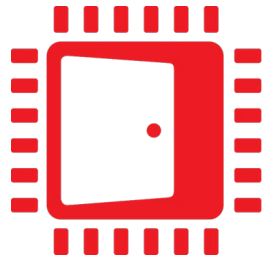
AMD
together we advance_

MESH SHADERS AND WORK GRAPHS - PERFECT PAIR

DR. ŁUKASZ IZDEBSKI
DEVELOPER TECHNOLOGY ENGINEER



AMD
GPUOpen

INTRODUCTION

- Mesh Shaders are getting more popular at Digital Dragons 2024
  - GPU-driven Rendering with Mesh Shaders in Alan Wake 2 by Erik Jansson (Remedy Entertainment) – Monday 20.03.2024 at 3:00 PM
  - Mesh Shaders – The Future of Rendering - Radosław Paszkowski (PixelAnt Games) – Monday 21.03.2024 1:00 PM

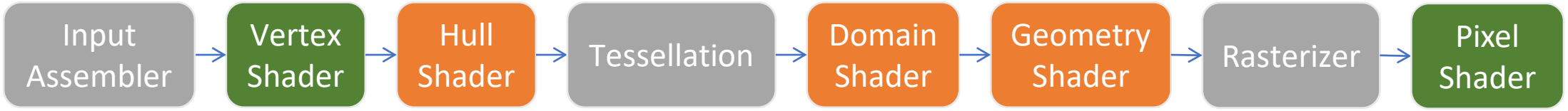
AGENDA

- Quick Mesh Shader technology introduction.
- Mesh Shader Example 1
- Mesh Shader Example 2
- Mesh Shader Example 3
- Graphics API new feature
- Mesh Shader Example 4
- Mesh Shader Example 5

SPOILER ALERT !!!
NO SHADER CODE INCLUDED

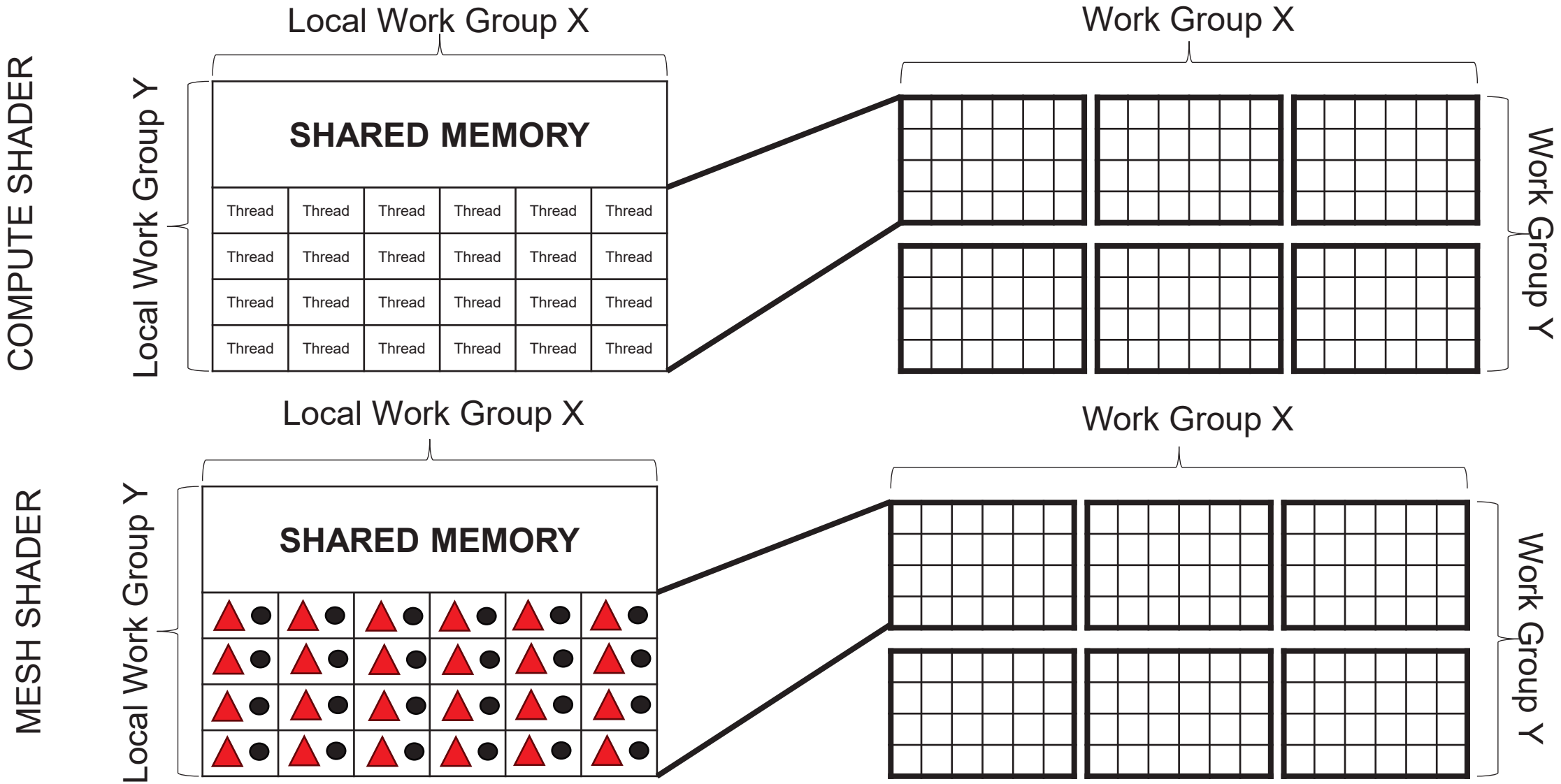
MESH SHADERS TECHNOLOGY 101

- Traditional Graphics Pipeline

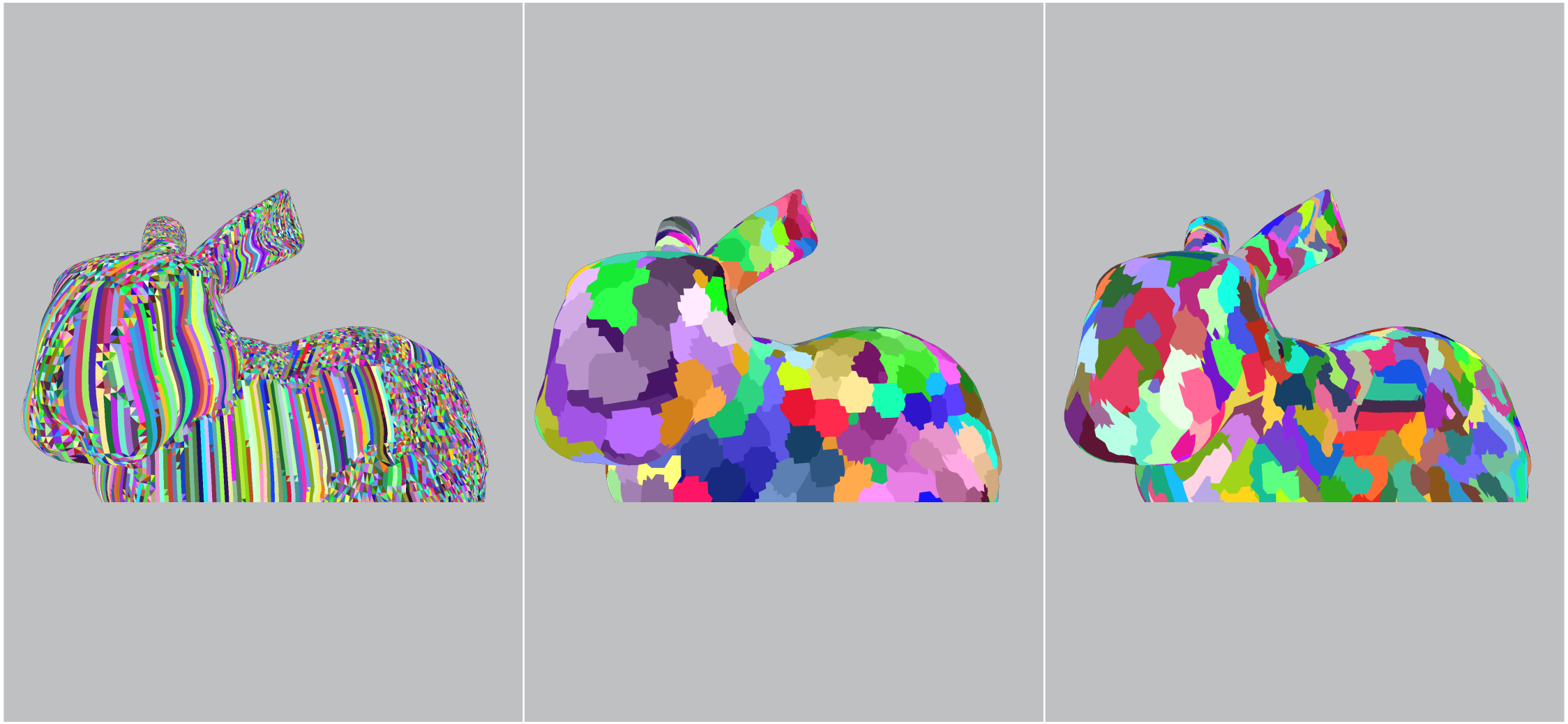


- Mesh Shader Pipeline

MESH SHADER VS COMPUTE SHADER



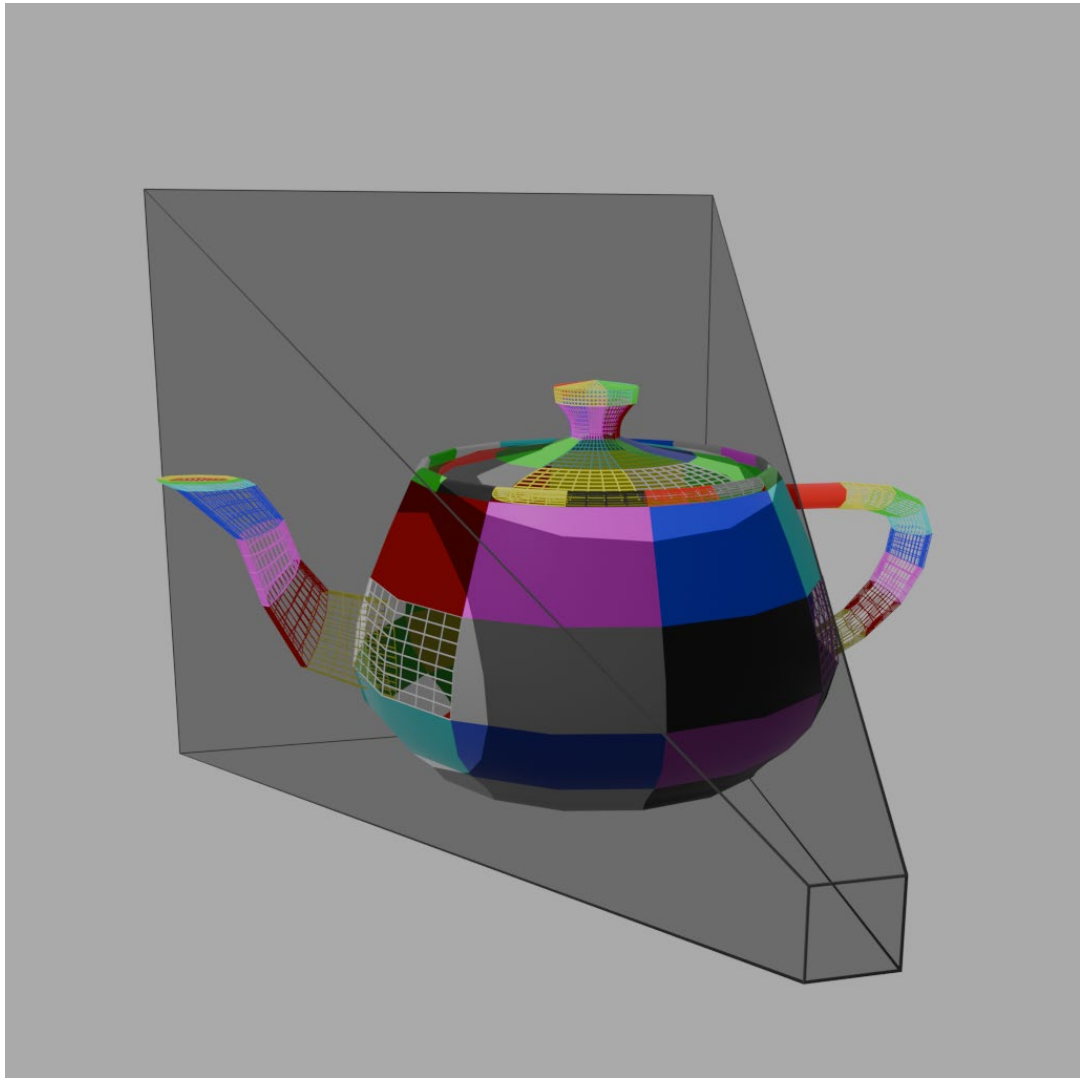
MESHLETS ZOO



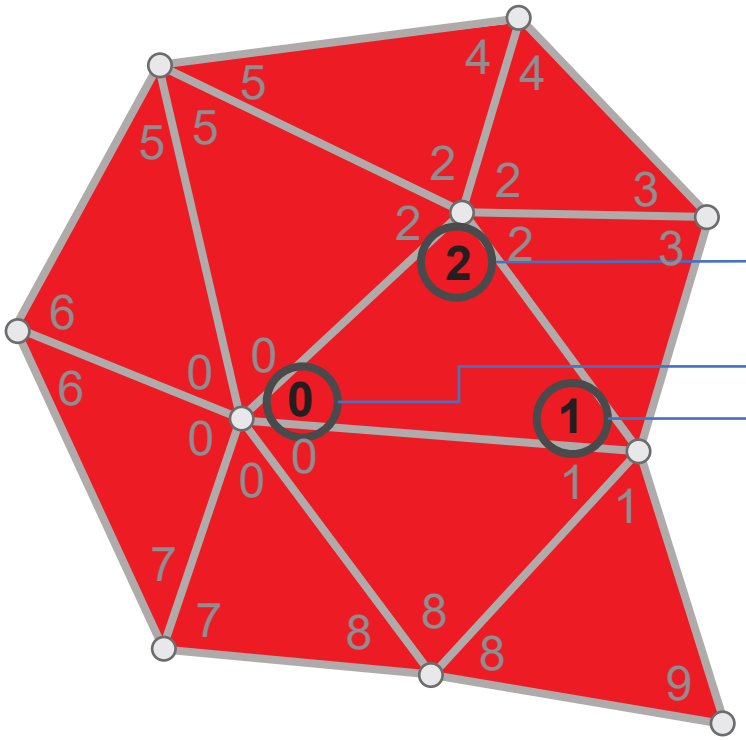
MESHLET GENERATION LIBRARIES

- More information about the generation of Meshlets:
  - https://gpuopen.com/learn/mesh_shaders/mesh_shaders-optimization_and_best_practices
  - https://meshoptimizer.org
  - https://github.com/microsoft/DirectXMesh

