Re-inventing the wheel for snow rendering

Paolo Surricchio

Senior Staff Rendering Programmer

Santa Monica Studio

Hello!

10+ years Engine/Rendering Engineer

Deadpool (360, PS3, PC)
Call of Duty: Advanced Warfare (360, PS3, PC, XOne, PS4)
Firewatch (PC, PS4)
God of War (PS4)
God of War: Ragnarök (PS4, PS5)

Santa Monica Studio

Goal of talk

How we write technology
to support high quality custom setups
by creating reusable toolset

Santa Monica Studio

Goal of talk

Look at re-write of height field system in God of War: Ragnarök
(no game spoilers)

Santa Monica Studio

Keep in mind

Less about tech details, more about why of each choice

Take-aways during the talk

Santa Monica Studio

How we do things at SMS

Art and design have lots of manual control

We work in Maya as our main level editor, run on kit

We use automation, but it's in service of creators control

Santa Monica Studio

Engineering challenge

Support creator team for the long term

Details are for our workflows/requirements, lessons are for all

Santa Monica Studio

Let's paint a scene

April 2018, God of War has shipped

Sequel is PS4/PS5, high quality on both

Santa Monica Studio

God of War: Ragnarök requirements

Good part of game is Midgard -> lots of snow

Fimbulvetr (Old Norse) ‘awful, mighty winter’

Santa Monica Studio

We already have that

Santa Monica Studio

Can't we re-use that? *Narrator voice: they couldn't*

Santa Monica Studio

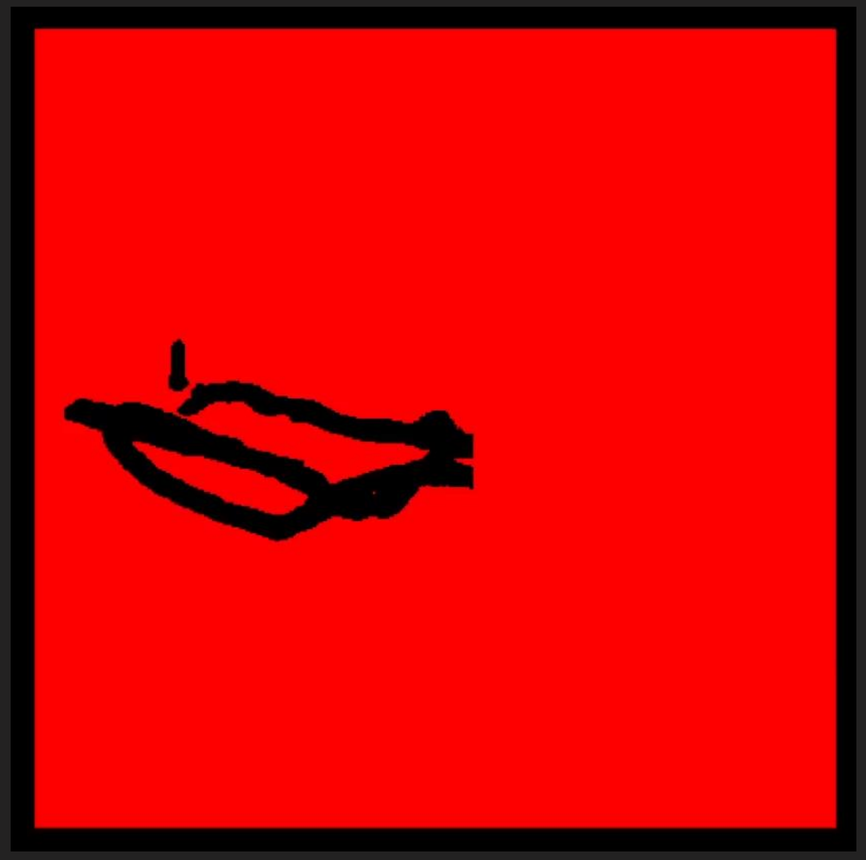
GoW 2018 Snow: rendering

Surface is rendered with ss-parallax mapping
All manual custom rendering different per material

