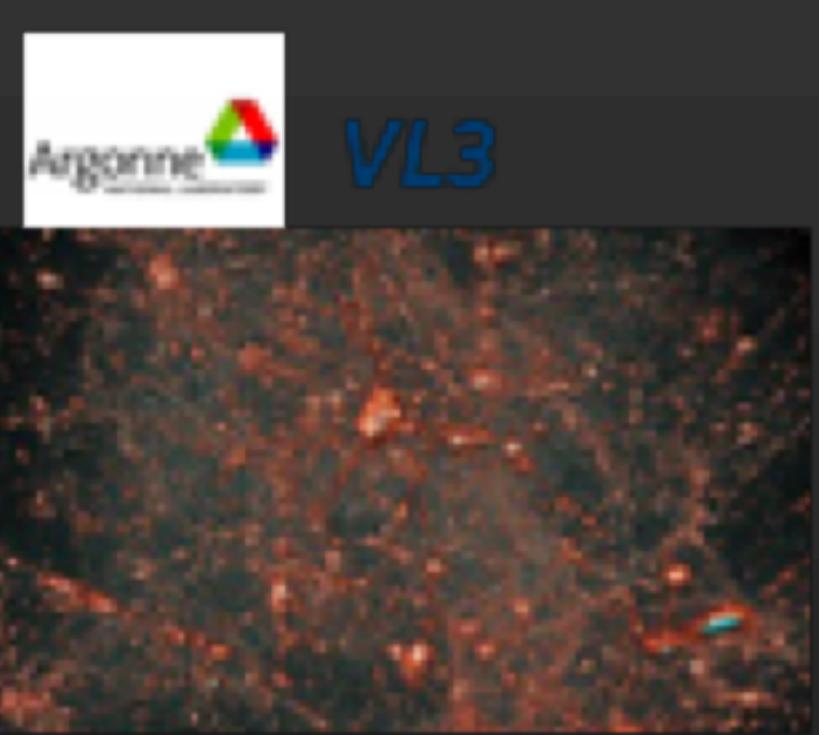
Brayns
V
L3



Courtesy John Stine Beckman
Institute, Univ Illinois Urbana-Champaign

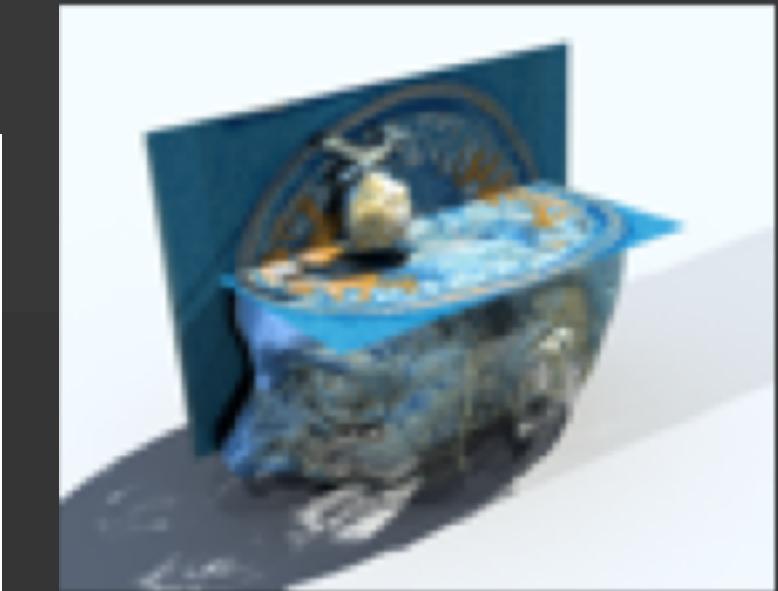


Data and Visualization courtesy
Cyrille Favreau, EPFL



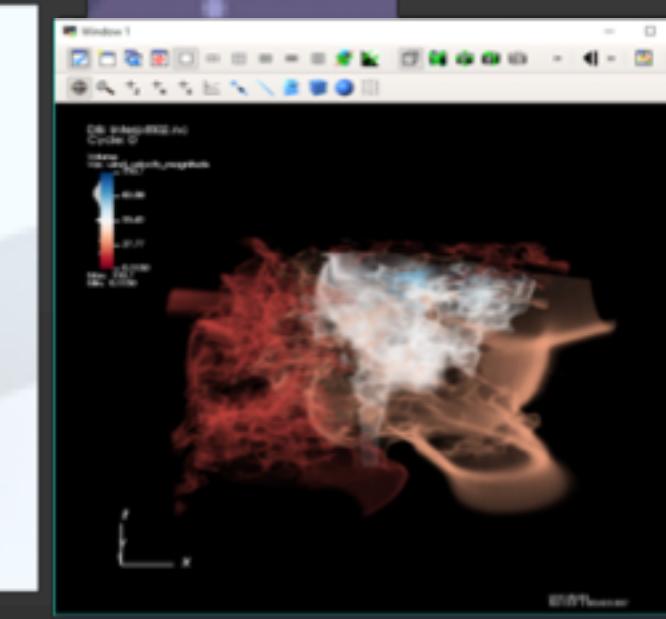
Data: Salman Habib, Kathrin Heermann,
and the HACC team.
Visualization: Joe Insley, Silvio Risi,
ANL

ParaView



Data courtesy Kitware
Visualization, Carson Brownlee, Jan Huang and Alex Hora, UTenn
Intel

visit



Courtesy Hank Childs, U Oregon,
Visualization, Carson Brownlee, Jan Huang and Alex Hora, UTenn
Intel

USD Plugin



#IntelGameDev

*OTHER NAMES AND BRANDS MAY BE CLAIMED AS THE PROPERTY OF OTHERS.

intel

13

INTEL® Open Image Denoise



#IntelGameDev



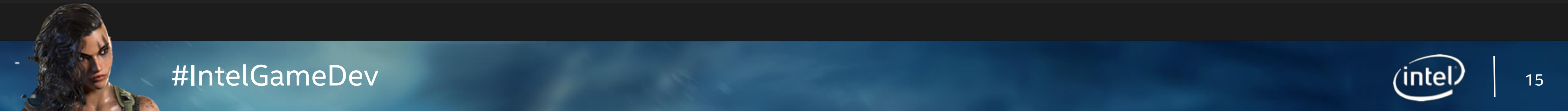
Why OIDN?

Not bound to a specific GPU vendor

Runs on any x86 CPU - does not require Intel hardware

Can achieve interactive performance

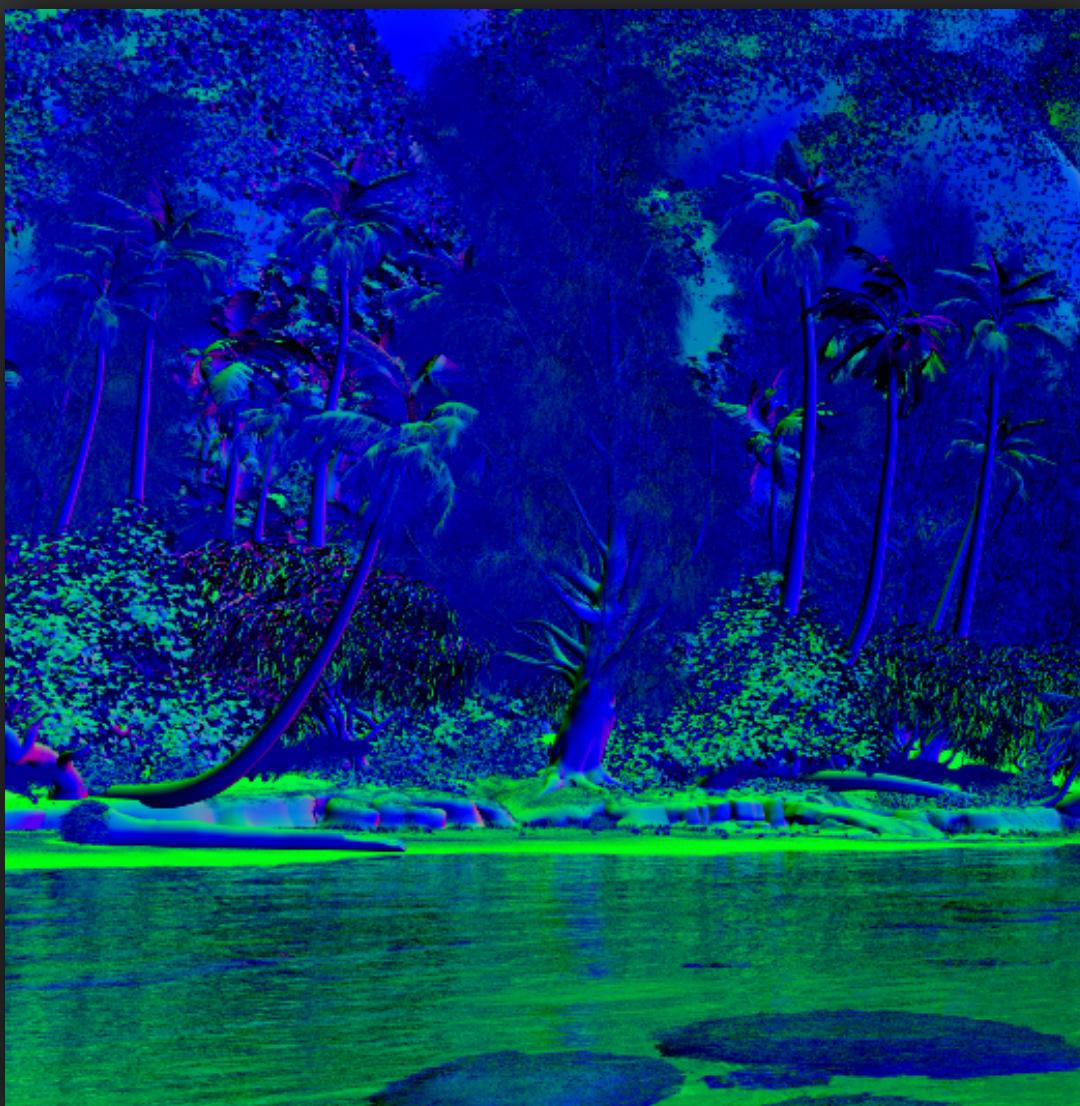
Open - modifications and community contributions welcome