MESH SHADERS CULLING AND LODS



MESH SHADER COMPRESSION

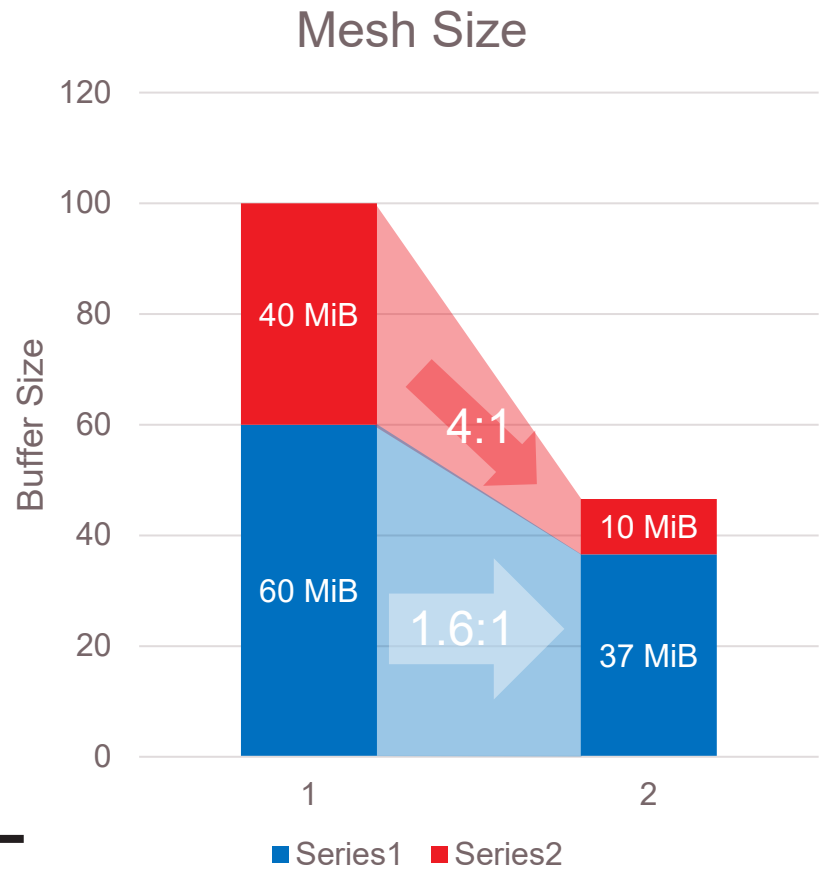


8 Bit Index

8 Bit Index

8 Bit Index

24 Bits per Triangle



MESH SHADER COMPRESSION

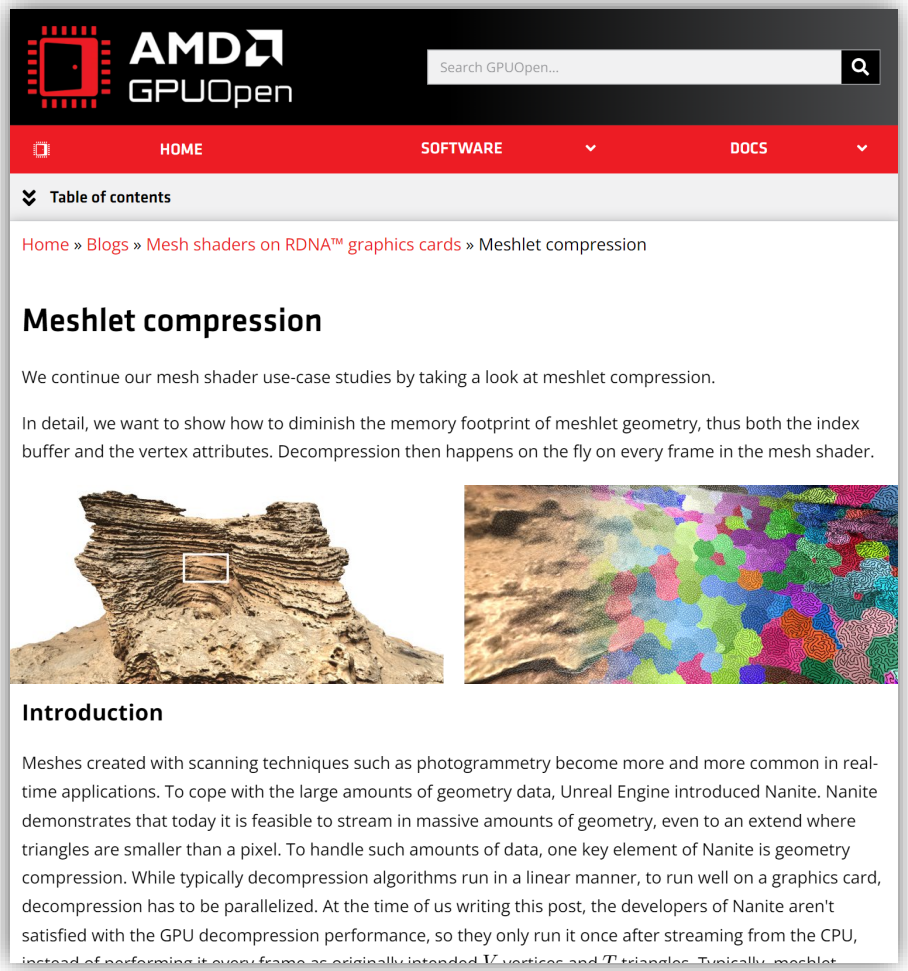
AMD GPUOpen

Keep an eye on GPUOpen.com

Follow us on X
https://twitter.com/GPUOpen

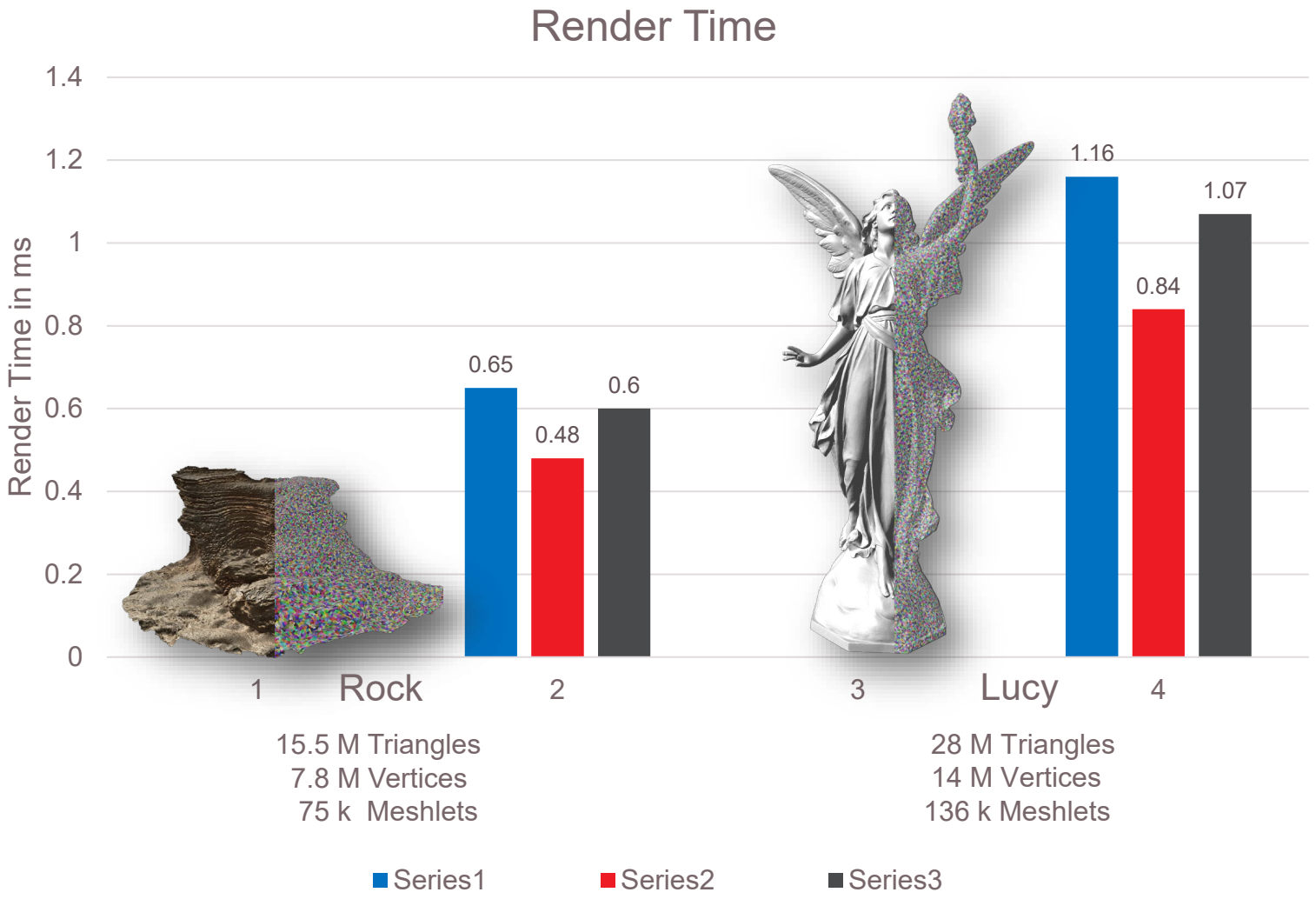
Follow us on Mastodon
https://mastodon.gamedev.place/@gpuopen

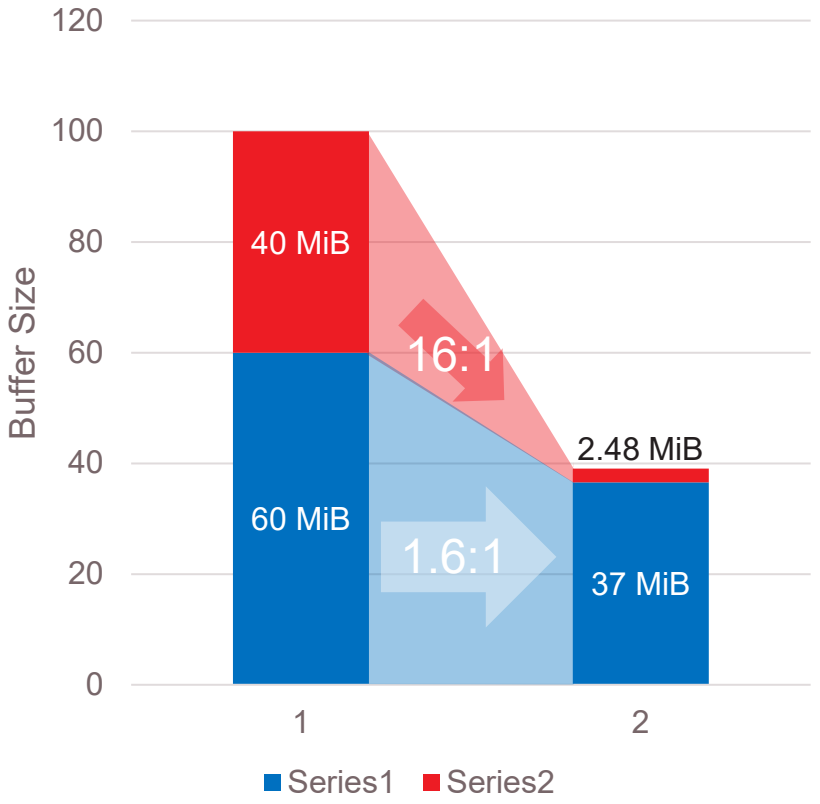
Follow us on Zhihu
https://www.zhihu.com/org/gpuopen-7



See endnotes for full test system configuration.

MESH SHADER COMPRESSION PERFORMANCE

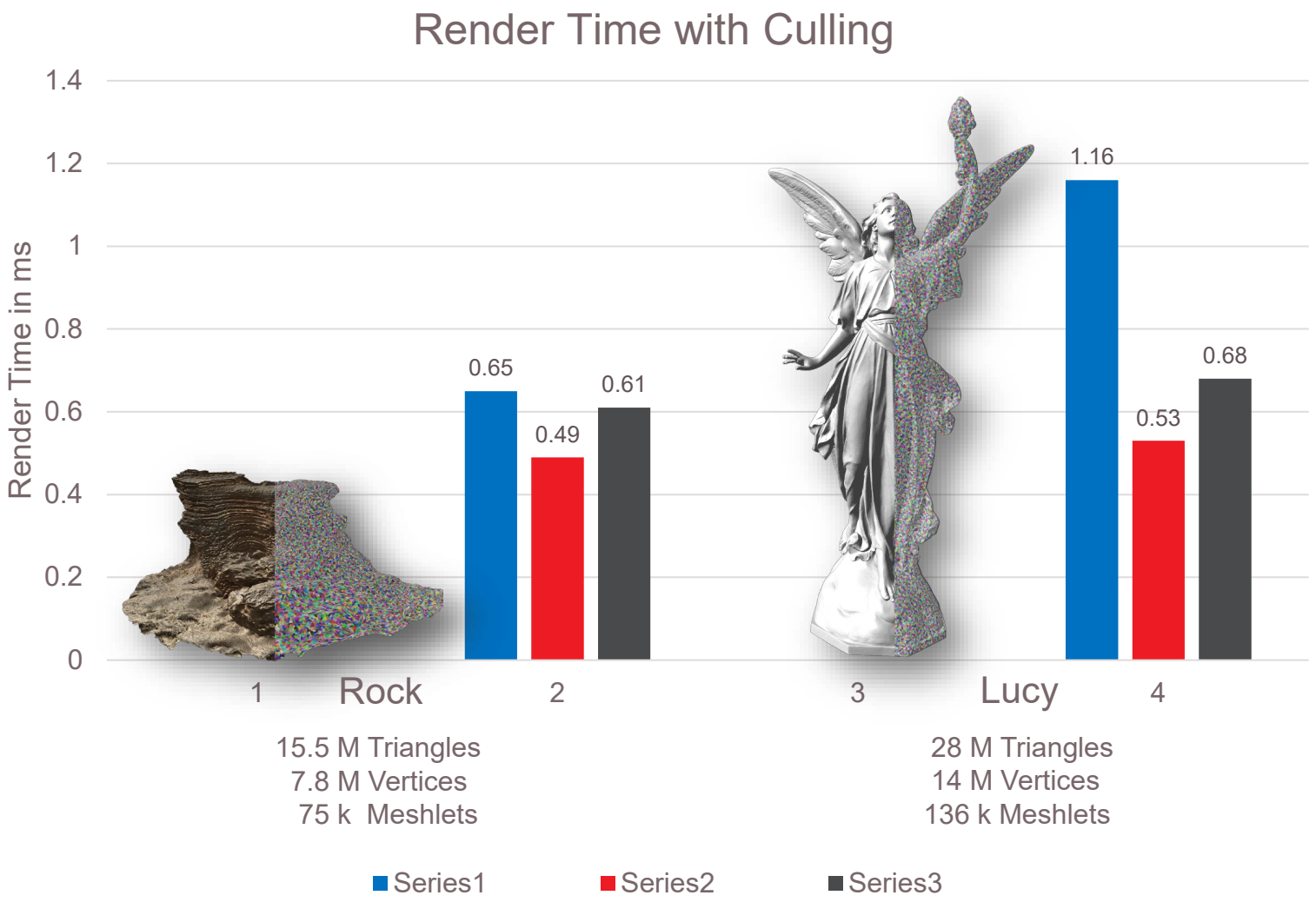


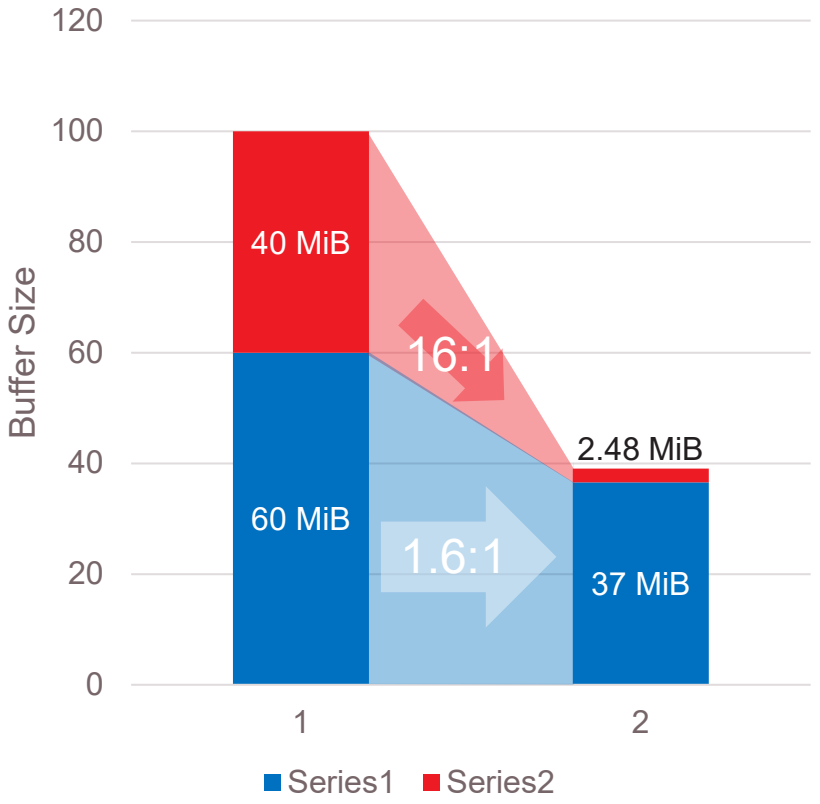


GPU: AMD Radeon™ RX 7900 XTX; driver version: 24.1.1; Meshes: Layered Rock by Aixterior, Lucy by Stanford Computer Graphics Laboratory

See endnotes for full test system configuration.