Santa Monica Studio

GoW 2018 Snow: heights and details

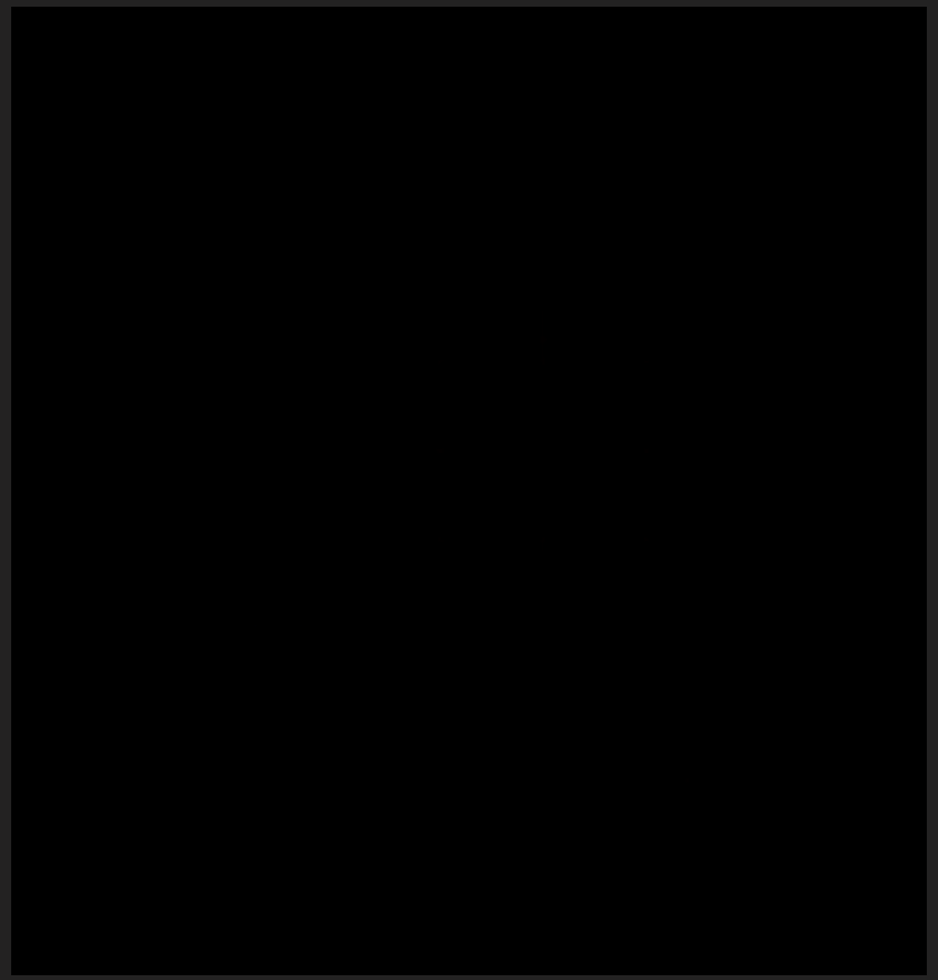
Carving shapes/VFX rendered into persistent top-down texture