Albedo



Normal



Color



+

+

=

OIDN Overview

First shown at SIGGRAPH 2018, Officially released Q1 2019

Open Source under Apache 2.0 License

CPU-based

AI-based denoising

Quality will continue to improve with increased training

Up to interactive performance @ 1080p



#IntelGameDev



OIDN Overview (2)

SIMD optimized up to AVX-512

Binary packages for x86 Linux/Mac/Windows at:

<https://openimagedenoise.github.io/downloads.html>

Utilizes Intel® Math Kernel Library for Deep Neural Networks (MKL-DNN)

and Intel® Threaded Building Blocks (TBB)

Support for HDR

Example application - Disney* Moana Island Scene

Intel® OSPRay Renderer

- > 28 million instances
- > 15 billion primitives
- > 100GB memory utilization per node

~31GB compressed texture data

8 Xeon 8180 nodes (7 nodes for rendering, 1 for denoising)



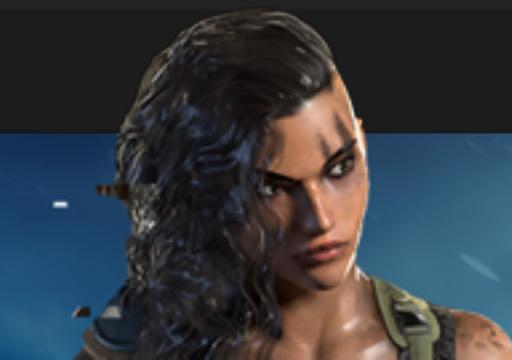
#IntelGameDev

*OTHER NAMES AND BRANDS MAY BE CLAIMED AS THE PROPERTY OF OTHERS.



Moana Island Scene

Demo



#IntelGameDev



20

OIDN API - Example

```
#include "OpenImageDenoise/oidn.hpp"
```

Include C++ headers. C headers also available.

```
oidn::DeviceRef device = oidn::newDevice();
```

Create OIDN device, set reference counted handle to new device.

```
const char* errorMessage;
if (device.getError(errorMessage) != oidn::Error::None)
| throw std::runtime_error(errorMessage);
device.setErrorFunction(errorCallback);
```

Check for reported errors, and optionally set an error callback for future errors.

```
device.commit();
```

Managed objects must be committed.



#IntelGameDev



OIDN API - Example (Part 2)

```
oidn::FilterRef filter = device.newFilter("RT");
```

Create a filter for monte carlo ray tracing.

```
filter.setImage("color", color.data, oidn::Format::Float3, width, height);
if (albedo)
| filter.setImage("albedo", albedo.data, oidn::Format::Float3, width, height);
if (normal)
| filter.setImage("normal", normal.data, oidn::Format::Float3, width, height);
filter.setImage("output", output.data, oidn::Format::Float3, width, height);
```

Set image buffers for inputs and outputs.

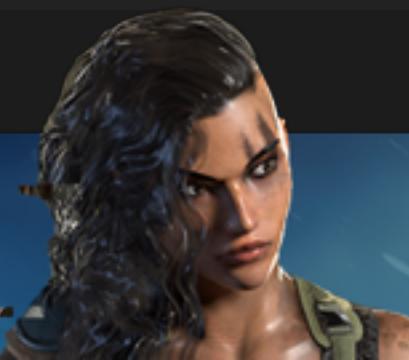
```
if (hdr)
| filter.set("hdr", true);

filter.commit();
```

Set filter parameters and commit.

```
filter.execute();
```

Execute filter. Denoised output is now in output buffer.



#IntelGameDev

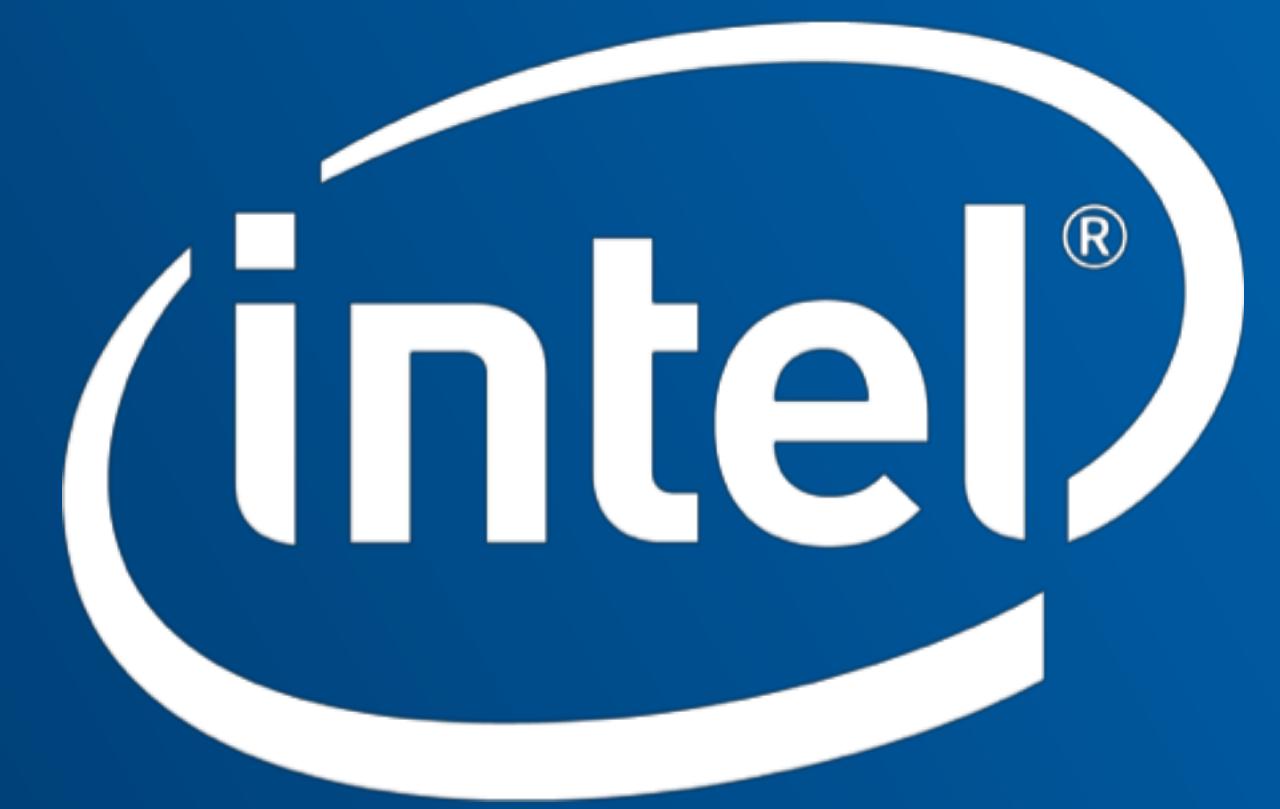


that's it!