MESH SHADER COMPRESSION PERFORMANCE

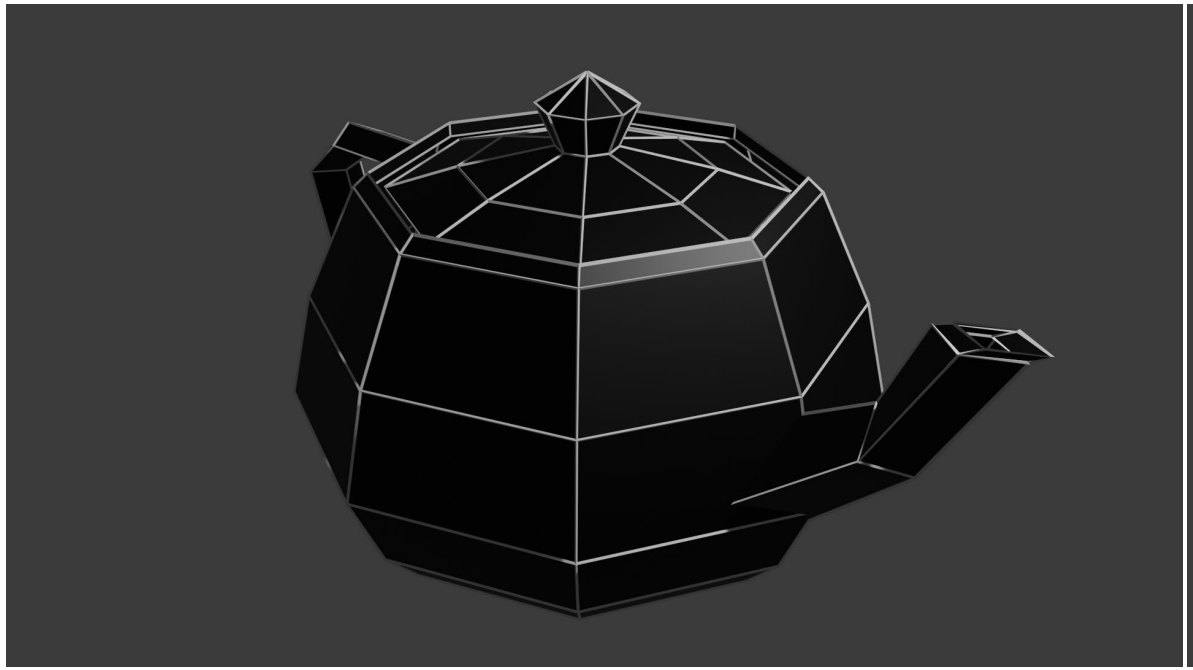




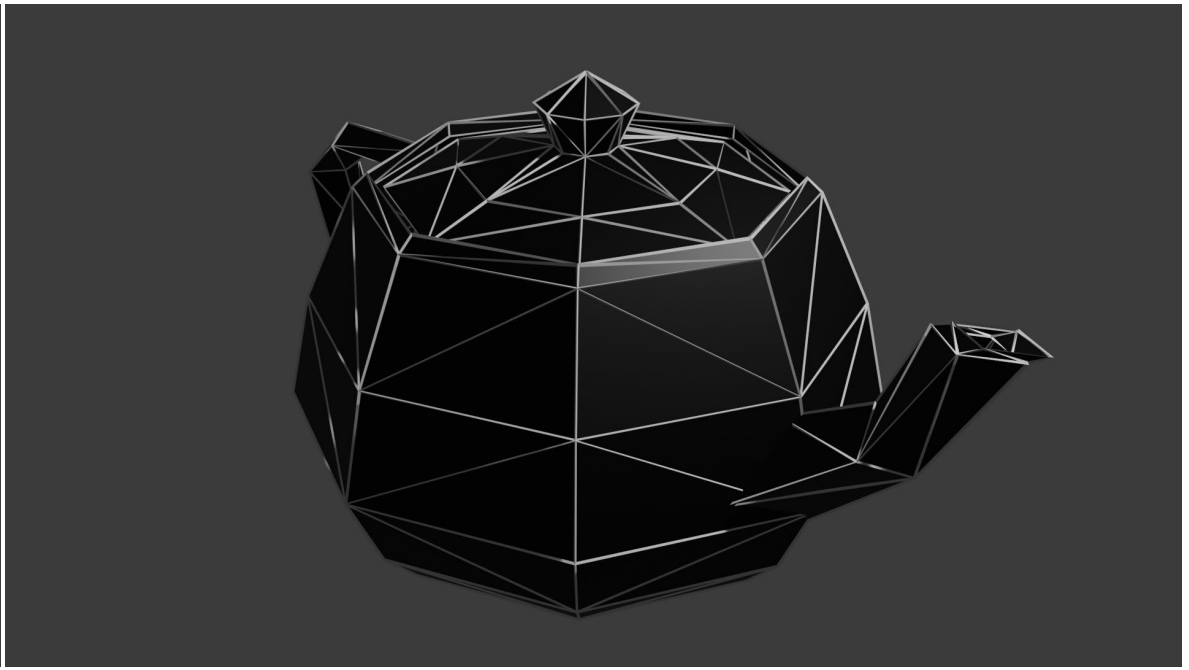
GPU: AMD Radeon™ RX 7900 XTX; driver version: 24.1.1; Meshes: Layered Rock by Aixterior, Lucy by Stanford Computer Graphics Laboratory

QUAD BASED AND TRIANGLE BASED MESH

QUAD BASED MESH



TRIANGLES BASED MESH



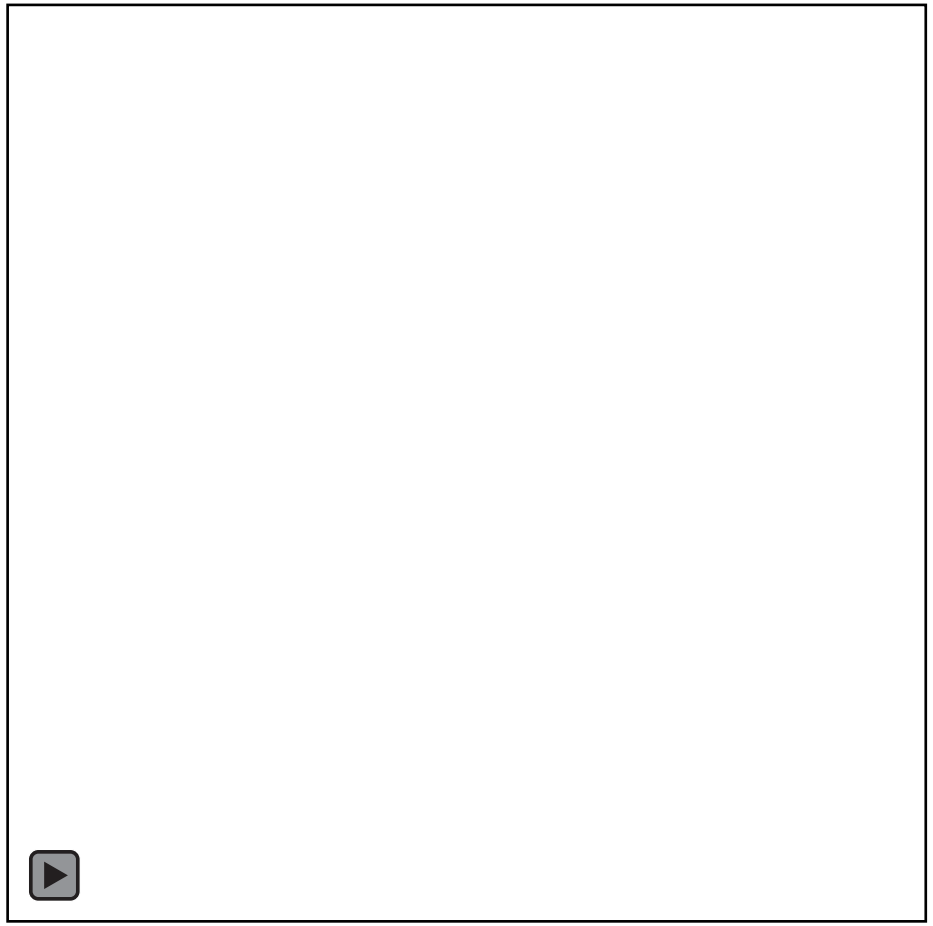
VERTEX ATTRIBUTES INTERPOLATION

BARYCENTRIC COORDINATES

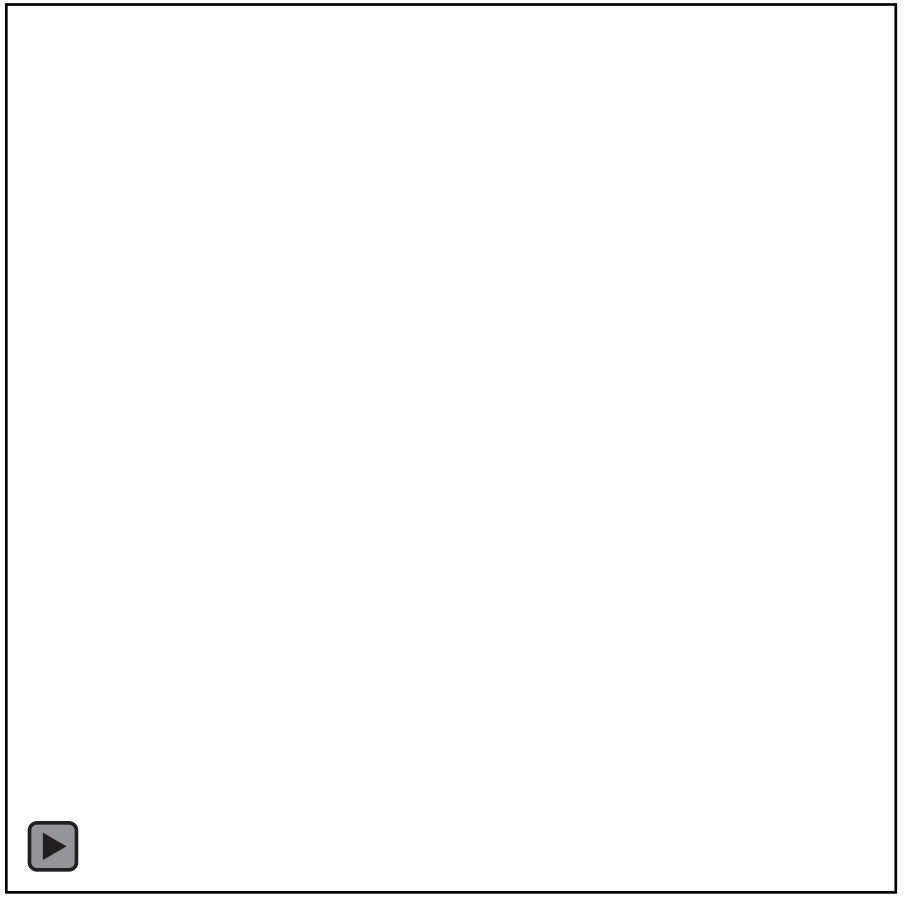
BILINEAR INTERPOLATION

INTERPOLATION AND ANIMATION

BARYCENTRIC INTERPOLATION



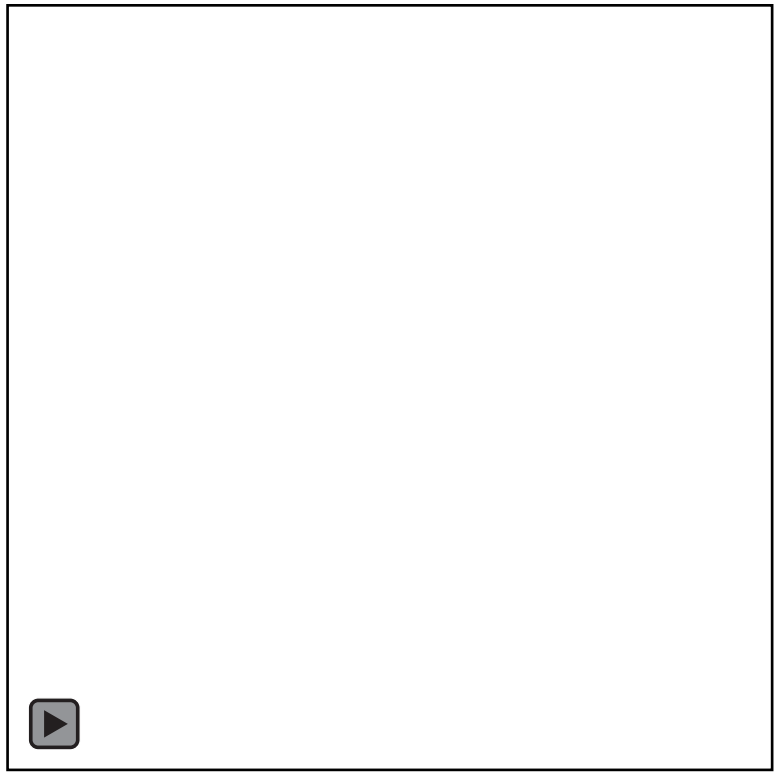
BILINEAR INTERPOLATION



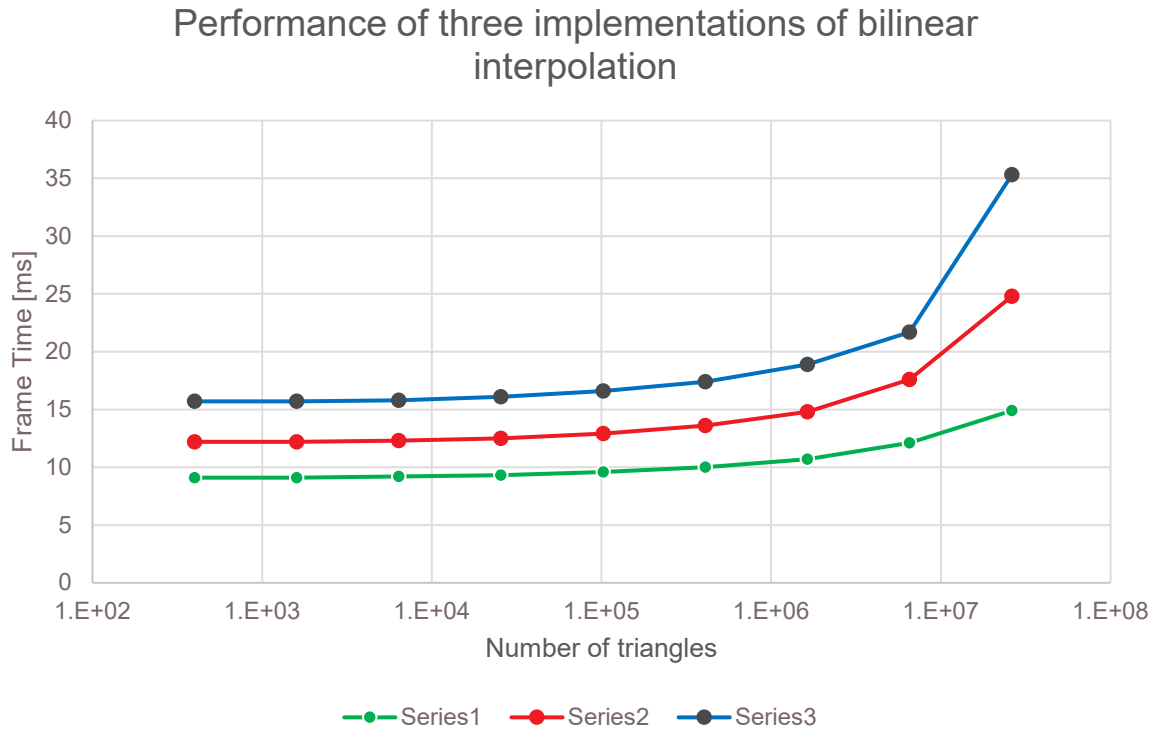
See endnotes for full test system configuration.