Santa Monica Studio

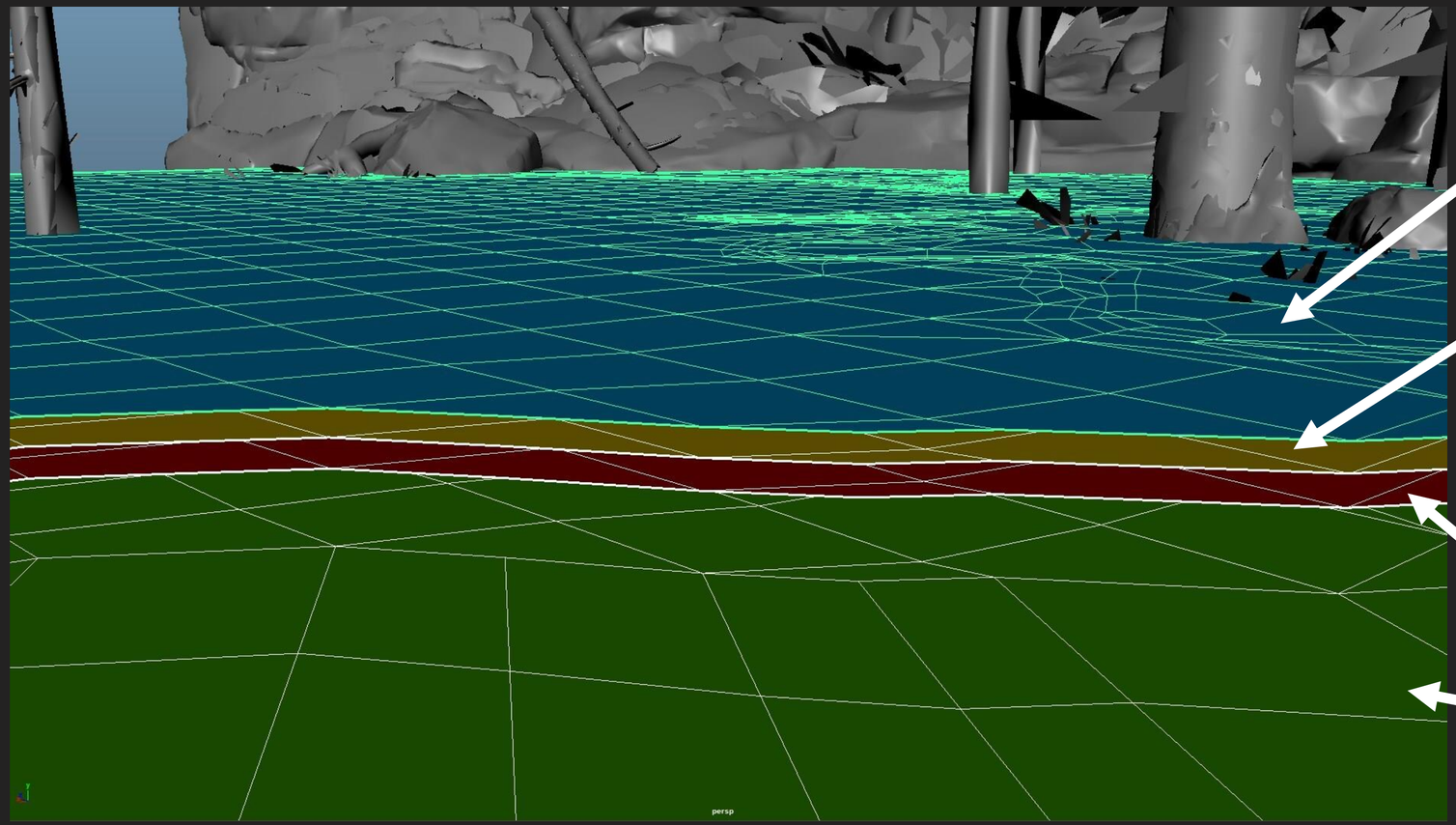
GoW 2018 Snow: heights and details

Project from top-down to screen, add VFX and details



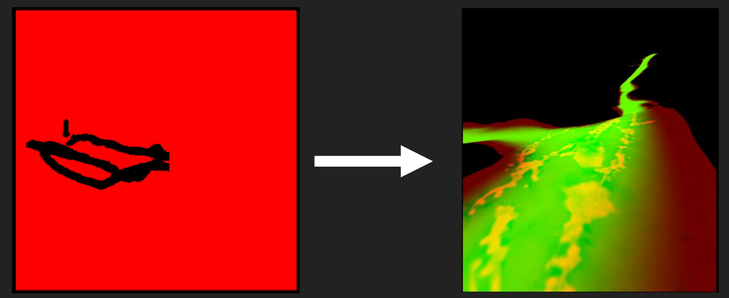
Santa Monica Studio

GoW 2018 Snow: Maya setup



Flat rendering surface

Projection surface



Max and min displacement

Santa Monica Studio

Pros...

Very detailed

Cost proportional to use

Santa Monica Studio

...and cons...

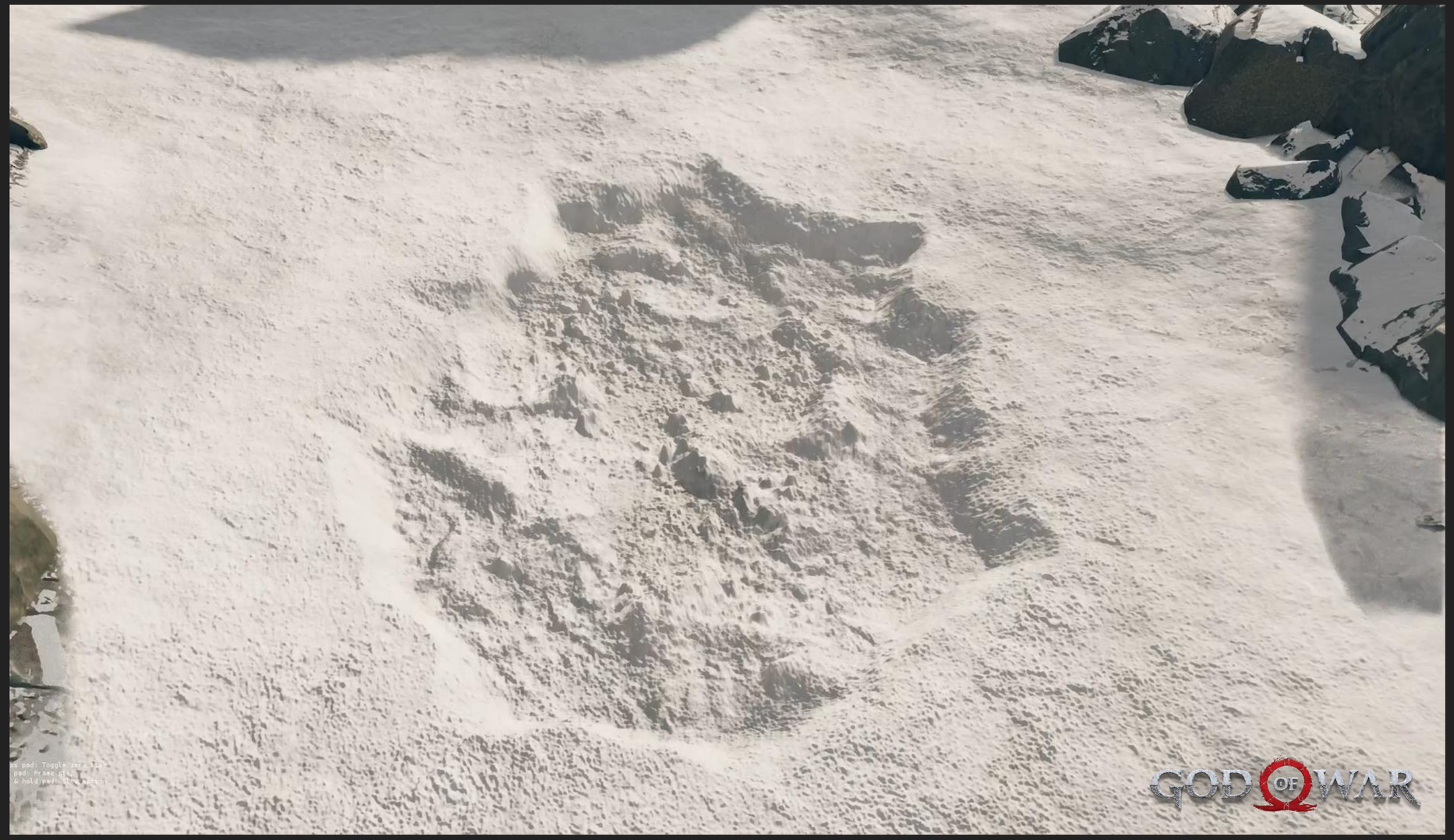
Variable cost, not good average/worst case

Tuning is hard, and error prone

Visual artifacts because of rendering technique

Santa Monica Studio

...and cons... (parallax and camera)



Santa Monica Studio

...and cons... (detail shimmering)

Santa Monica Studio

...and cons

Lots of limitations, lots of workarounds

Only flat terrains, camera always above

Art not making art (and not just them)

Santa Monica Studio

Consequences: Limited use

Santa Monica Studio

If all you have is a hammer...