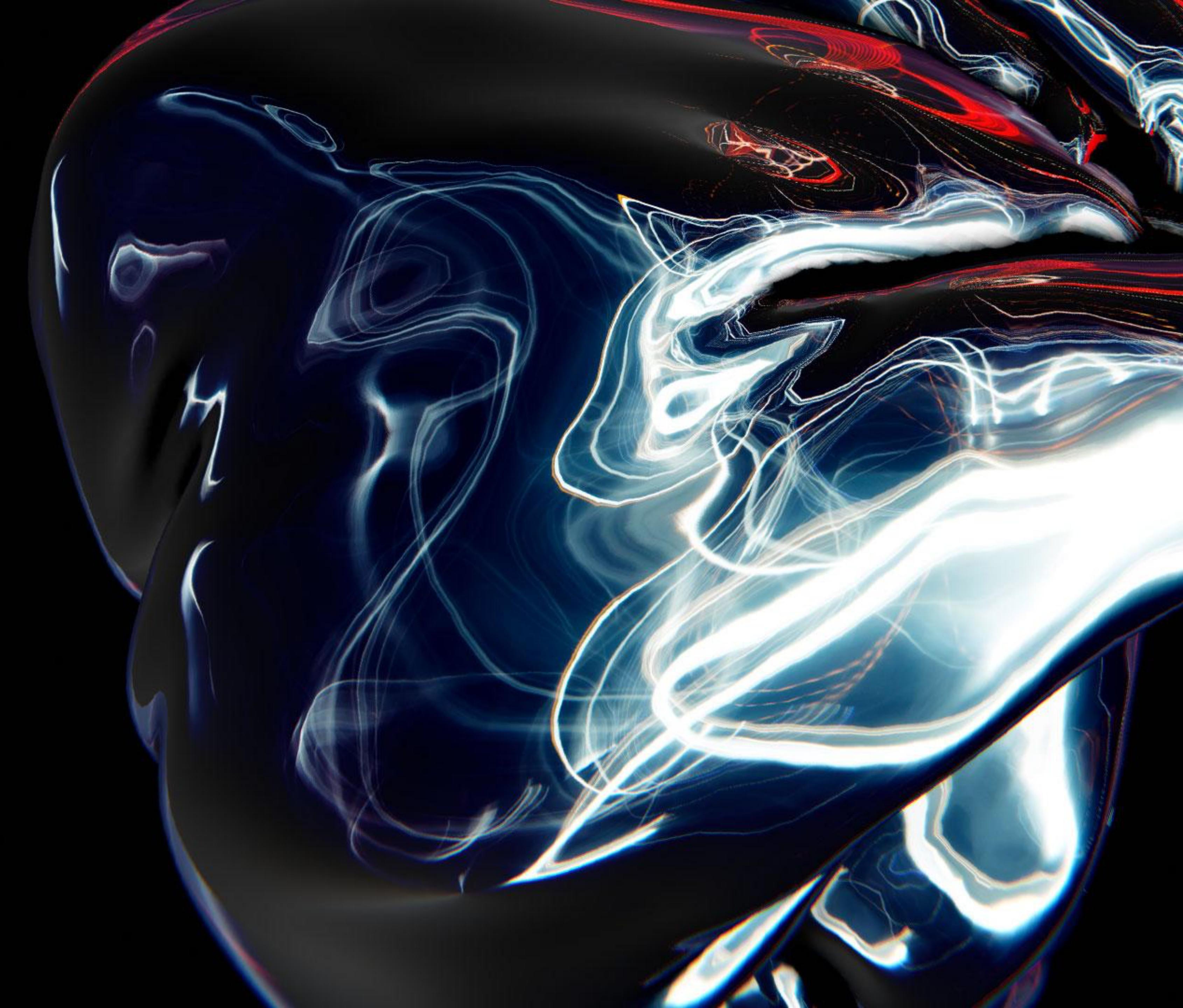
#IntelGameDev

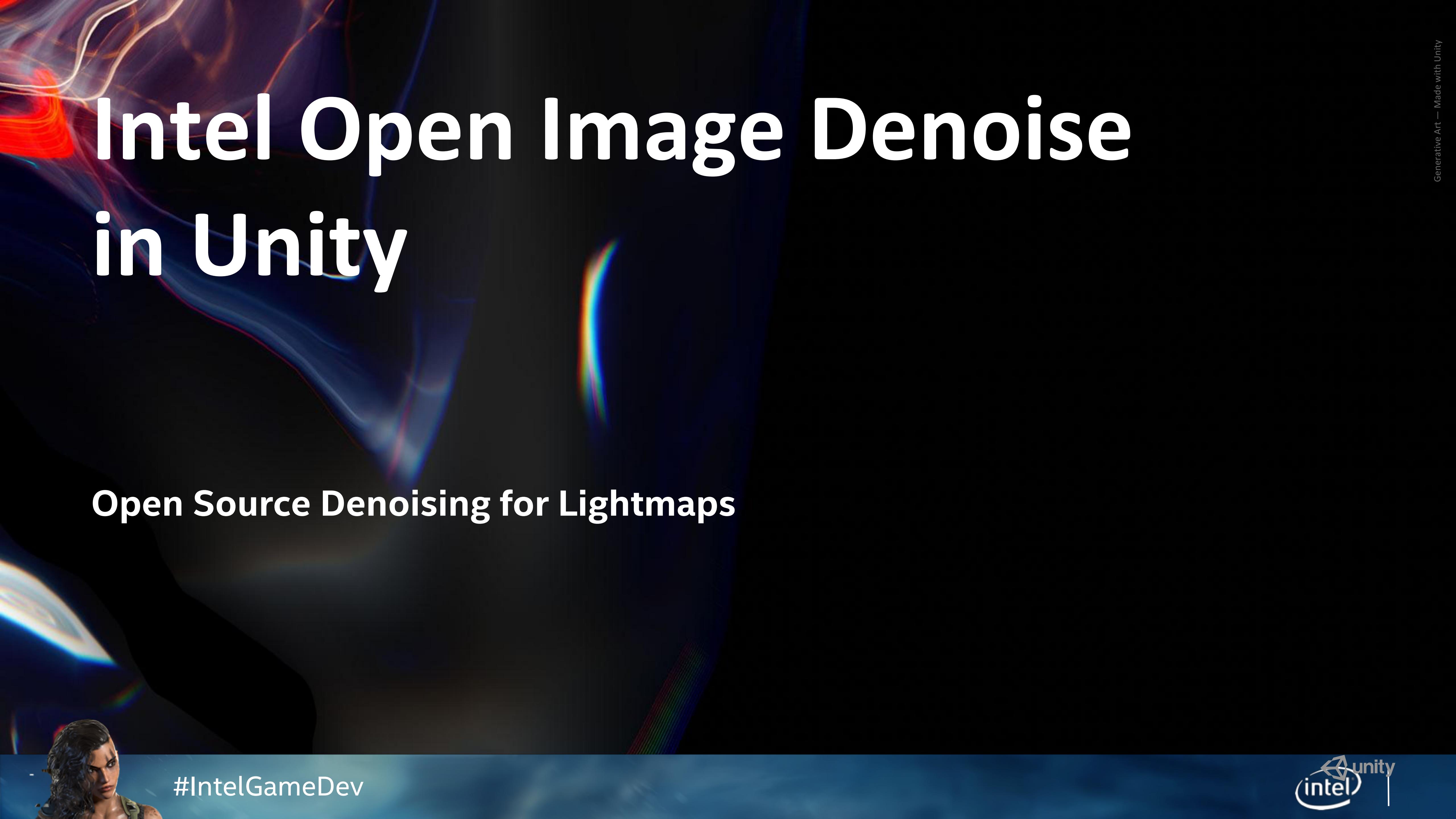






**GDC
2019**





Intel Open Image Denoise in Unity

Open Source Denoising for Lightmaps

#IntelGameDev



Support our creators.



#IntelGameDev



#IntelGameDev

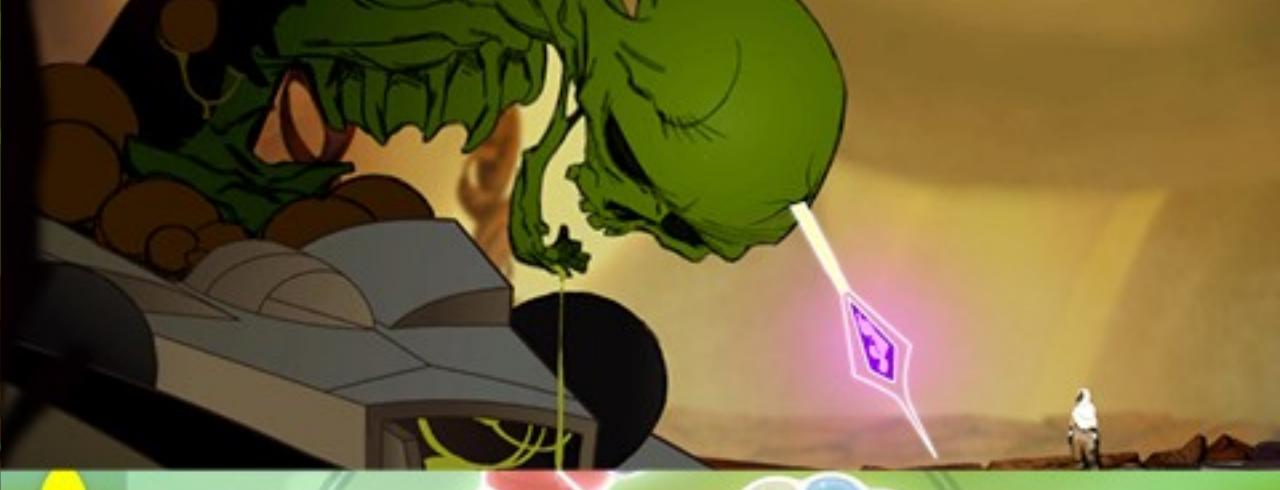
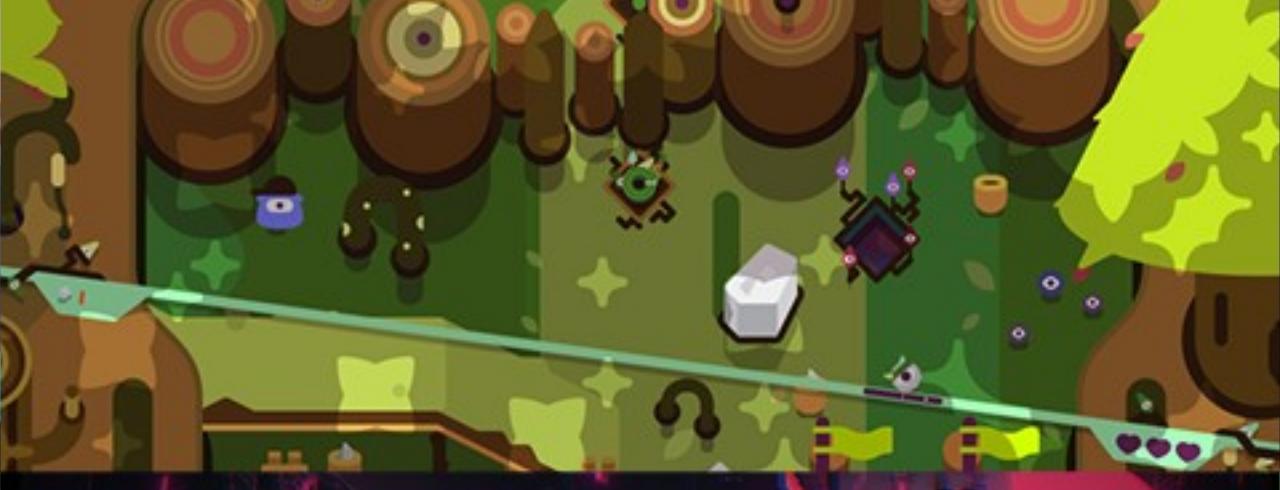
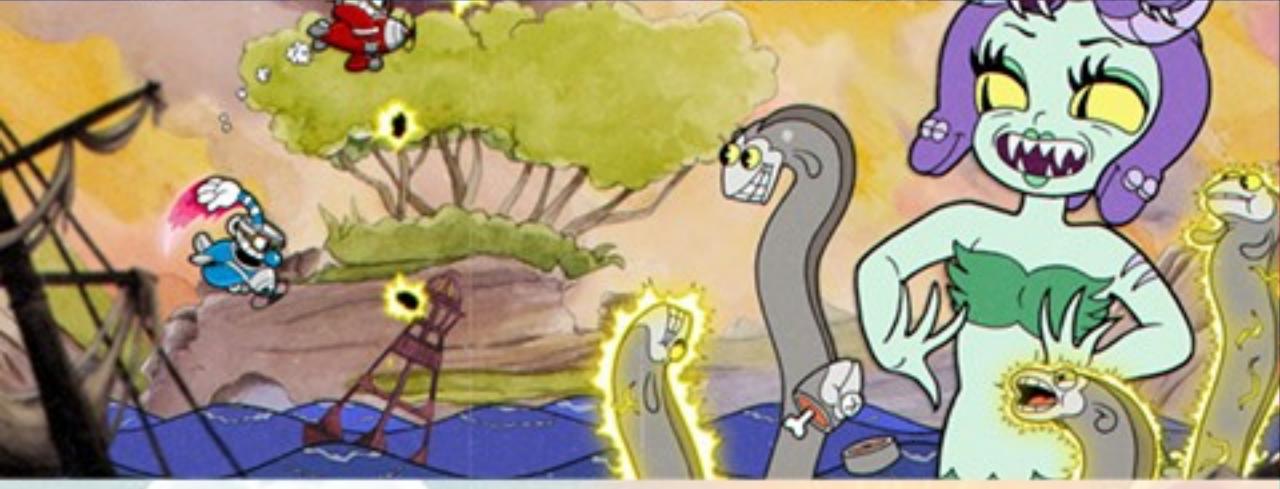