QUADRILATERAL PRIMITIVE RASTERIZATION

Bilinear Interpolation

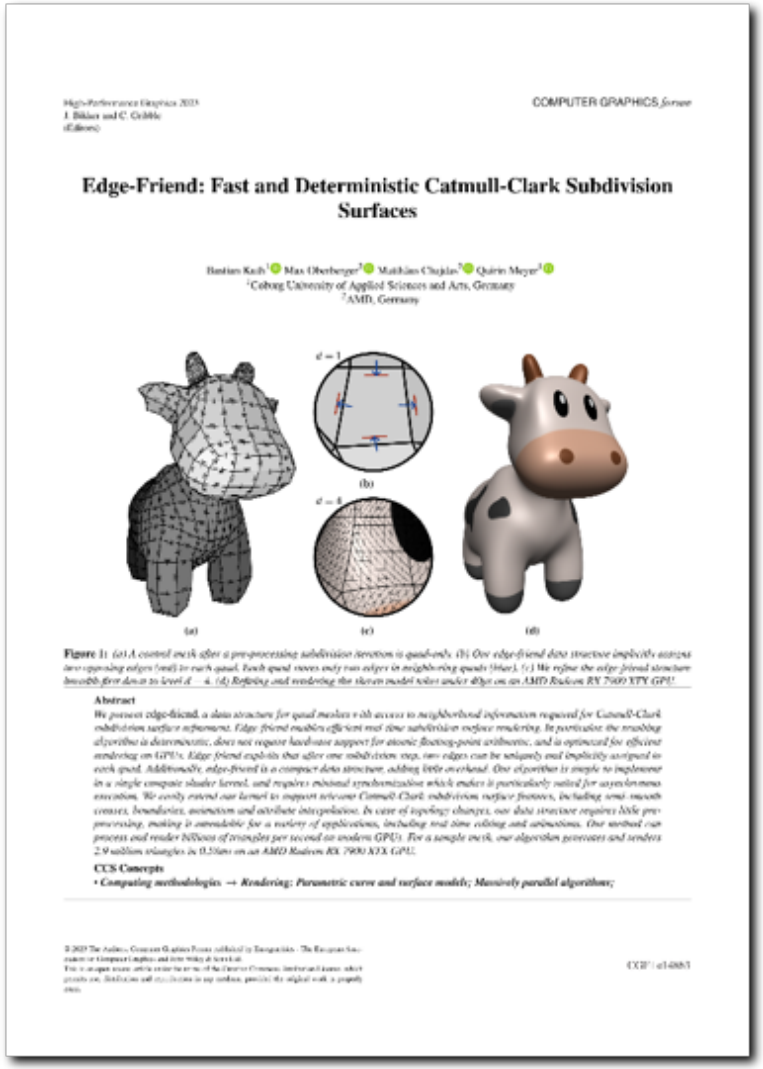


Faster than Tessellation/Geometry pipeline!



GPU: AMD Radeon™ RX 7900 XTX; driver version: 24.1.1

MESH SHADERS SUBDIVISION SURFACES



High-Performance Graphics 2023
J. Bikker and C. Gribble
(Editors)

COMPUTER GRAPHICS *forum*

Edge-Friend: Fast and Deterministic Catmull-Clark Subdivision Surfaces

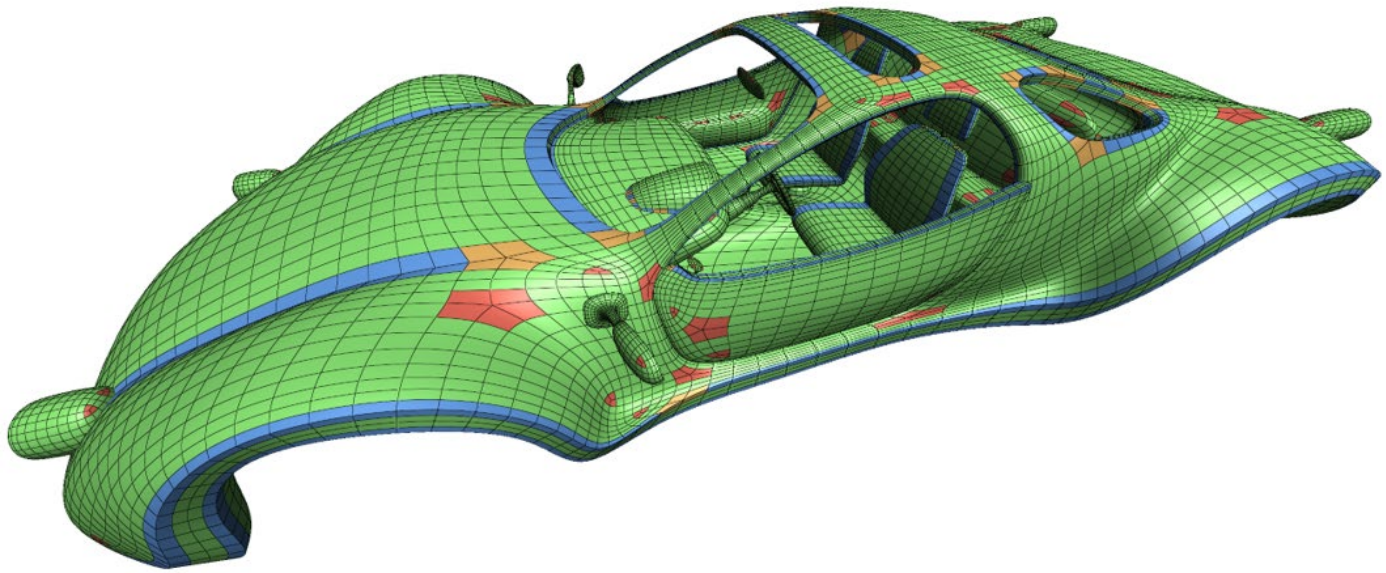
Bastian Kuth[1] Max Oberberger[2] Matthäus Chajdas[2] Quirin Meyer[1]
[1]Coburg University of Applied Sciences and Arts, Germany
[2]AMD, Germany

Abstract

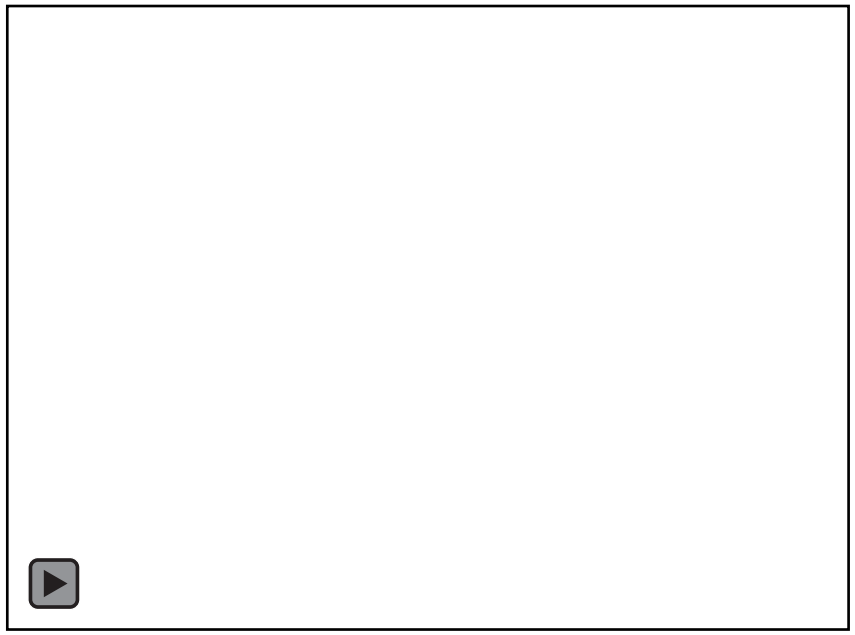
We present edge-friend, a data structure for quad meshes with access to neighborhood information required for Catmull-Clark subdivision surface refinement. Edge-friend enables efficient real-time subdivision surface rendering. In particular the resulting algorithm is deterministic, does not require hardware support for atomic floating-point arithmetic, and is optimized for efficient rendering on GPUs. Edge-friend exploits that after one subdivision step, two edges can be uniquely and implicitly assigned to each quad. Additionally, edge-friend is a compact data structure, adding little overhead. Our algorithm is simple to implement in a single compute shader kernel, and requires minimal synchronization which makes it particularly suited for asynchronous execution. We easily extend our kernel to support relevant Catmull-Clark subdivision surface features, including semi-smooth creases, boundaries, quantum and attribute interpolation. In case of topology changes, our data structure requires little preprocessing, making it amendable for a variety of applications, including real-time editing and animations. Our method can process and render billions of triangles per second on modern GPUs. For a single mesh, our algorithm generates and renders 2.9 million triangles in 0.53ms on an AMD Radeon RX 7900 XTX GPU.

CCS Concepts

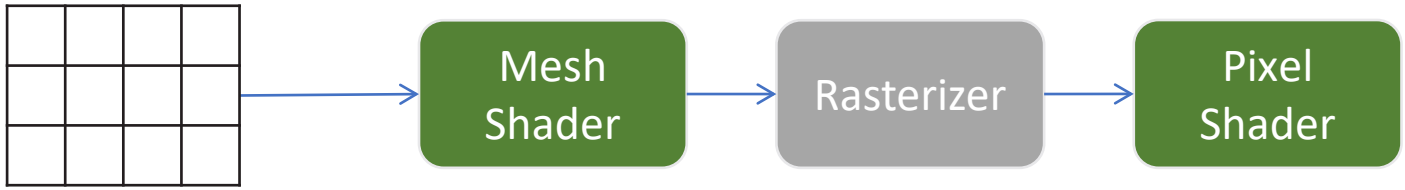
• *Computing methodologies* → *Rendering; Parametric curve and surface models; Massively parallel algorithms;*



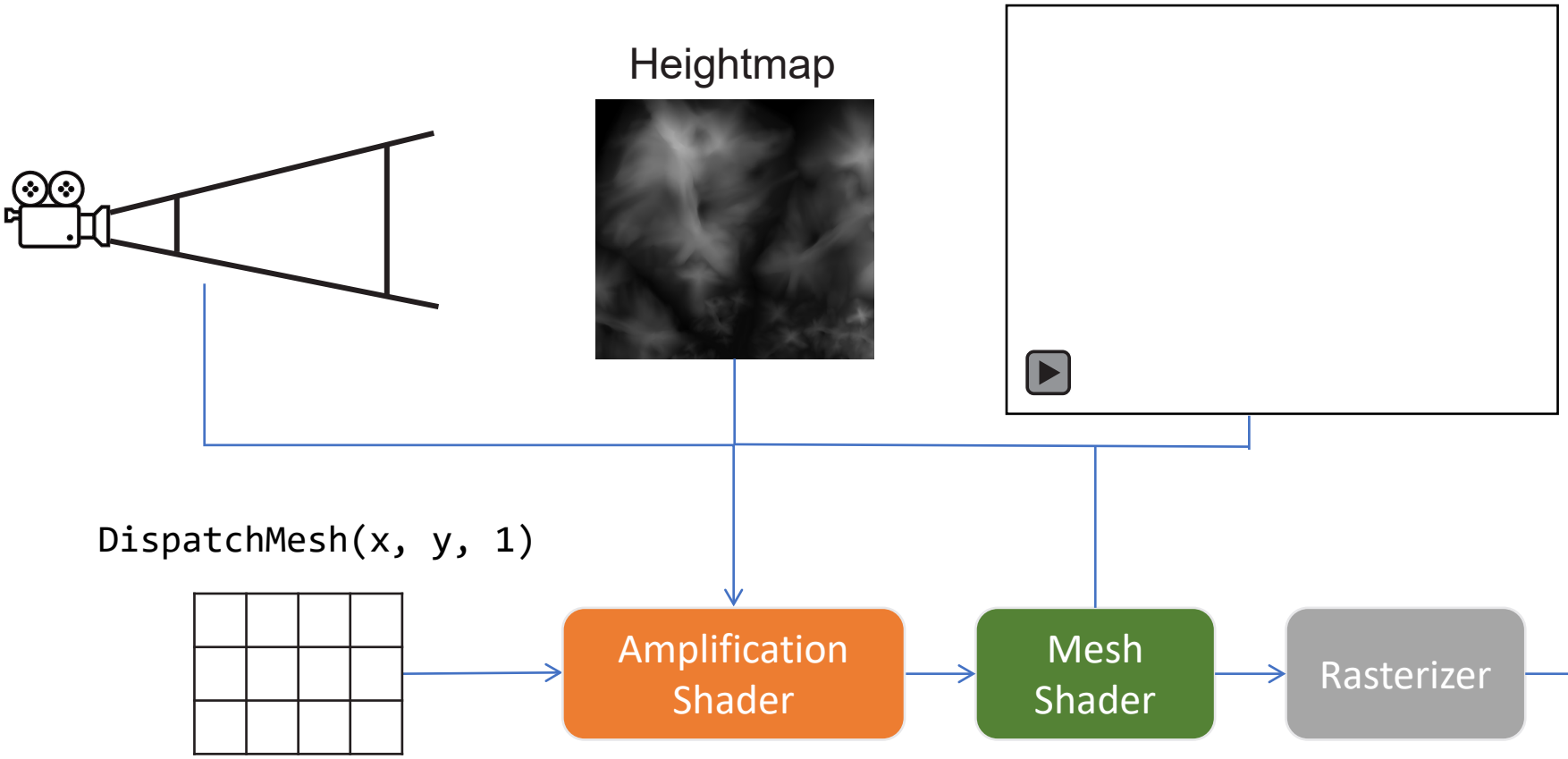
PROCEDURAL GEOMETRY



```
DispatchMesh(1, 1, 1)
```