Parallax itself is great

Great on water, still used it in Ragnarök

Not good for organic terrains

Good enough at the time given constraints

Santa Monica Studio

Summary of issues

Too much control, not right reasons

Too much because of parallax

Santa Monica Studio

A “Respectful Big Nope”

Art meeting with rendering:
current system won’t work

Santa Monica Studio

Why?

Bigger game

Bigger team

More teams

Santa Monica Studio

Game requirements

Snow is not an exception, it's the worst case

From 3 to ~50 areas 😐 (and more cases)

Santa Monica Studio

What didn't work

Change workflow: more automatic

Change core: geometrical approach

Santa Monica Studio

Our requirements

No terrain system; arbitrary meshes

Highly customizable with material features

Minimal setup

Aim for good average perf, no fixed memory cost

Must run well/look good on PS4, scale up on PS5

Santa Monica Studio

Compromise

Not same pixel quality

Art team gave the thumbs up

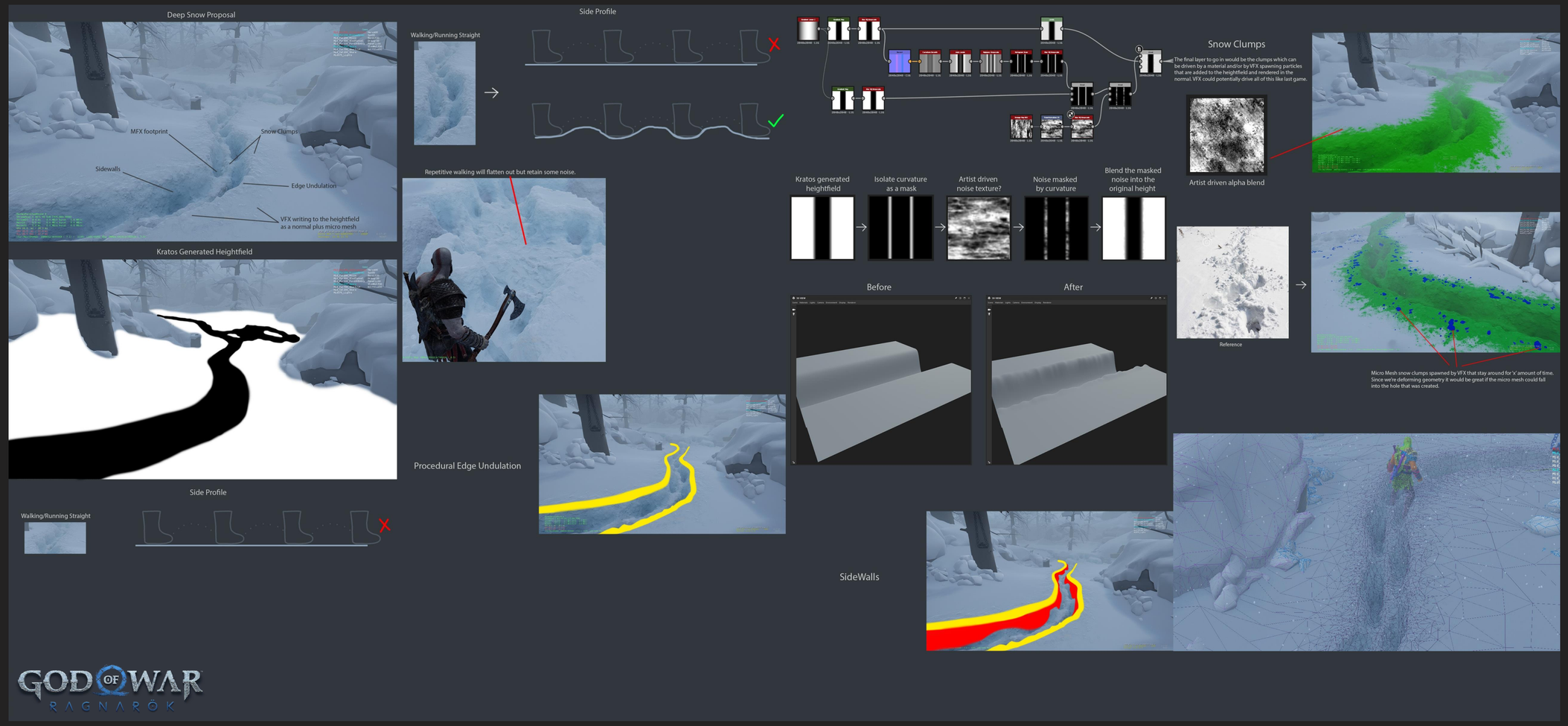
Santa Monica Studio

First step

Establish visual goal: get a spec from art

Santa Monica Studio

Oh, you want a spec?



Santa Monica Studio

GOD OF WAR RAGNARÖK

Second step

Back of the napkin math for perf

Same area as before test with tessellated mesh by hand (25 m^2)

Santa Monica Studio

Takeaway #1

Start from product, do back of napkin math

Make your knowledge “better”

Santa Monica Studio

Next steps

How do we go about dynamic geometry displacement?