#IntelGameDev



#IntelGameDev





Our creators are doing
well.



#IntelGameDev

Our Reach

3 Billion

devices reached with Made with Unity
experiences in the last 12 months

29 Billion

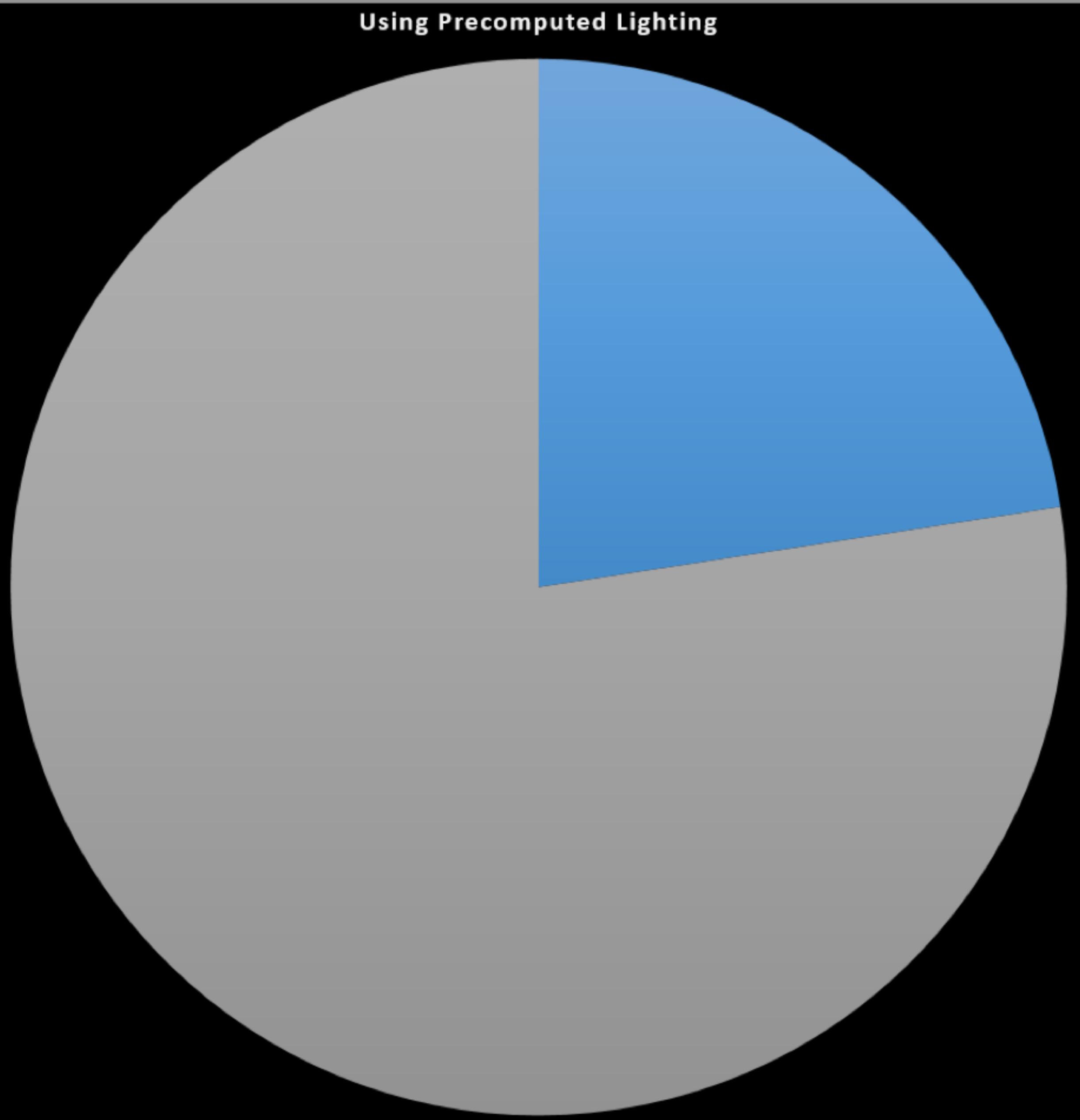
installs of Made with Unity experiences in
the last 12 months

How many creators are using
lightmapping?