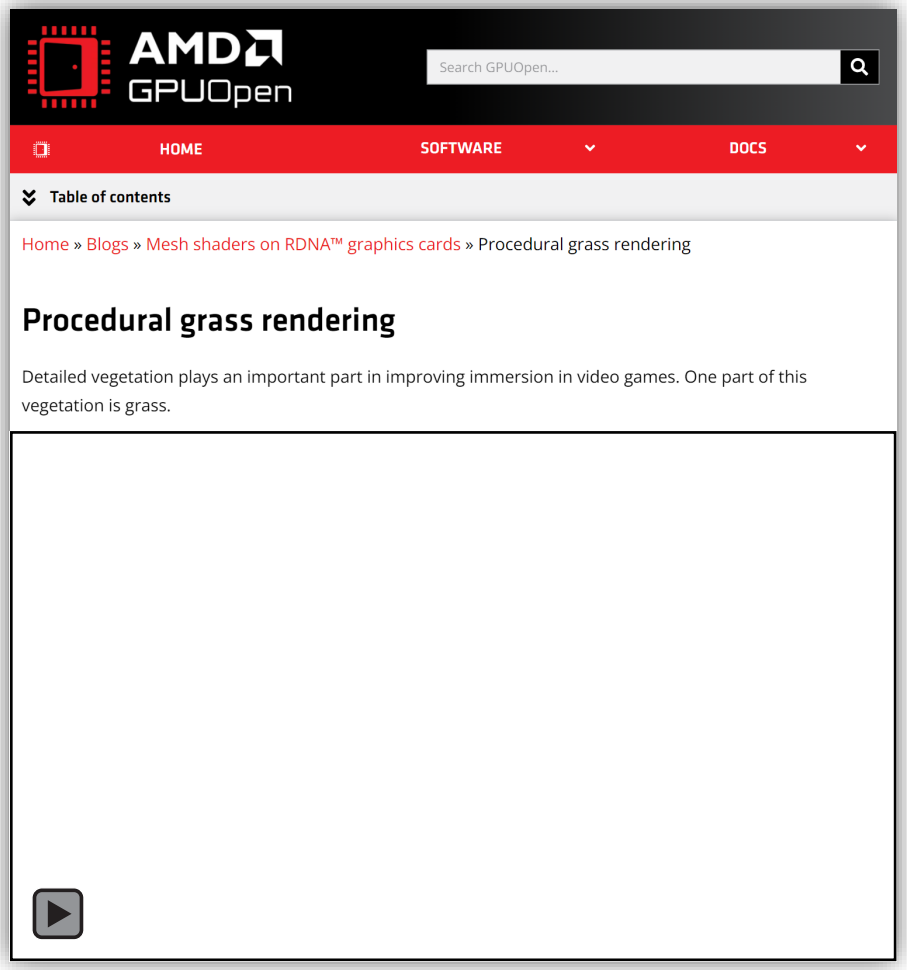
PROCEDURAL GEOMETRY



PROCEDURAL GEOMETRY

AMD GPUOpen

Learn more on GPUOpen.com



Follow us on X
https://twitter.com/GPUOpen

Follow us on Mastodon
https://mastodon.gamedev.place/@gpuopen

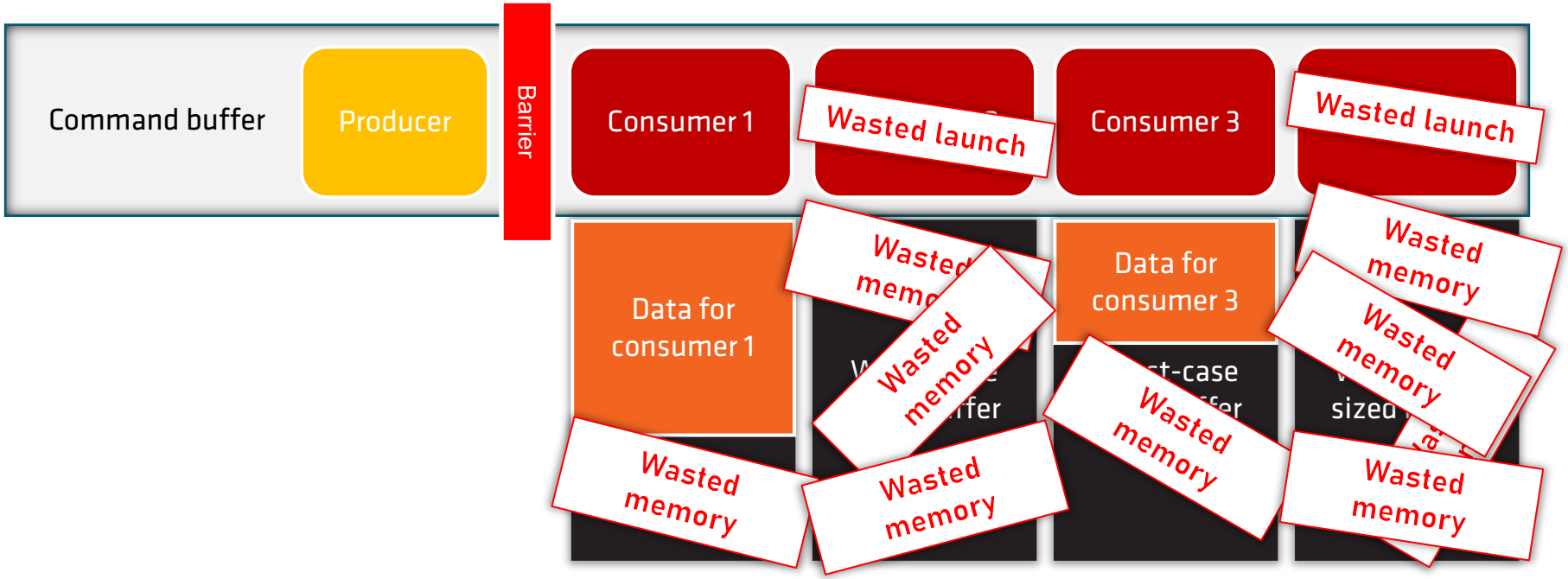
Follow us on Zhihu
https://www.zhihu.com/org/gpuopen-7

WORK GRAPHS 1.0

- Multi-year collaboration between Microsoft and industry partners
- Developer preview released June 2023
- Out of preview from March 2024
- https://devblogs.microsoft.com/directx/d3d12-work-graphs/

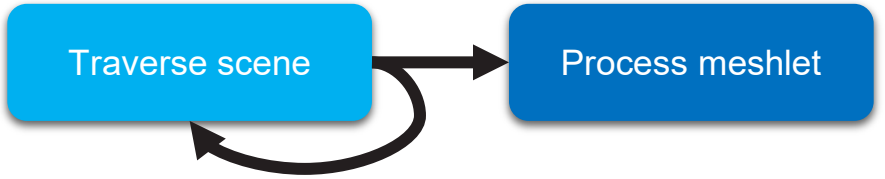
EXECUTE INDIRECT OVERHEAD

- Using execute indirect can introduce memory and performance overhead.

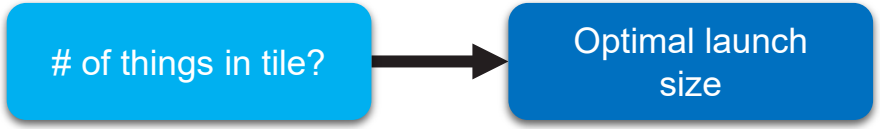


WORK GRAPHS AWSER FOR EXECUTE INDIRECT LIMITATIONS

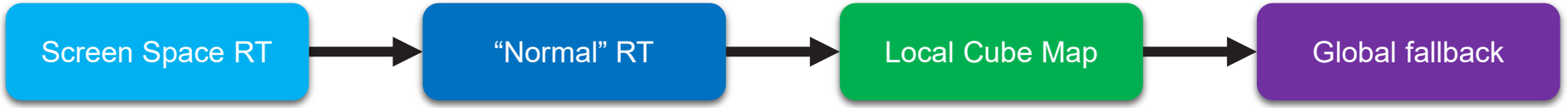
- Recursive algorithms: Scene traversal, ...



- Adaptive algorithms (launch more/less work): Physics, ...

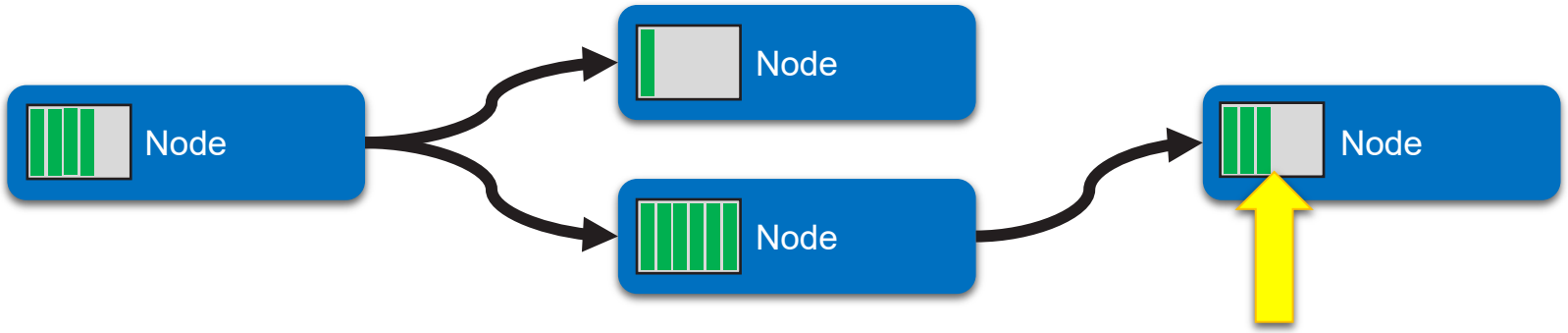


- Long execution chains: Lighting algorithms, ...



WORK GRAPHS IN A NUTSHELL

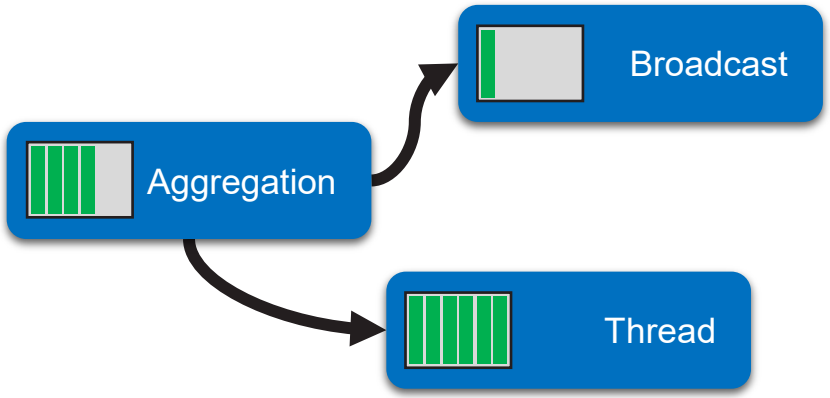
- **Nodes** connected with **edges**
- Each node has a virtual **queue**
- Nodes launch as soon as “enough” work waits for them
  - Enough depends on the GPU, driver, ...
  - Runtime can merge/fuse nodes, reorder outputs, sort, etc.



Scheduler launches dispatch to consume the data

GPU WORK GRAPHS

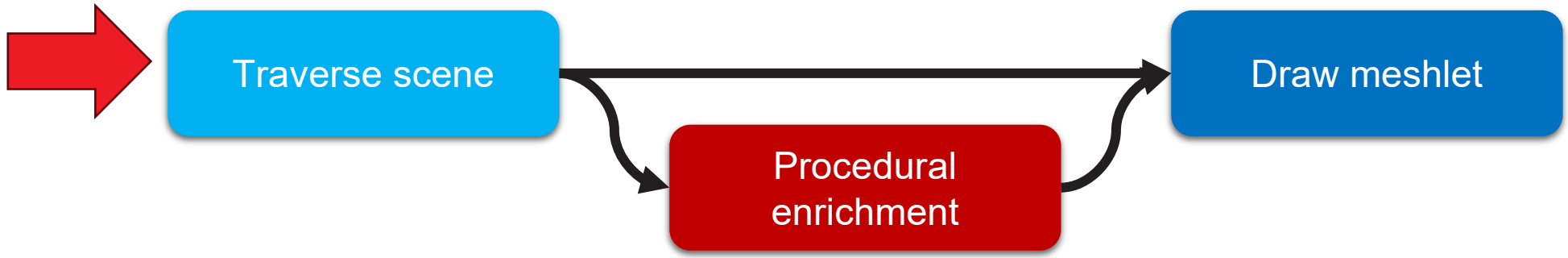
- a data flow model
- Work moves from node to node in the form of small “work items” (think: a `struct`)
- Work items get “queued up”
- Once enough work is pending,
  the GPU launches a dispatch



WORK GRAPHS – DRAW NODES

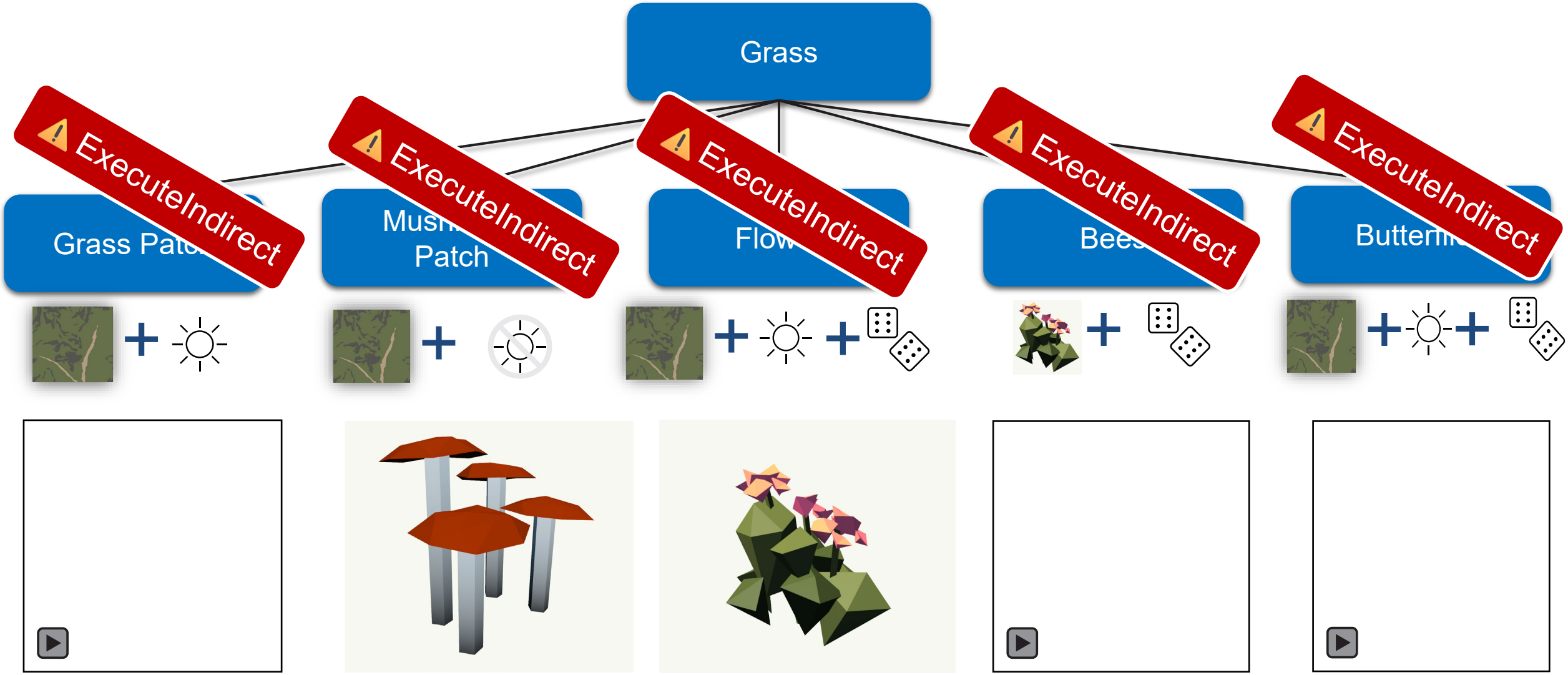
- **Preview feature** announcement: Draw nodes
- Draw “inside” the work graph using “draw nodes”
- Enables fully compute-driven scene traversal (**with PSO switching**)
- Draw nodes: Feed into a **mesh shader pipeline**
- Work graph acts like an amplification shader on steroids