Standing on shoulders of giants

Santa Monica Studio

What's out there

Terrain/fixed meshes/fixed tessellation

T-Junctions

High memory cost (or not flexible enough)

Santa Monica Studio

Do It Yourself

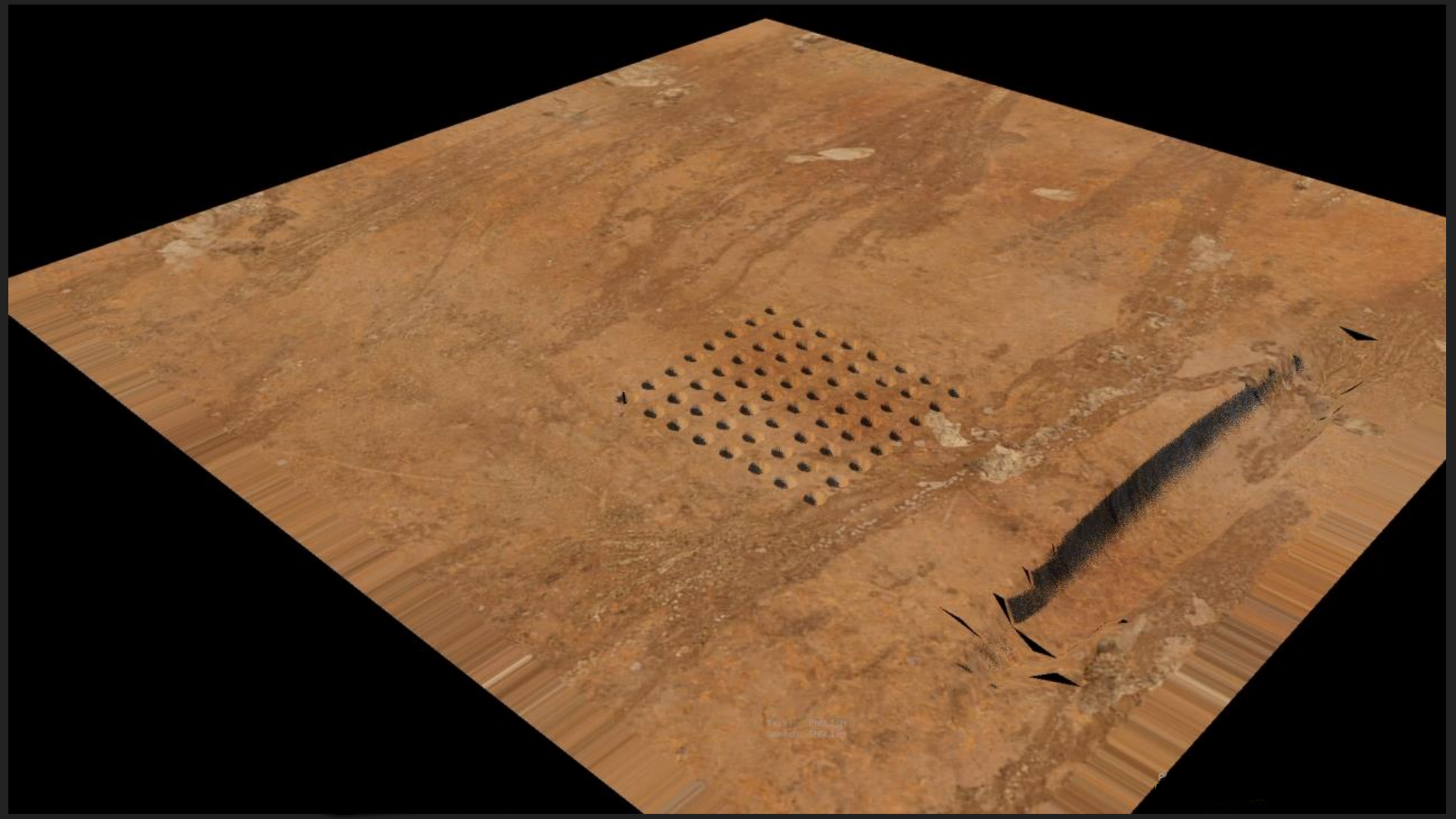
Some RnD but production/team constraints

Idea: indirect draws of differently tessellated tris and skin to original mesh – only where needed

Didn't pan out. Was it wasted? *(narrator voice: it wasn't)*

Santa Monica Studio

Do It Yourself: Programmer art edition



Santa Monica Studio

Sometimes you get lucky

Another programmer was working on hardware tessellation pipeline

Has issues, but fits all our boxes

Santa Monica Studio

Takeaway #2

Force yourself into a modular approach
(moral of the story)

"No Core Pillars"

Santa Monica Studio

How it works – 1st pass

Any mesh in Maya: just a checkbox (limitation on topology)