#IntelGameDev

23% of our creators
use baked lighting

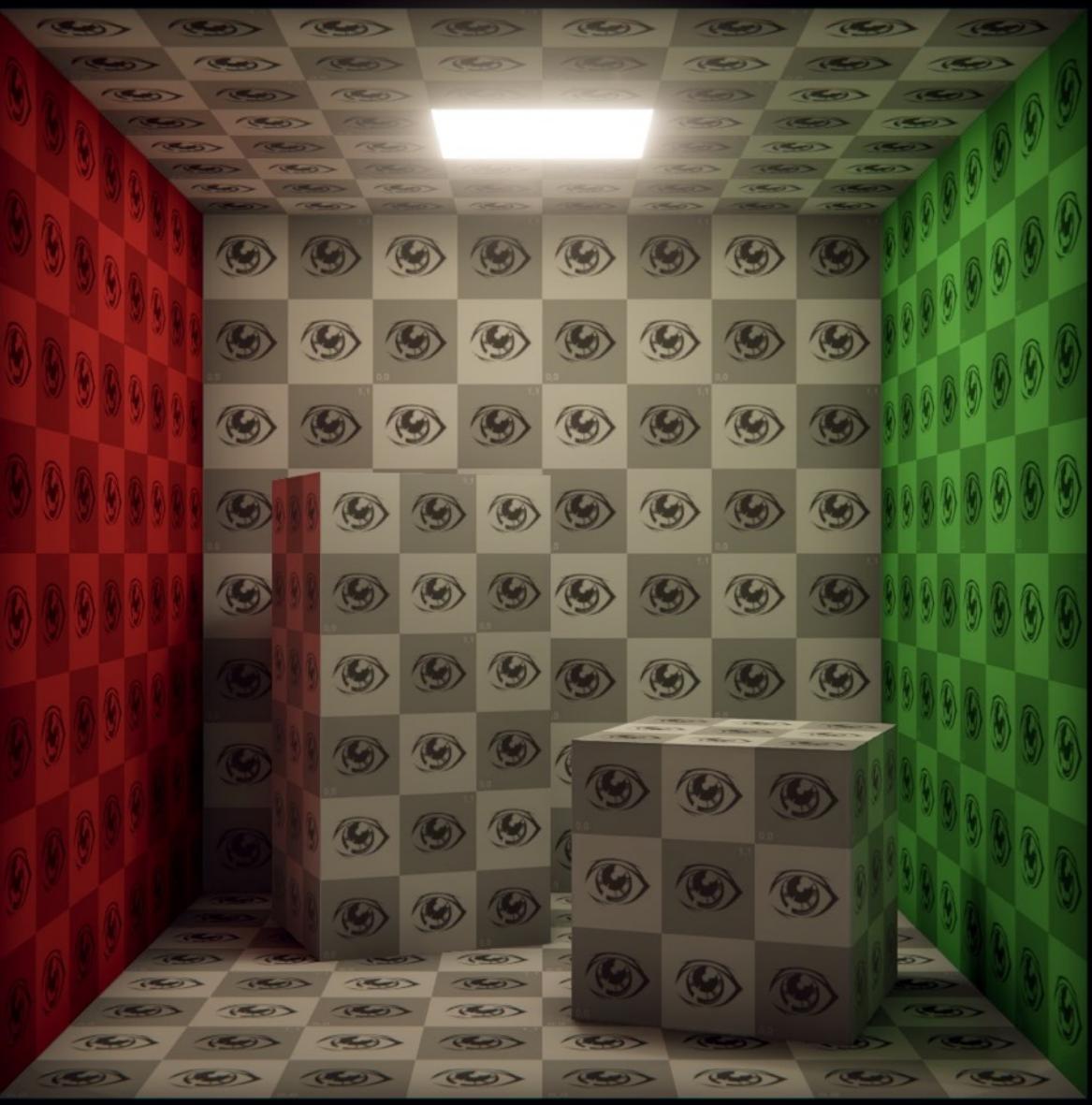


Monthly 80M+ bakes

~30 bakes per second

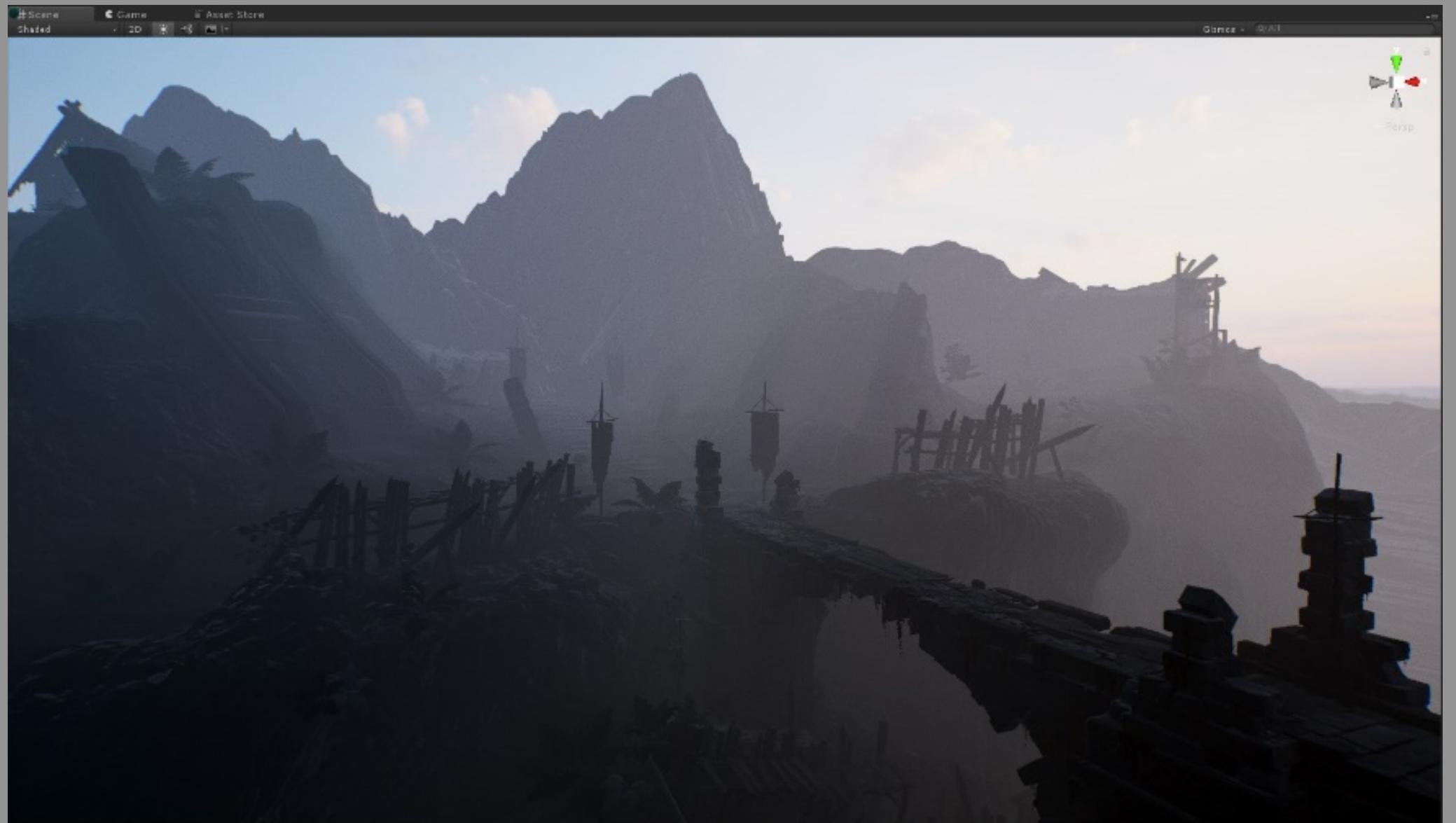


#IntelGameDev



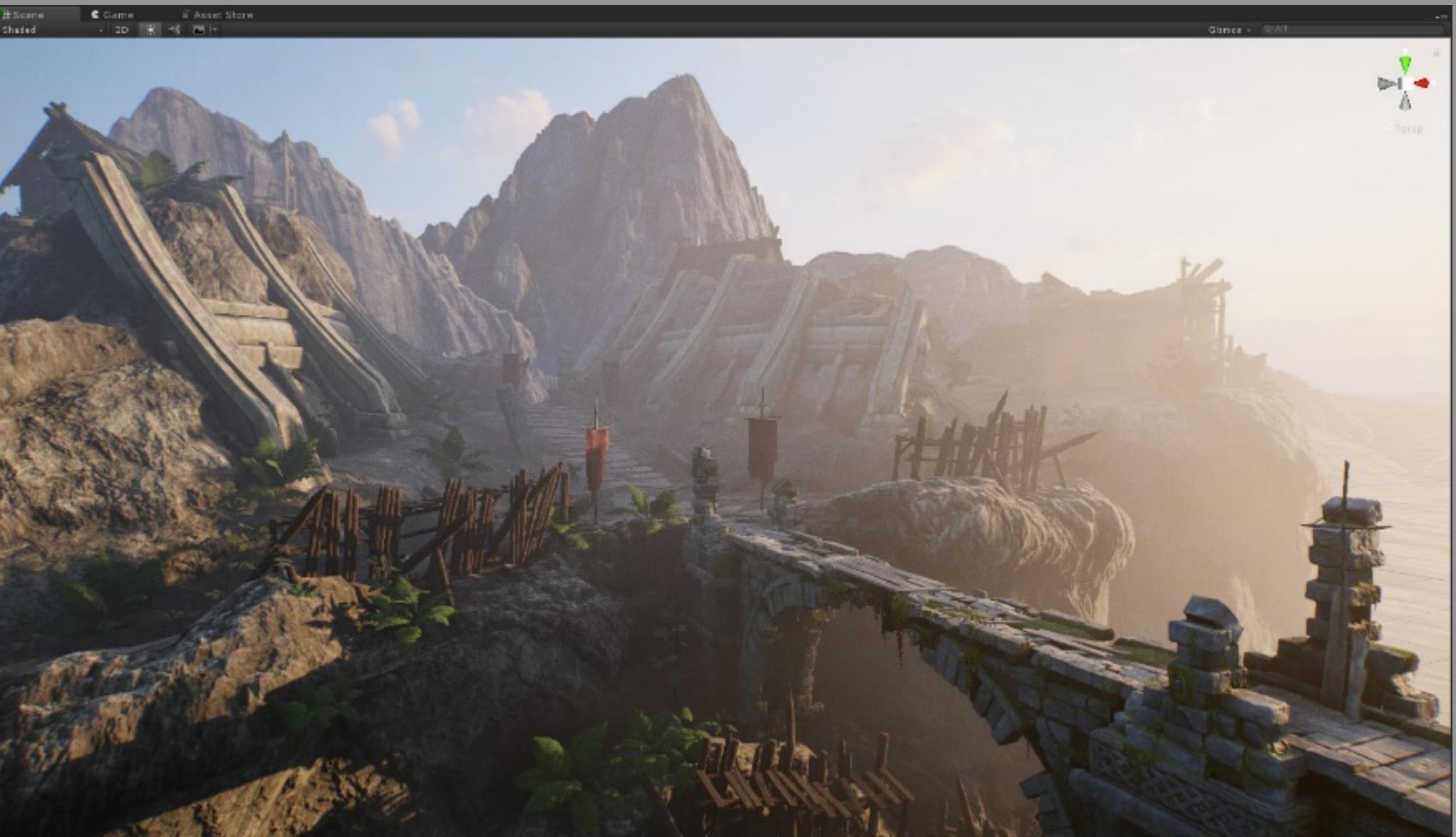
A bake can take anywhere from
seconds to hours...