All in **one** graph



“THE MEADOW” EXAMPLE

GRASS GENERATION WITHOUT WORK GRAPHS



GRASS GENERATION WITH MESH SHADERS

"🚫 No `ExecuteIndirect`" one

PREVIEW

Grass

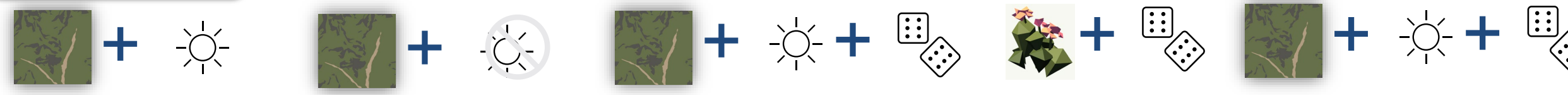
Grass Patch

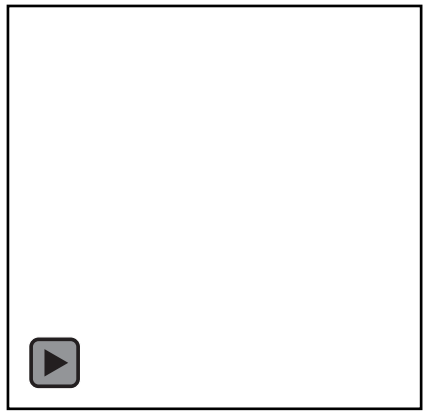
Mushroom Patch

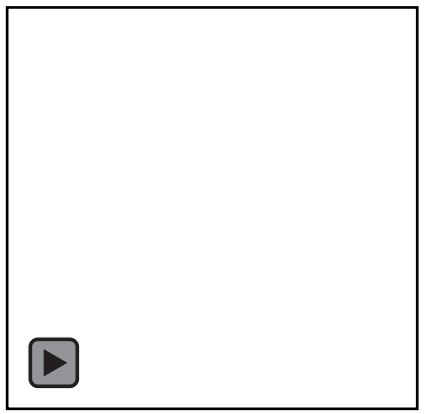
Flower

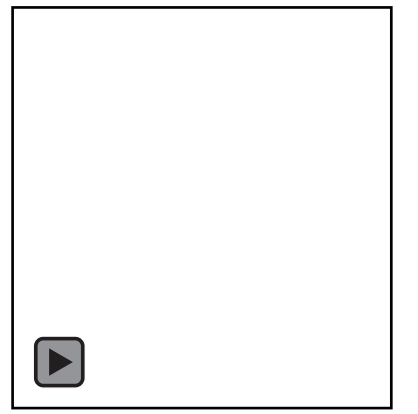
Bees

Butterflies



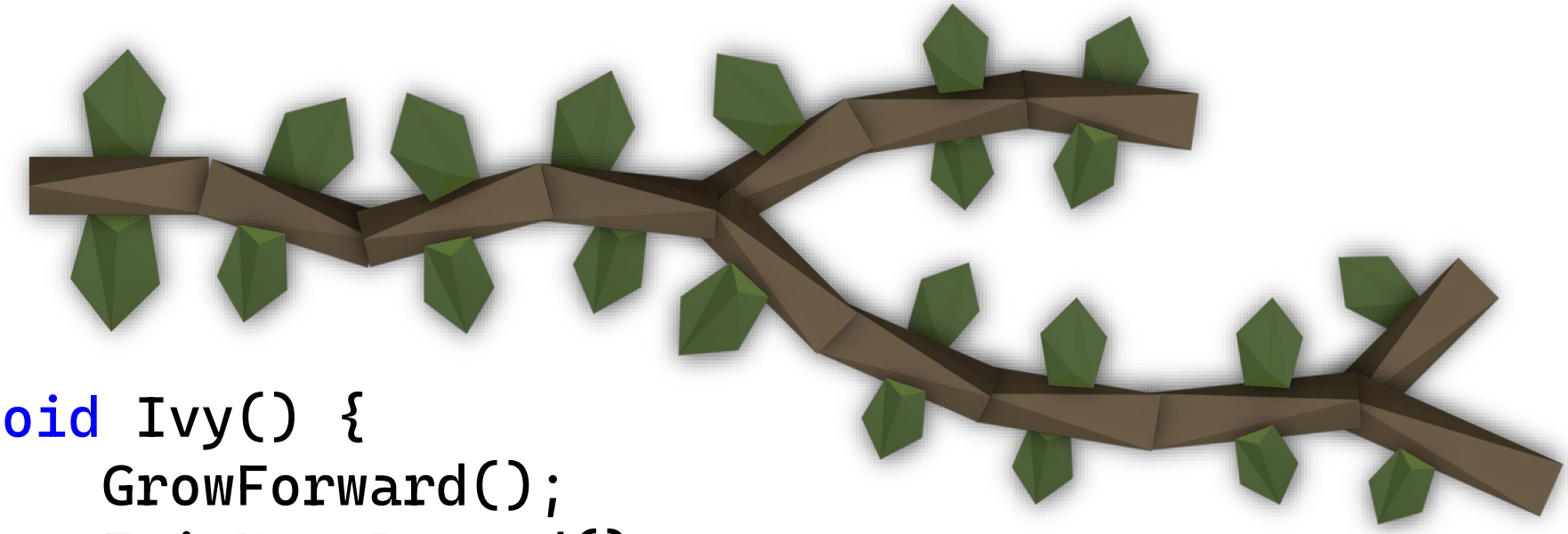






“THE BRIDGE” EXAMPLE

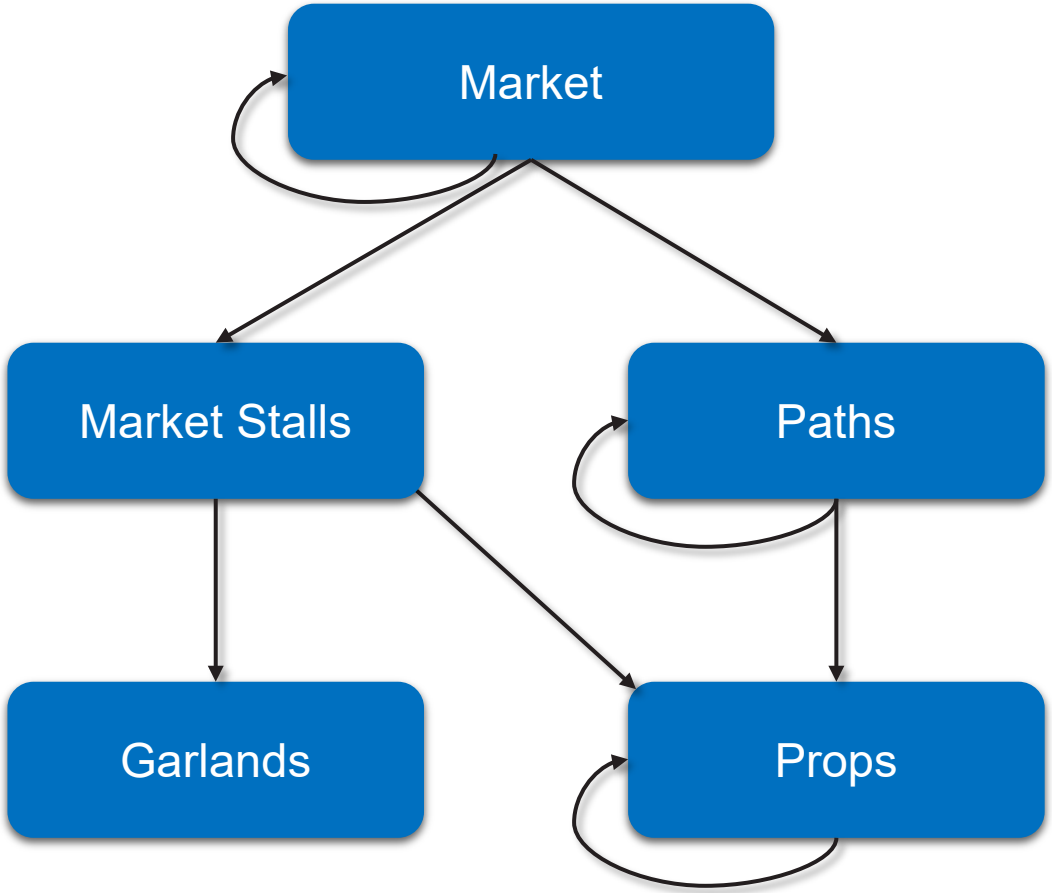
RECURSIVE ALGORITHM EXAMPLE



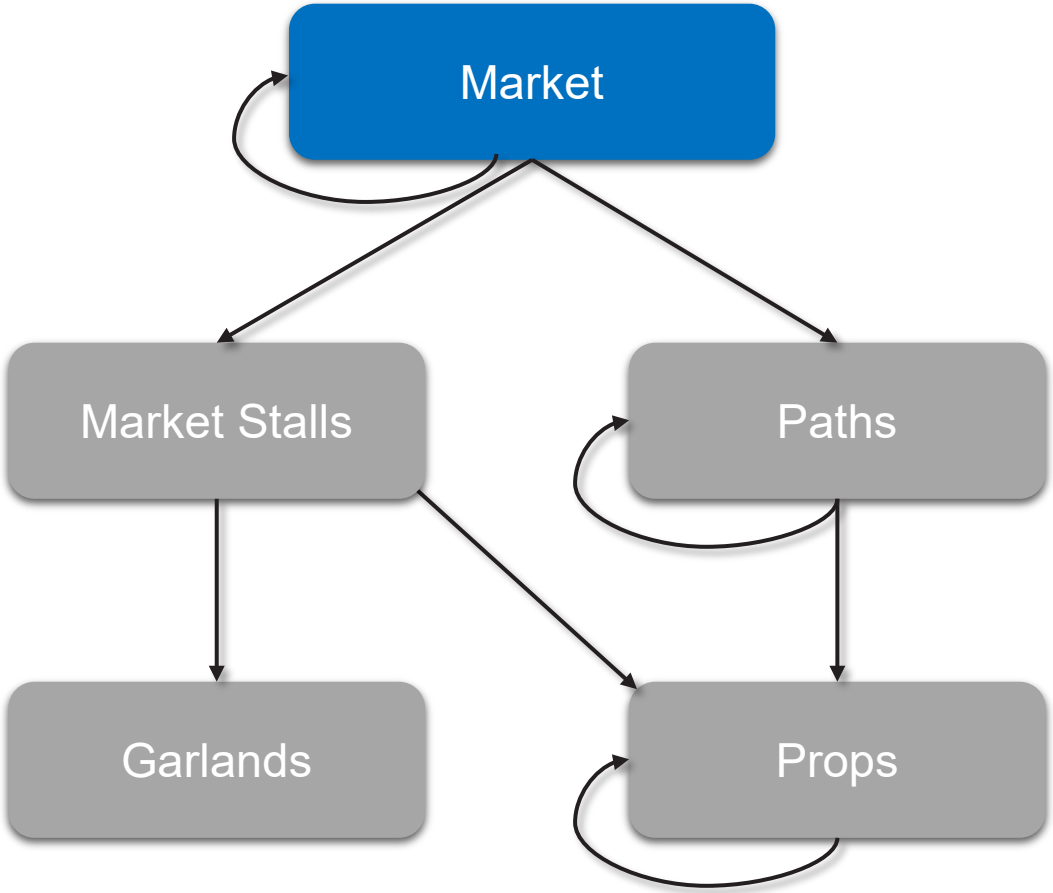
```
void Ivy() {
    GrowForward();
    EmitNextRecord();

    if (forked) {
        EmitNextRecord();
    }
}
```