Reuse what worked before

Santa Monica Studio

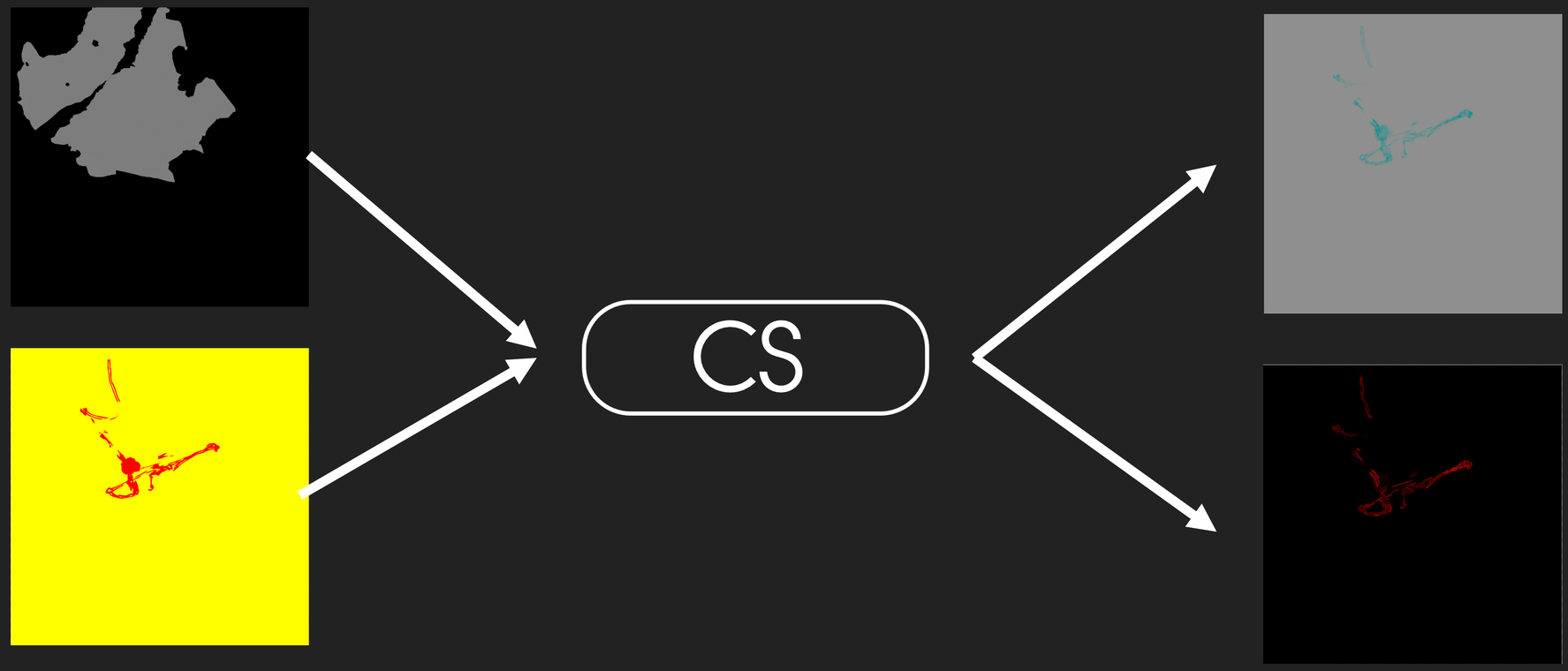
How it works – 1st pass

Automated heights and carving

Santa Monica Studio

How it works – 1st pass

Compositing pass:
smoothing heights & calculate tessellation factor



Santa Monica Studio

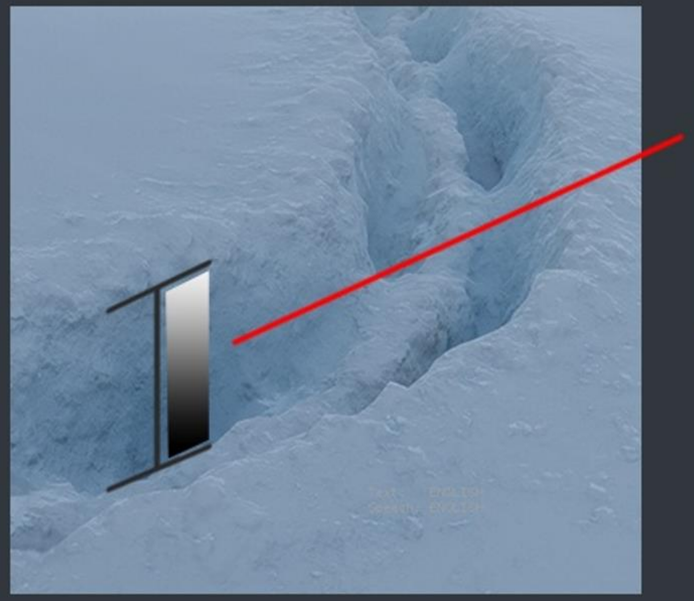
How it works – 1st pass

Hardware tessellation for mesh

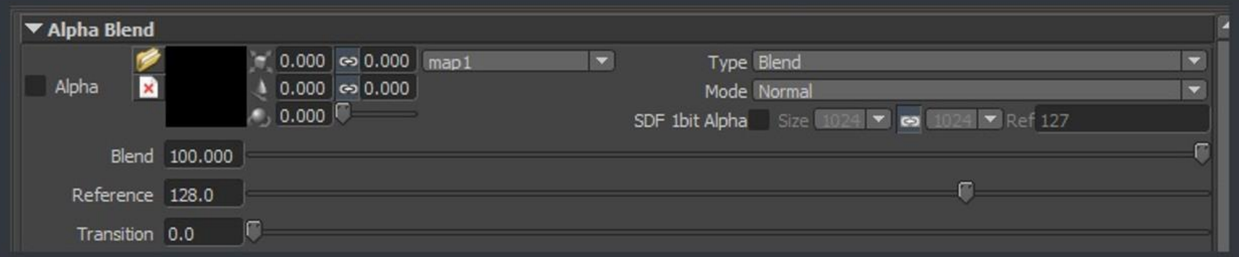
Santa Monica Studio

How it works – 2nd pass

Layers for blending



Using the depth of the geometry as an alpha will allow us to mask in the sidewall material and/or reveal materials underneath the snow



Santa Monica Studio

GOD OF WAR RAGNARÖK

How it works – 2nd pass

Santa Monica Studio

How it works – 2nd pass

Art starts requesting masking feature: red flag?