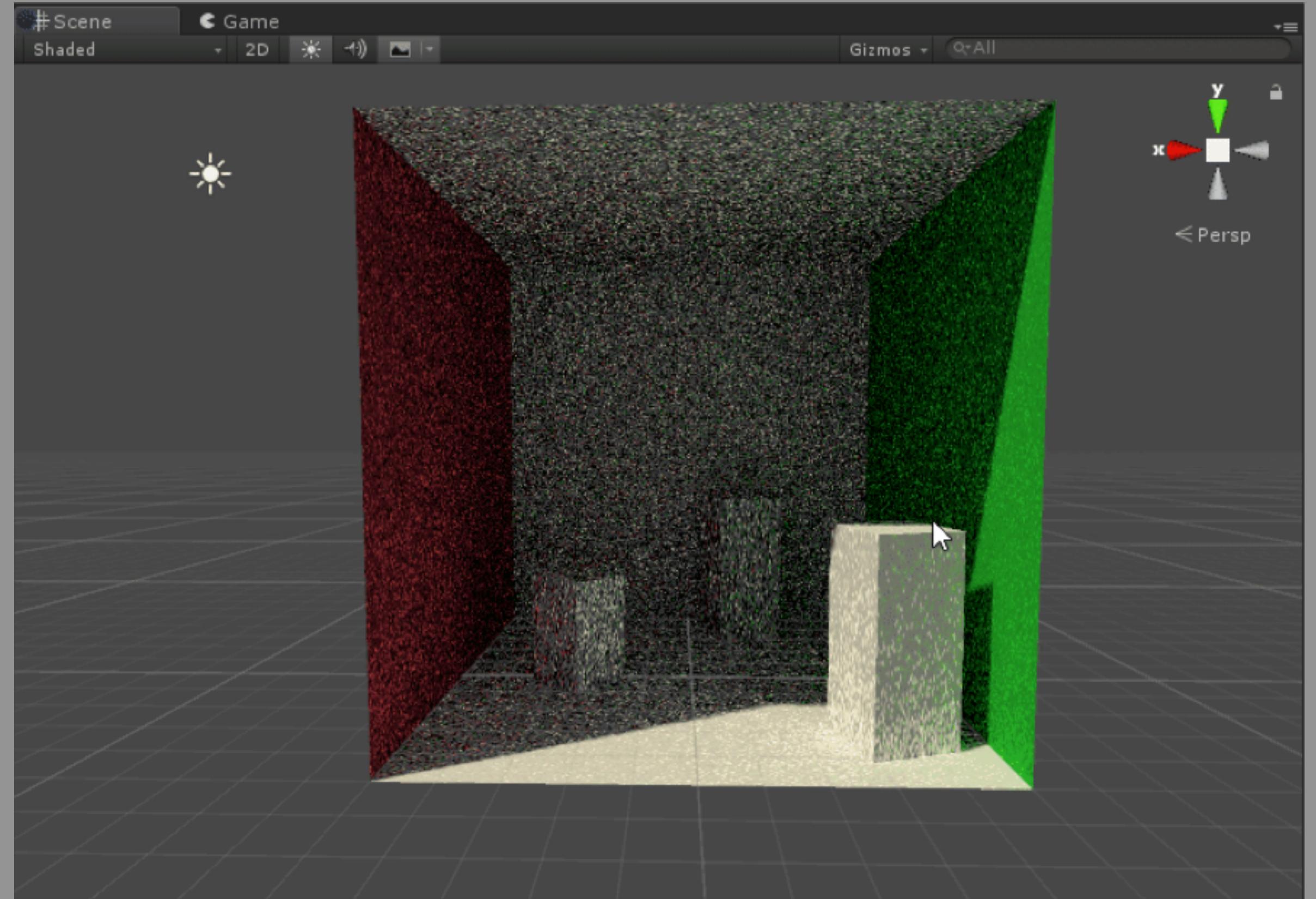
press
wait...

Generate Lighting



A better workflow

- Progressive updates
- View prioritization
- GPU Lightmapping
 - Shoot rays faster!
- Sampling strategies
 - Shoot fewer rays!
- We're done!



But wait there's more.

...Tracing rays is not enough...



#IntelGameDev

We post-process the lightmaps



#IntelGameDev

Lightmap post-processing



A-Trous filtering:
Two column body text lorem
ipsum dolor sit amet, consectetur
adipiscing elit.
• Will fail at low sampling rates



• Biased at high sampling rates
#IntelGameDev



Bi-lateral Gaussian:
One column body text lorem
ipsum dolor sit amet, consectetur
adipiscing elit.
• Biased at high sampling rates