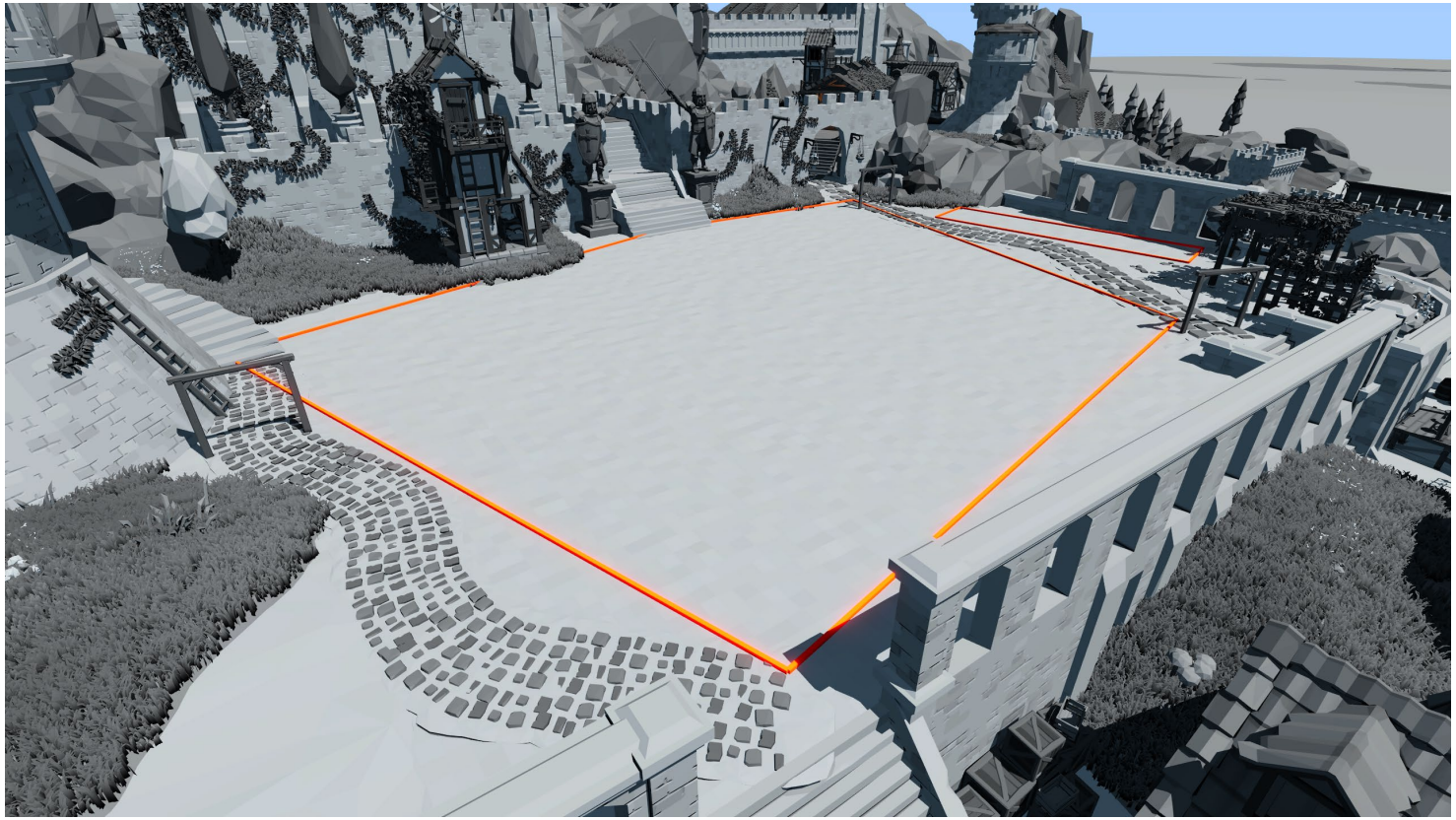
“THE MARKET” EXAMPLE

PROCEDURAL GENERATION EXAMPLE

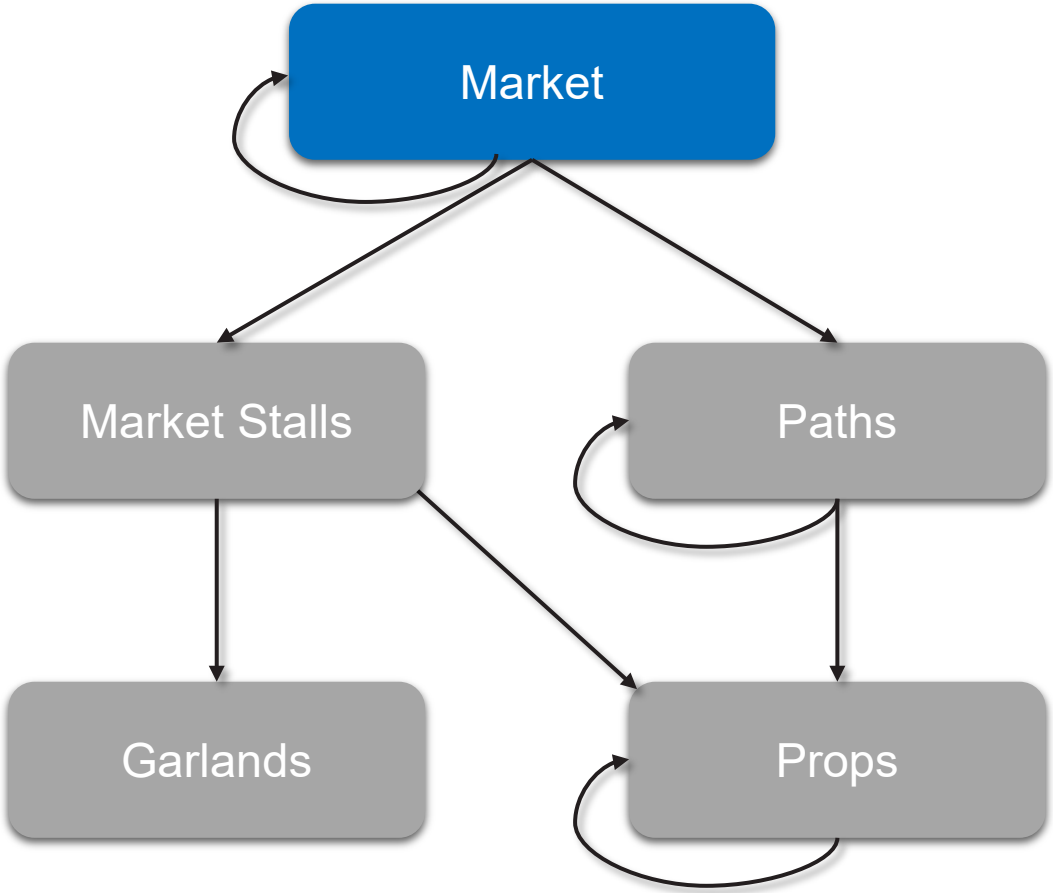


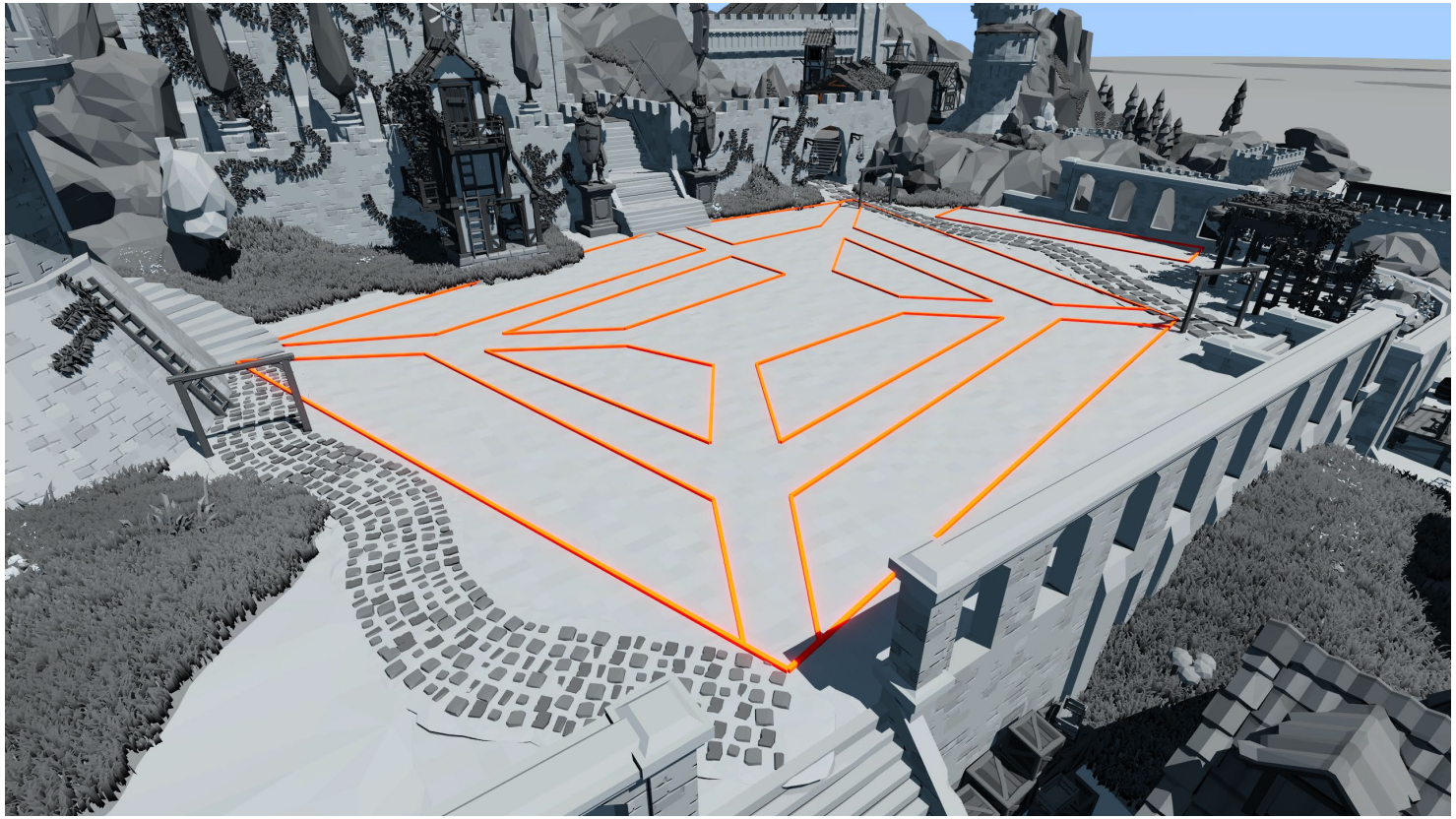
PROCEDURAL GENERATION EXAMPLE



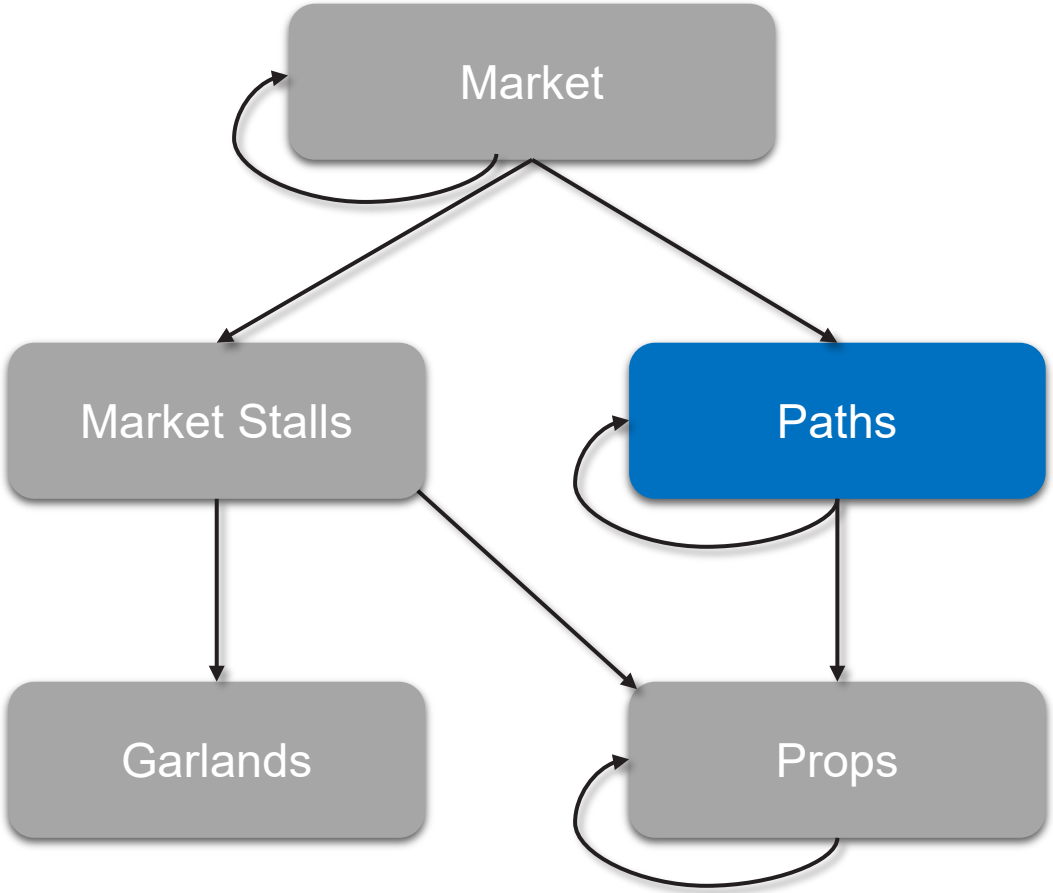


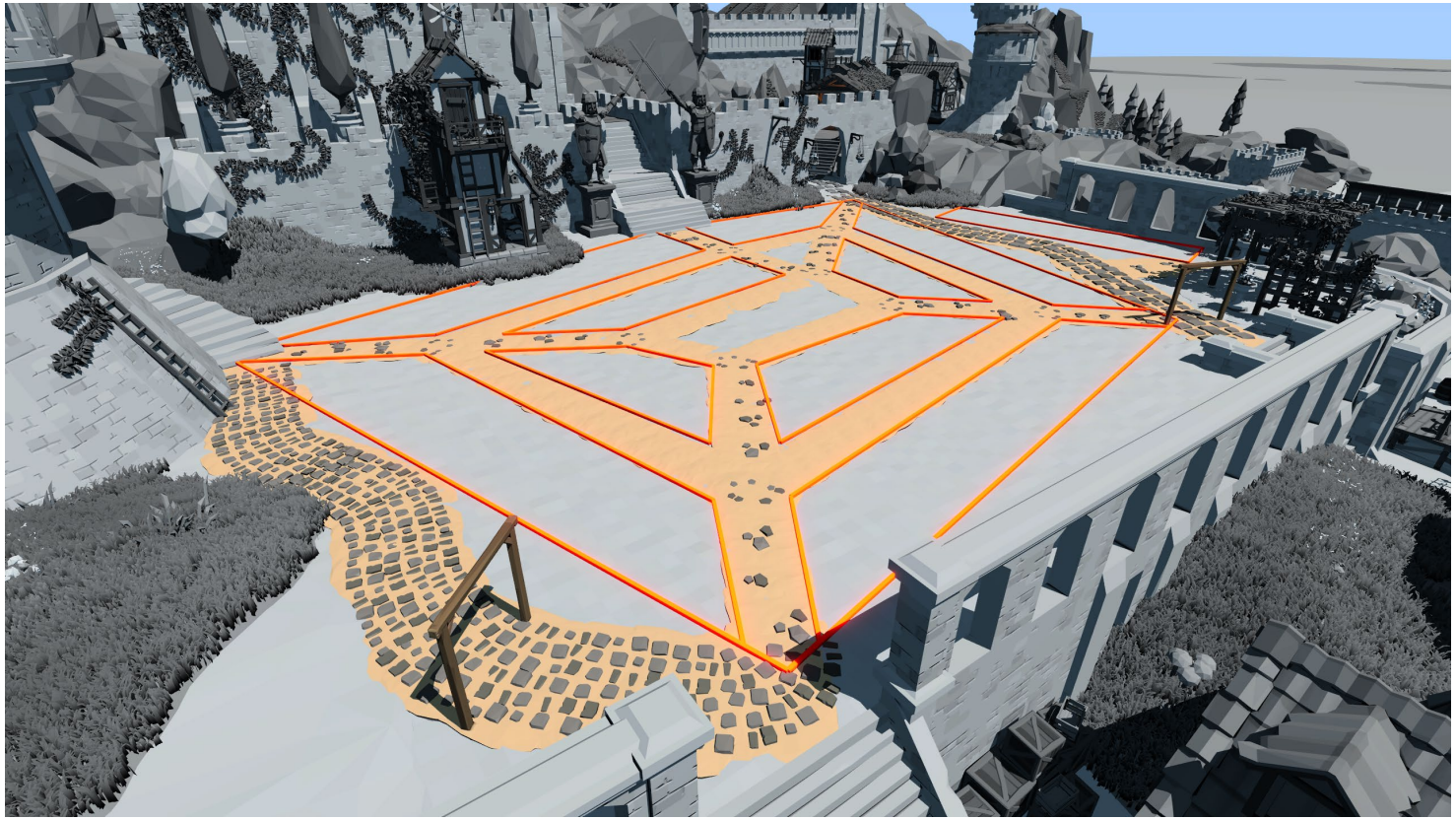
PROCEDURAL GENERATION EXAMPLE



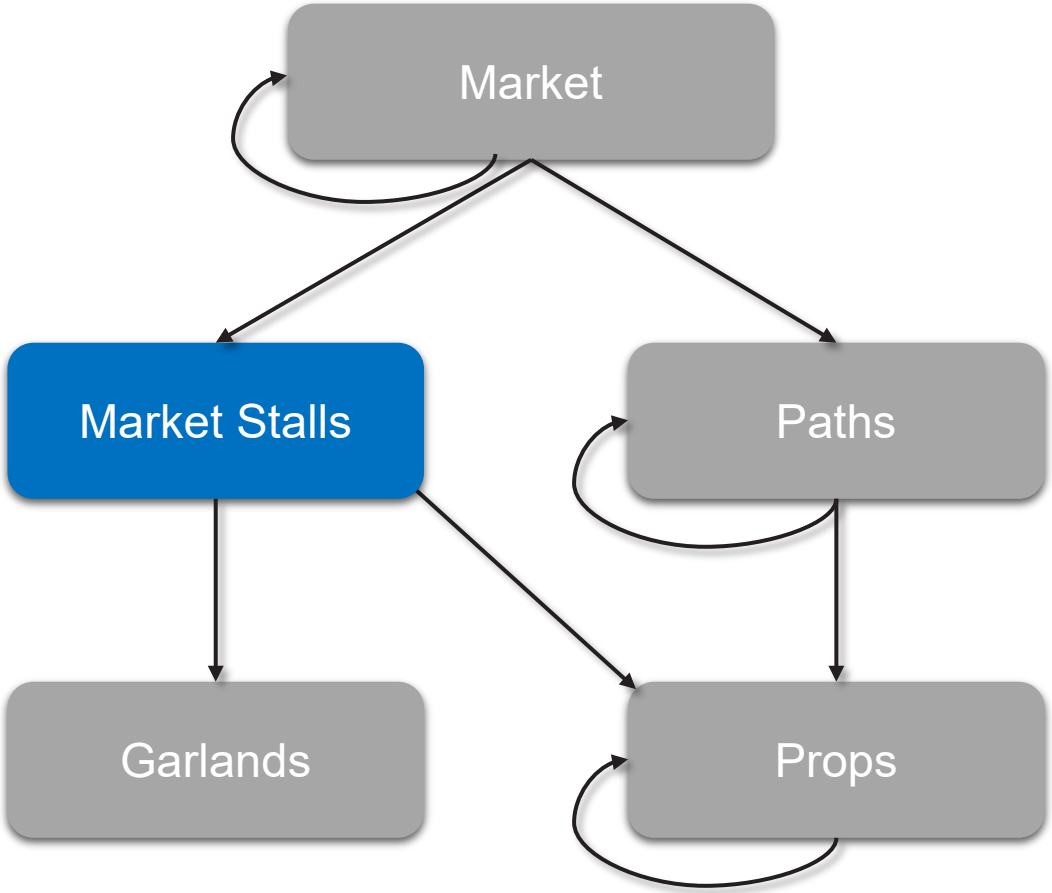


PROCEDURAL GENERATION EXAMPLE

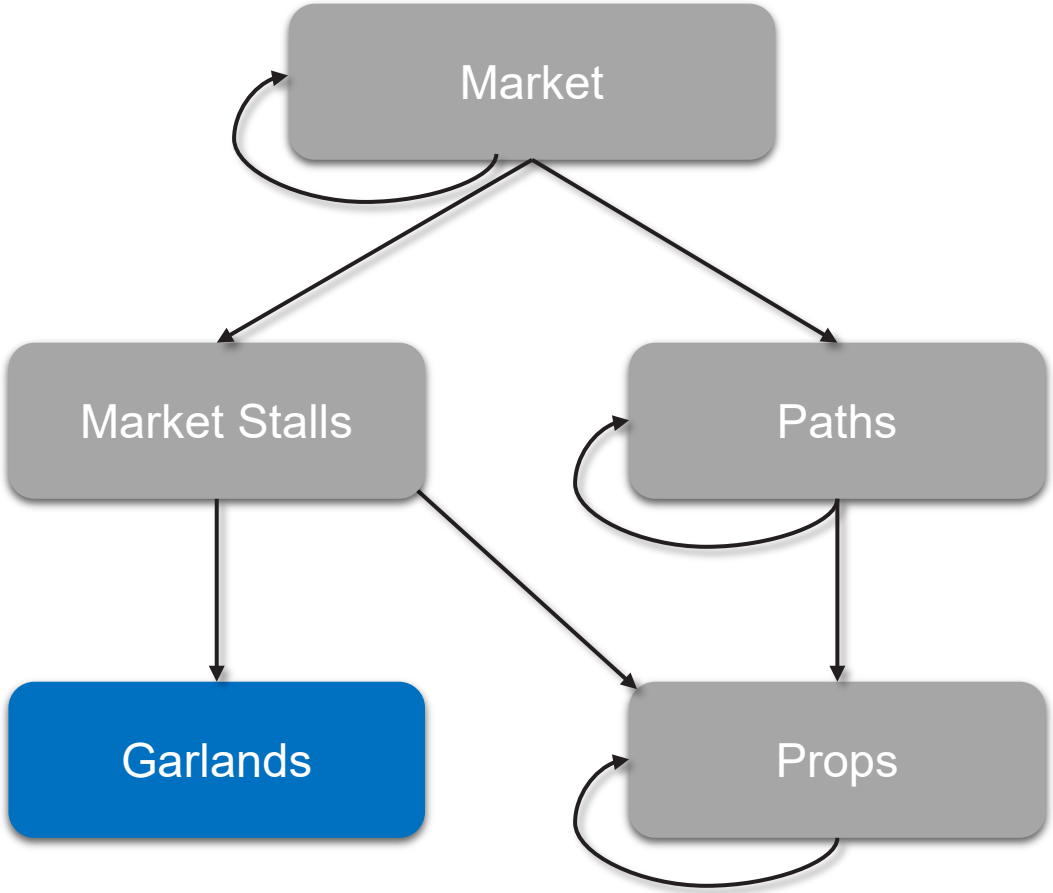


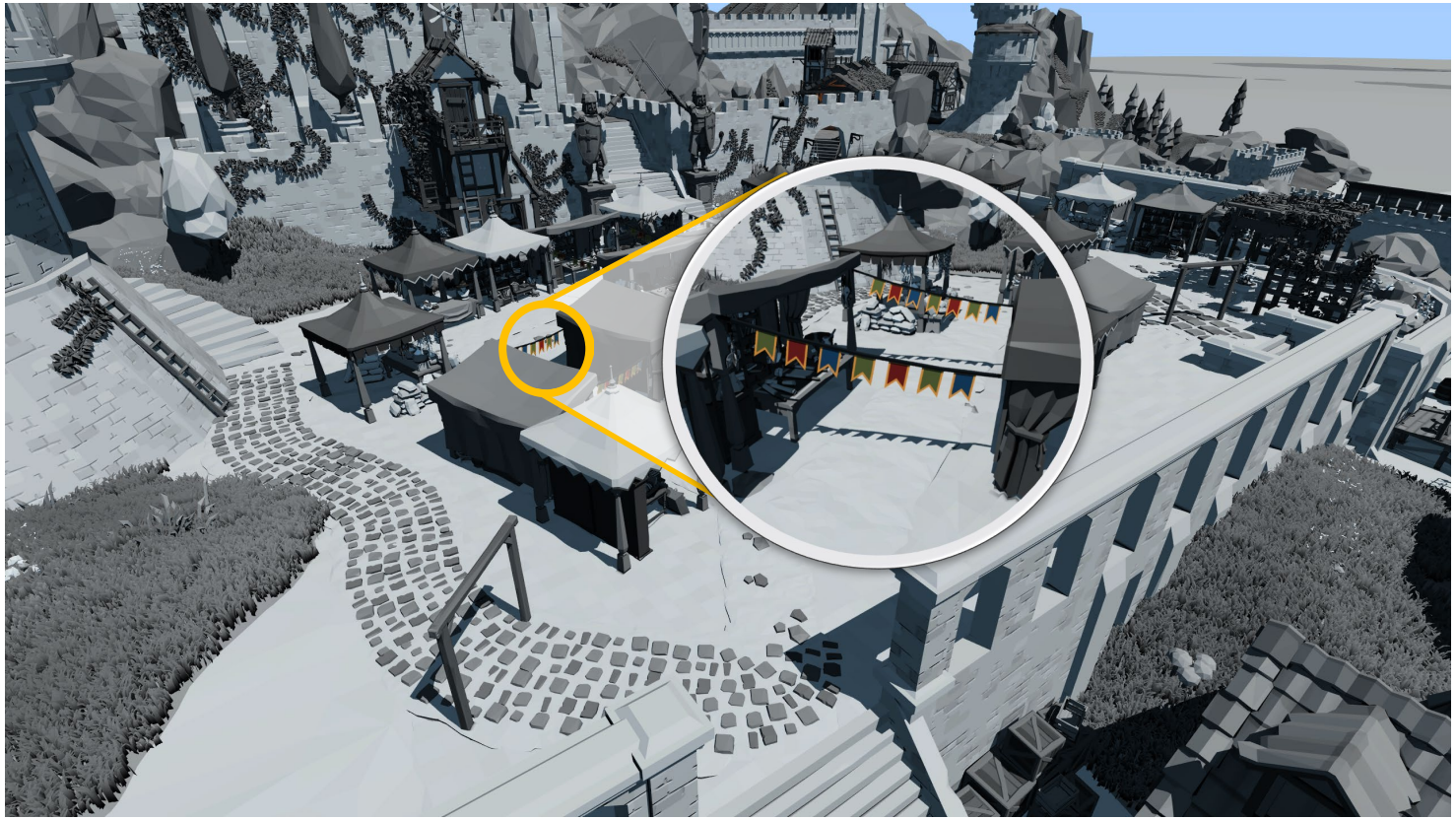


PROCEDURAL GENERATION EXAMPLE

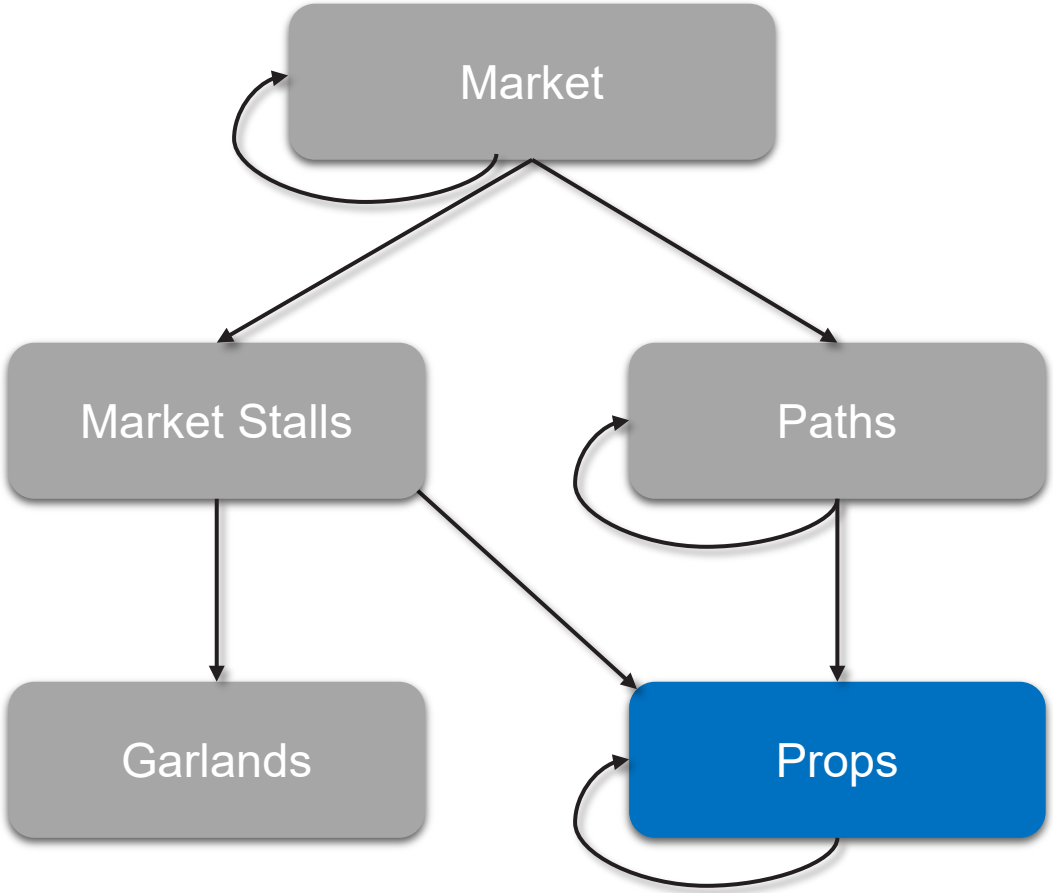


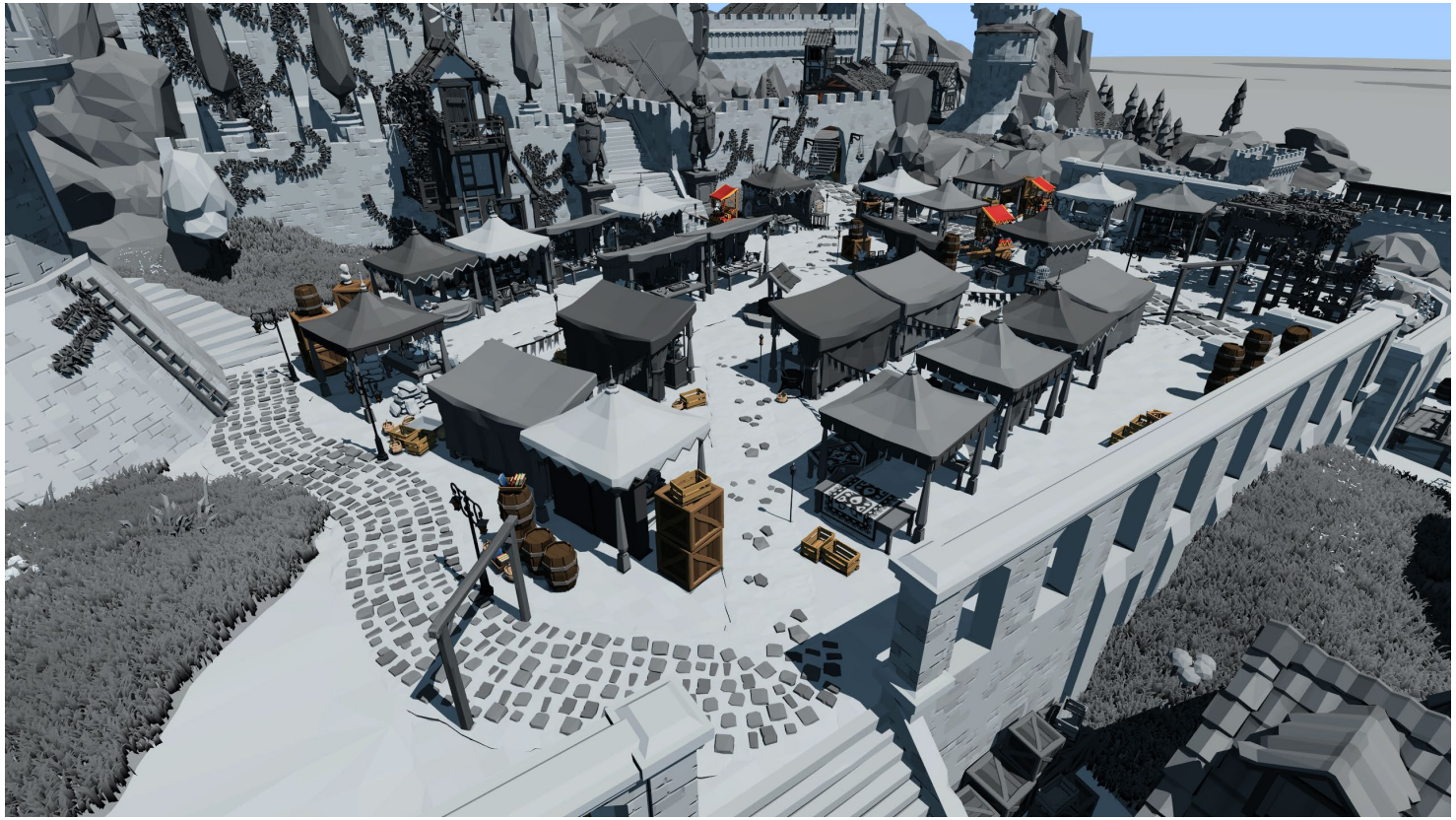
PROCEDURAL GENERATION EXAMPLE





PROCEDURAL GENERATION EXAMPLE





SPECIAL THANKS

- Lou Kramer
- Max Oberberger
- Matthäus G. Chajdas

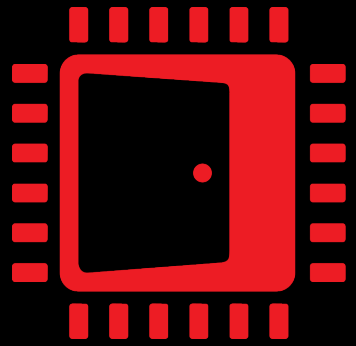
DISCLAIMER

The information contained herein is for informational purposes only and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale. GD-18