OIDN – 32 samples



Bi-lateral Gaussian – 32 samples



None – 32 samples

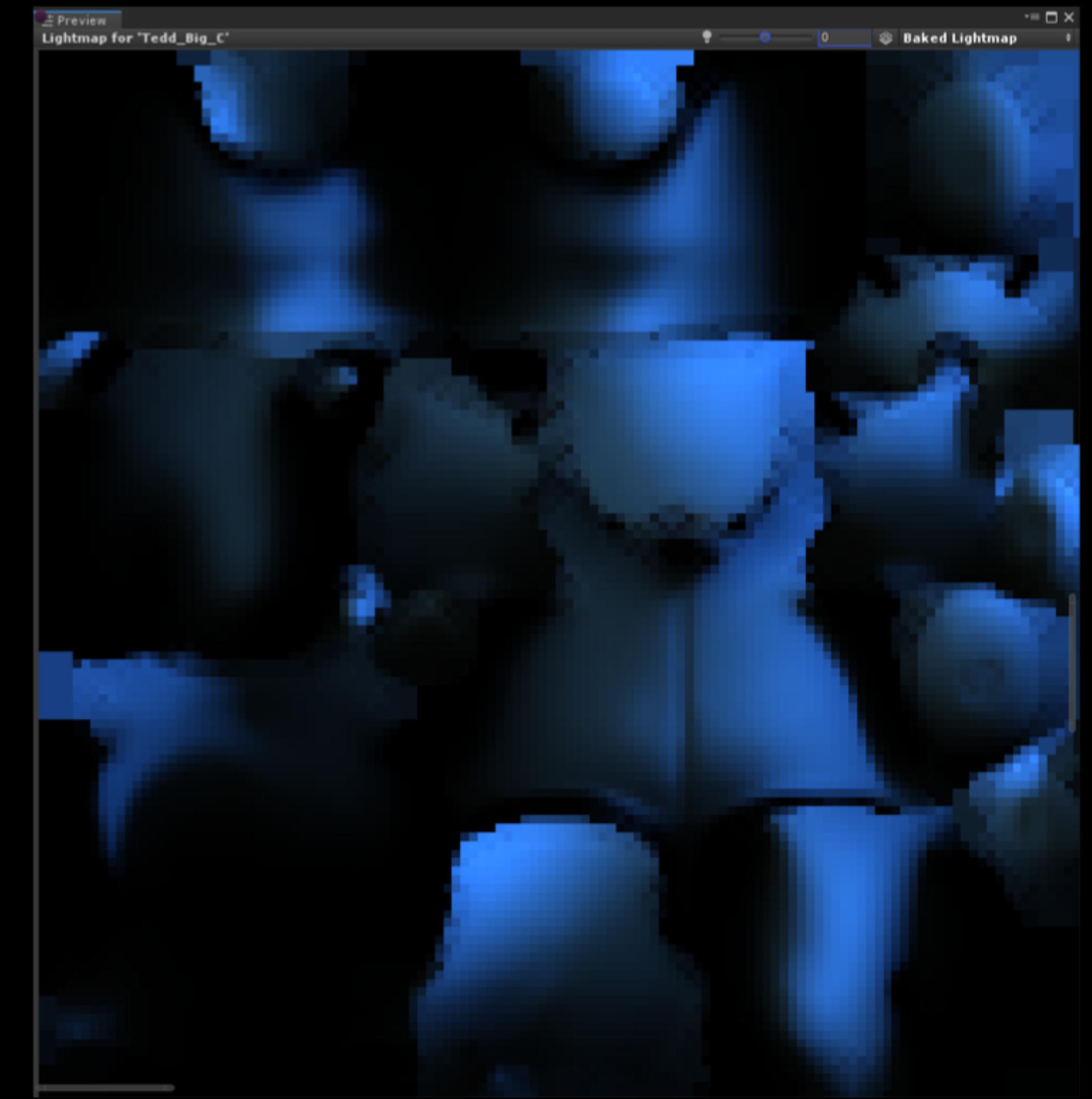


#IntelGameDev

We integrated Open Image Denoise in Unity



#IntelGameDev



#IntelGameDev

Enter the Room



#IntelGameDev

NONE



8 samples



OIDN



#IntelGameDev

NONE

16 samples

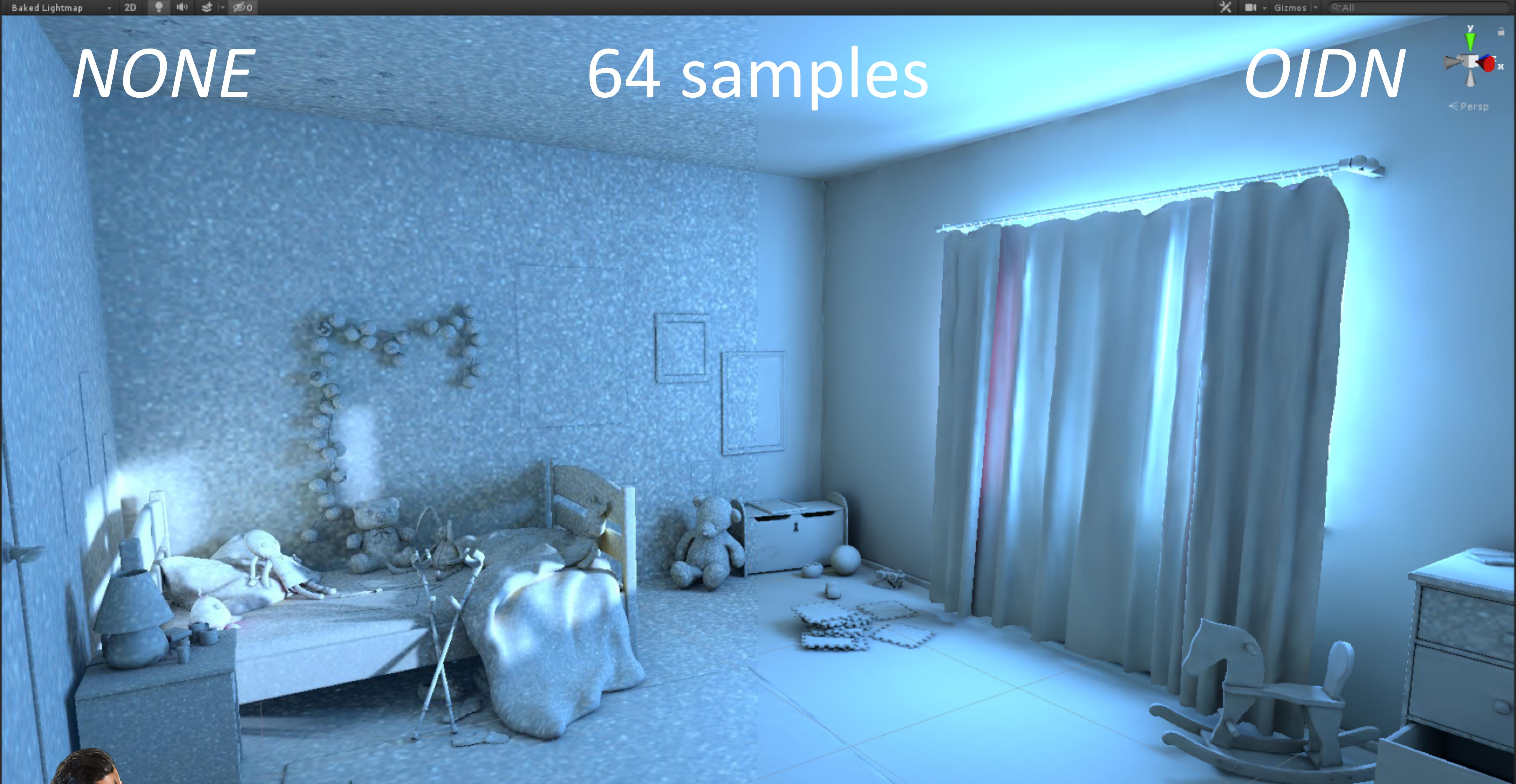
OIDN



NONE

64 samples

OIDN



NONE

128 samples

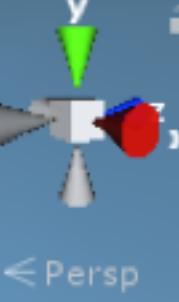
OIDN



NONE

256 samples

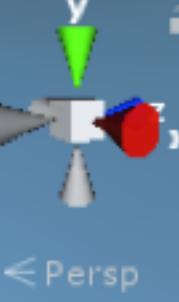
OIDN



NONE

512 samples

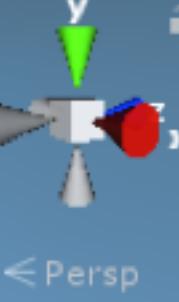
OIDN



NONE

1024 samples

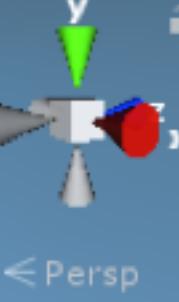
OIDN



NONE

2048 samples

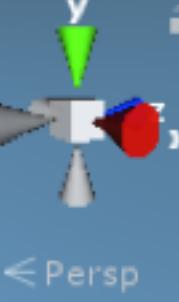
OIDN



NONE

4096 samples

OIDN

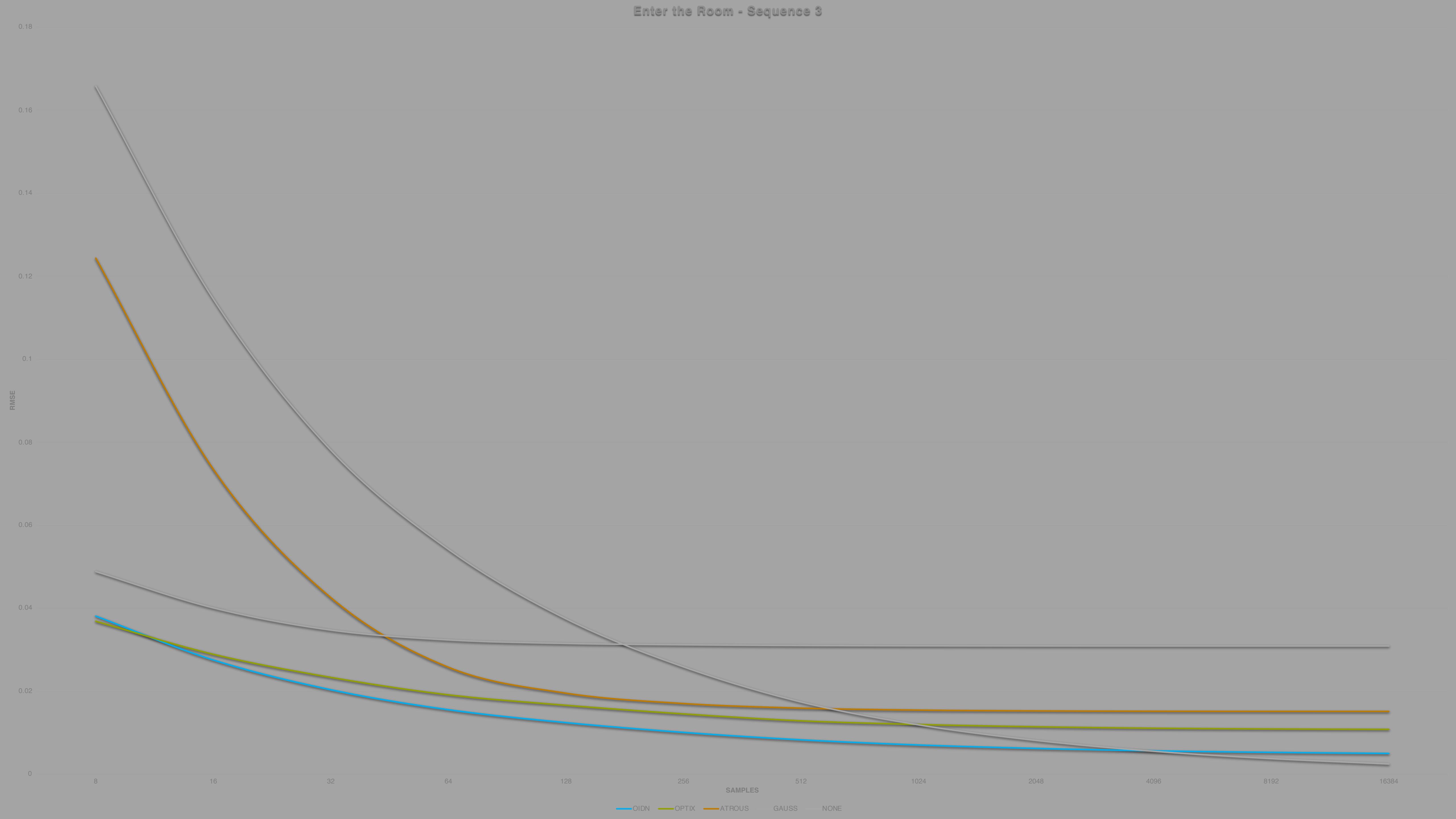


NONE

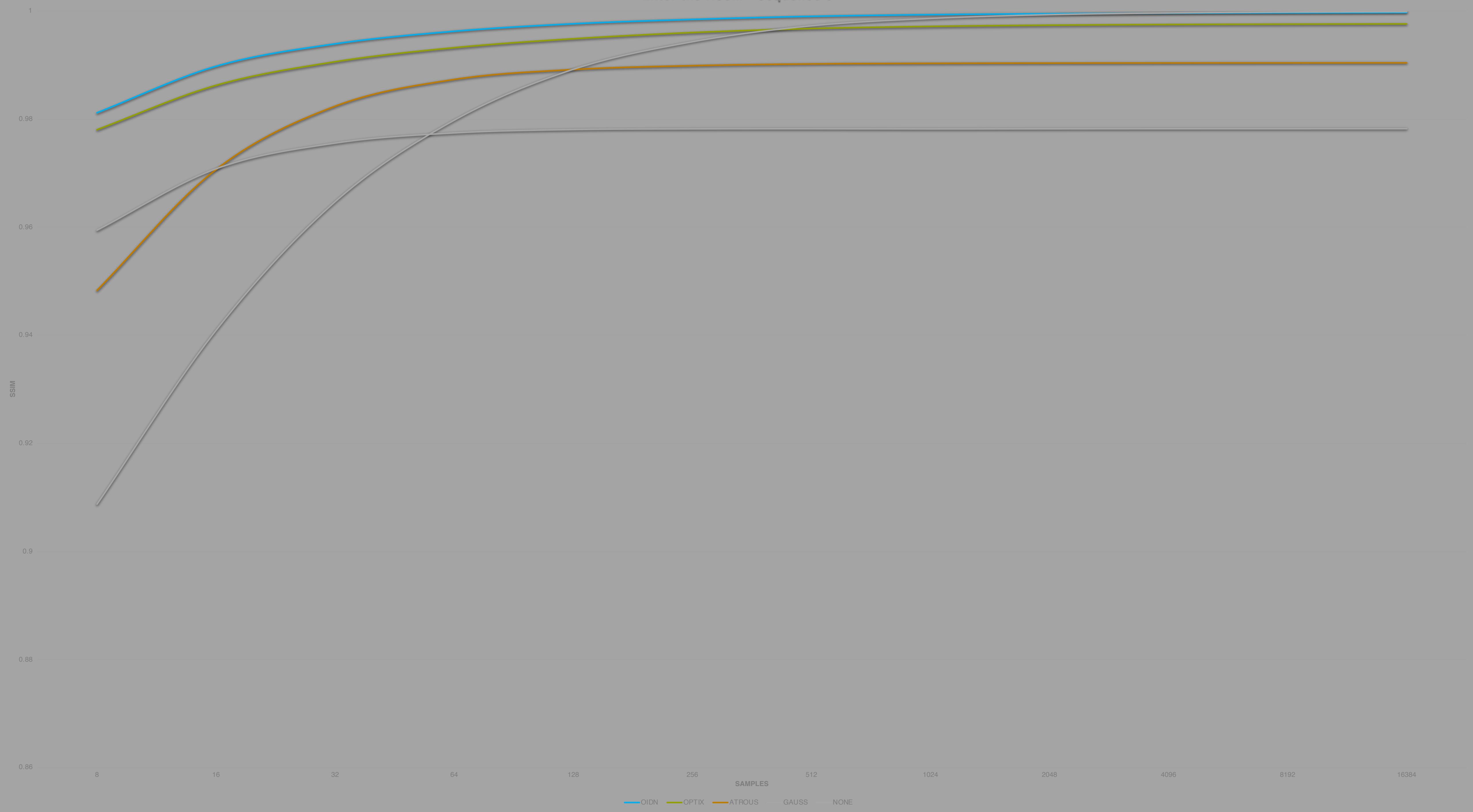
64 samples

OIDN





Enter the Room - Sequence 3



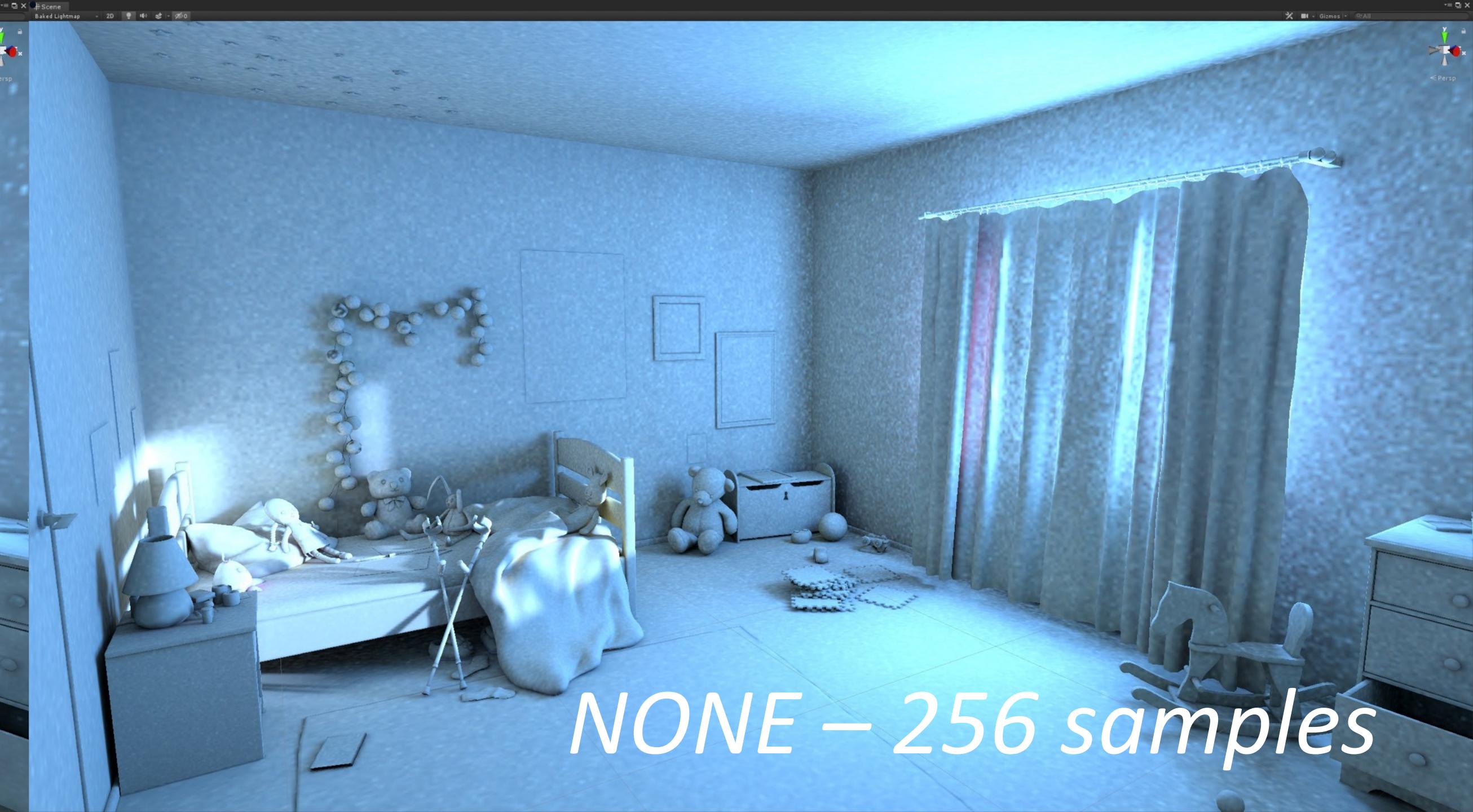
OIDN – 16 samples



OPTIX – 16 samples



ATROUS – 64 samples



NONE – 256 samples

OIDN – 16 samples



OPTIX – 16 samples