THIS INFORMATION IS PROVIDED 'AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Performance testing (slides 23, 30, and 31) done on following system: AMD Ryzen™ 9 7950X, 128GB DDR5-6000 memory, Asus ROG CROSSHAIR X670E HERO Motherboard, XFX RX 7900 XTX, 4TB M.2 NVME SSD, Windows® 11 Pro 22H2, AMD Software: Adrenalin Edition 23.40.14.01

Performance testing (slide 39) done on following system: AMD Ryzen™ 7 5800X, AMD Radeon™ RX 7900 XTX, AsusTeK TUF Gaming X570-Plus, 32GB DDR4-3600, Windows 10 Home 22H2, AMD Software: Adrenalin Edition 24.2.1



AMD
GPUOpen

AMD
together we advance_

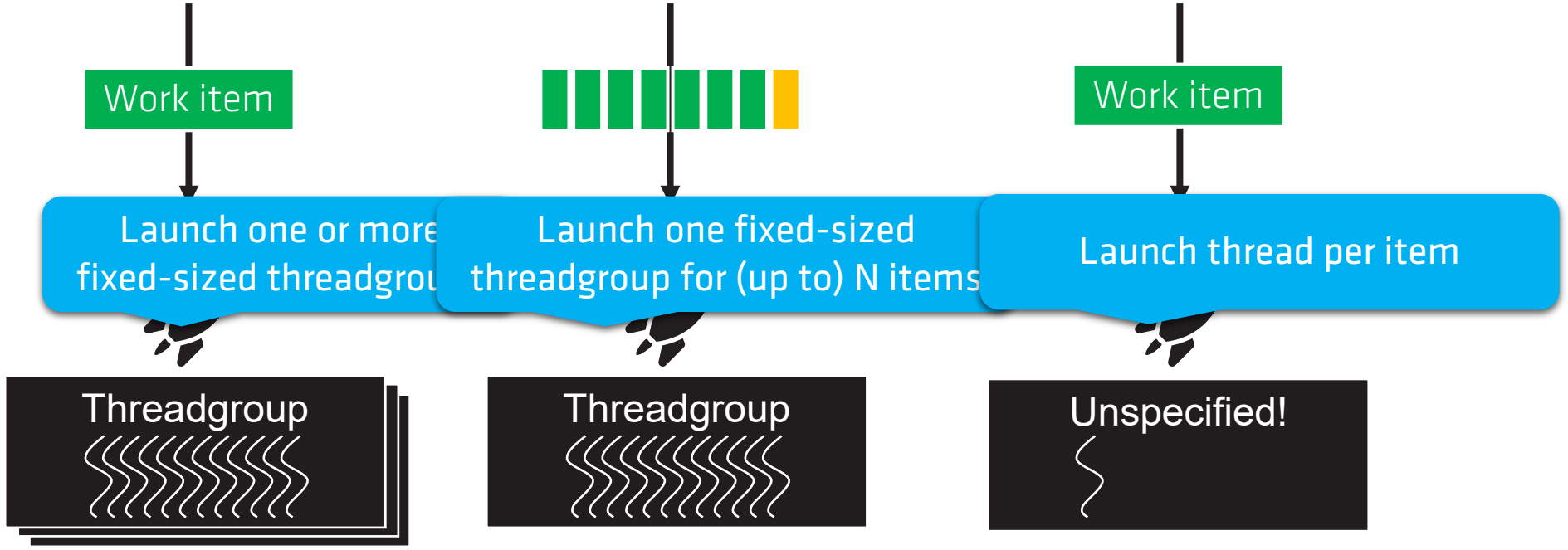
AMD
EPYC

AMD
RYZEN

AMD
RADEON

together we advance_

- You can select how things launch. Work items can ...
  - trigger a dispatch (“broadcast”)
  - be aggregated (“coalescing”)
  - be treated as independent launches (“thread”)



WORK GRAPHS – DRAW NODES

- Draw nodes: Feed into a mesh shader pipeline
- Work graph acts like an amplification shader on steroids
- Runtime ensures PSO switching isn’t too expensive
- Will buffer up draw calls per state
- Will optimize state changes
- The more similar the states are, the better – the cheapest state change is swapping out shaders only