ATROUS – 64 samples



NONE – 256 samples





NONE – 4096 samples

OIDN – 1024 samples



NONE – 4096 samples



OIDN – 1024 samples

Open Image Denoise Benefits

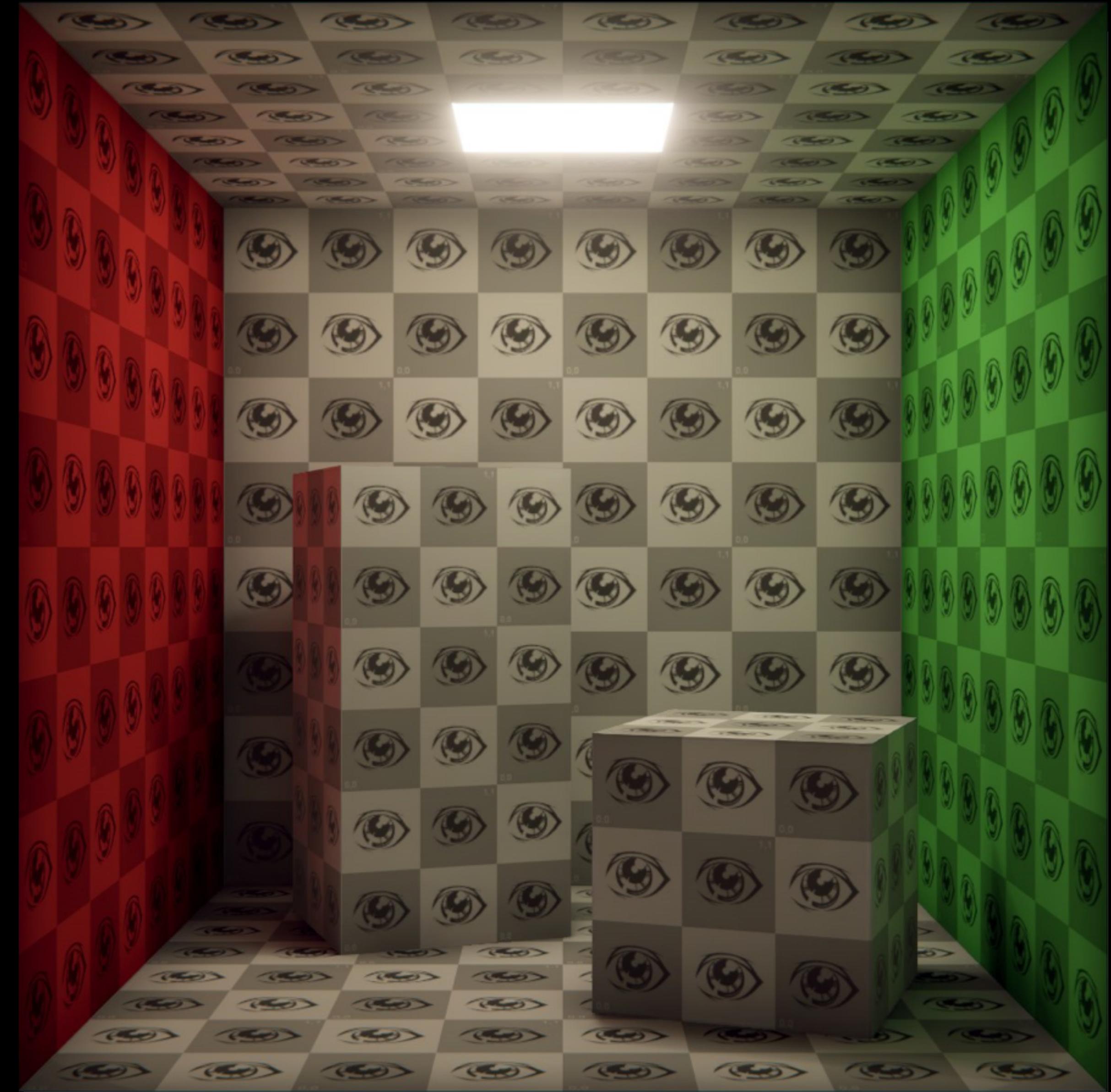
- No arcane knobs
- Can reduce samples by 4x+
- Works at low sample rates
- Can “fix” difficult lighting
- Integrates easily
- Platform/GPU agnostic
- We’re done!



#IntelGameDev

Future work

- Support normal+RGB mode
- Model trained on diffuse data
- Train directly on Unity assets
- Denoising directional data



#IntelGameDev

Acknowledgements

- Attila Afra, Intel
 - For assisting in integrating OIDN in Unity
- Carson Brownlee, Intel
 - For inviting us here today
- Bruno Samper, Nedd Agency and Red Cross
 - For letting us use “Enter The Room”
- Rasmus Rønn Nielsen
 - Rust code for SSIM/RMSE/RMAE metrics
- Kristijonas Jalnionis & David Llewelyn & Jennifer Nordwall, Unity
 - For preparing the assets used

Thank you for listening!

#unityatgdc

#IntelGameDev



Visit the
Unity Booth,
#S227

#unityatgdc

#IntelGameDev



The end...



#IntelGameDev