You CAN use it, but you don't HAVE to

Santa Monica Studio

How it works – 3rd pass

Detail Mask:

Persistent

Used as mask for detail normal map

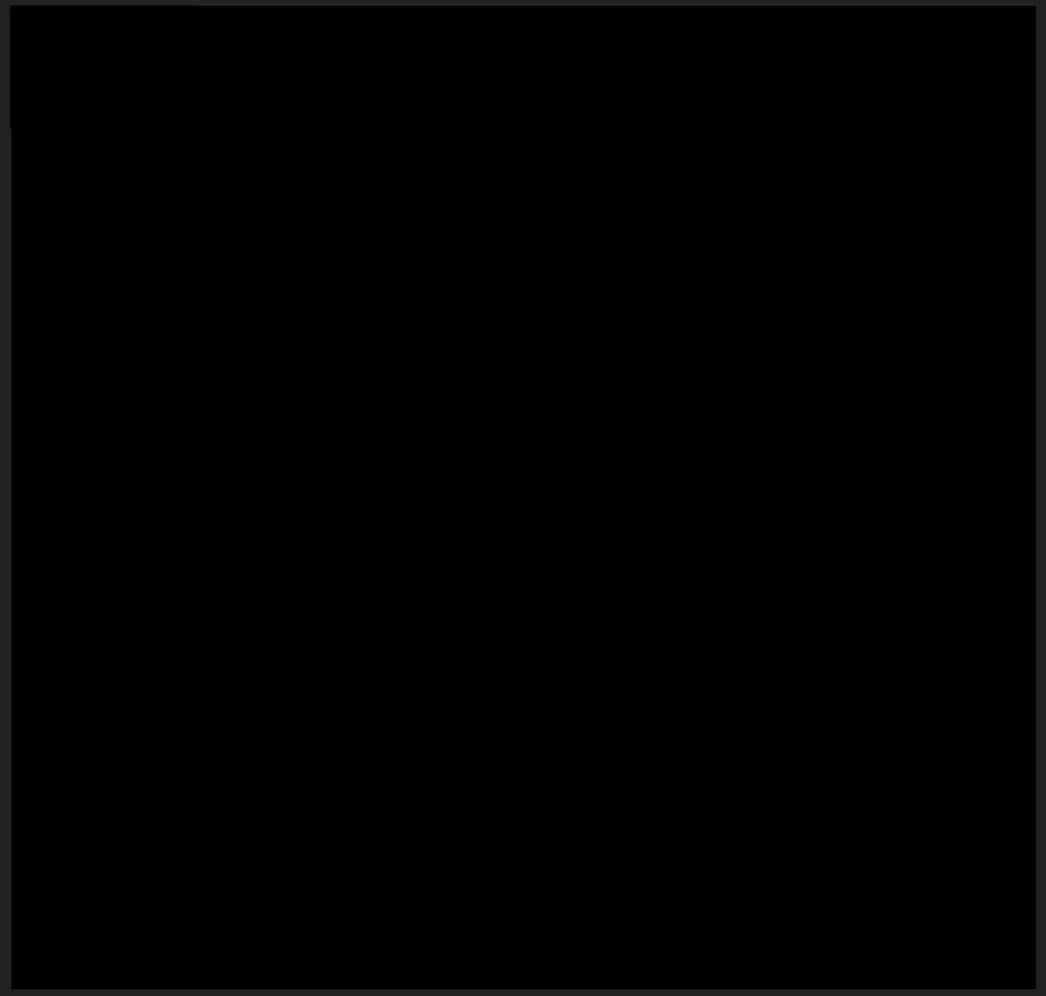
Santa Monica Studio

How it works – 3rd pass

Dynamic Details:

Non-Persistent

Generate dynamic normals in shader



Santa Monica Studio

How it works – 3rd pass

Addressed mesh topology limitation

Santa Monica Studio

What does it look like so far

Santa Monica Studio

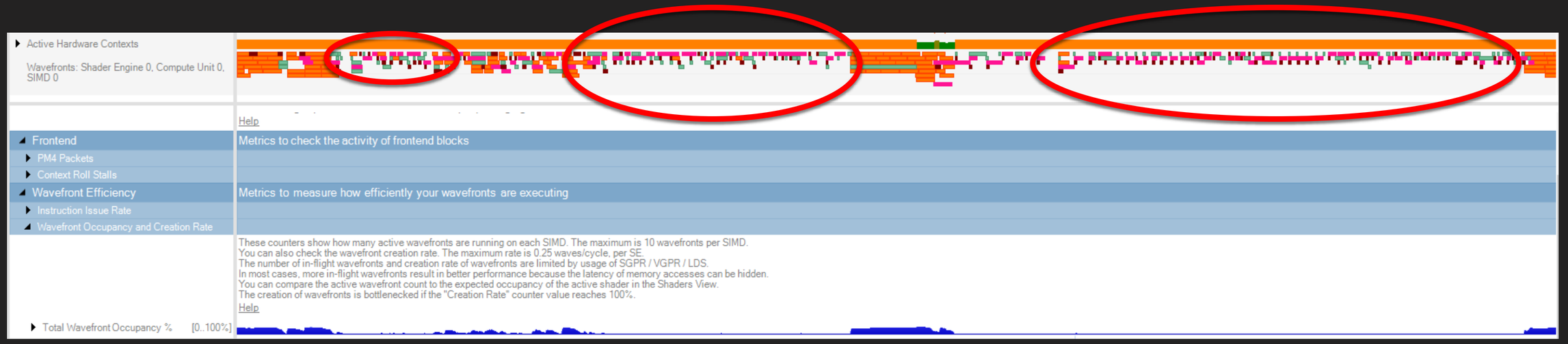
How does it run

As we start using it everywhere, cost creeps up

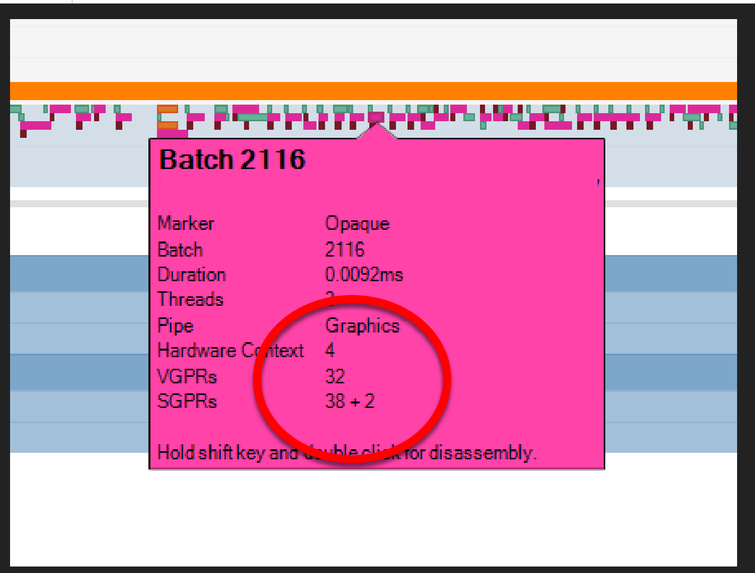
We expected it, but not as much

Santa Monica Studio

Performance: PS4 capture



Occupancy ☹



Santa Monica Studio

Takeaway #3

Be prepared

Santa Monica Studio

How did we address it?

Problem is tessellated tri count

Our use case is worst case scenario

Artist are nice people, tried to offer help

Santa Monica Studio

Solution overview - Summary

Inspired by GPU driven rendering

Partition the mesh and only render what's needed

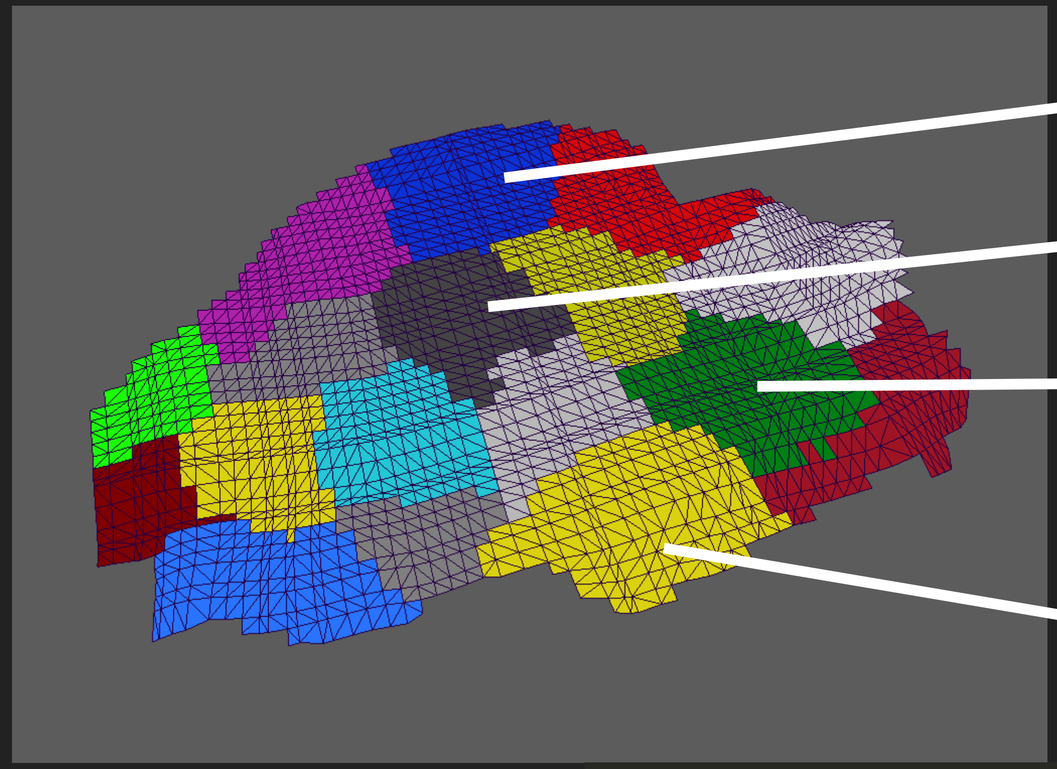
Saved a lot, thanks to research done before

Solved a specific problem, gained knowledge

Santa Monica Studio

How: Data

Divide mesh in chunks/meshlets
Create offset and culling info buffers

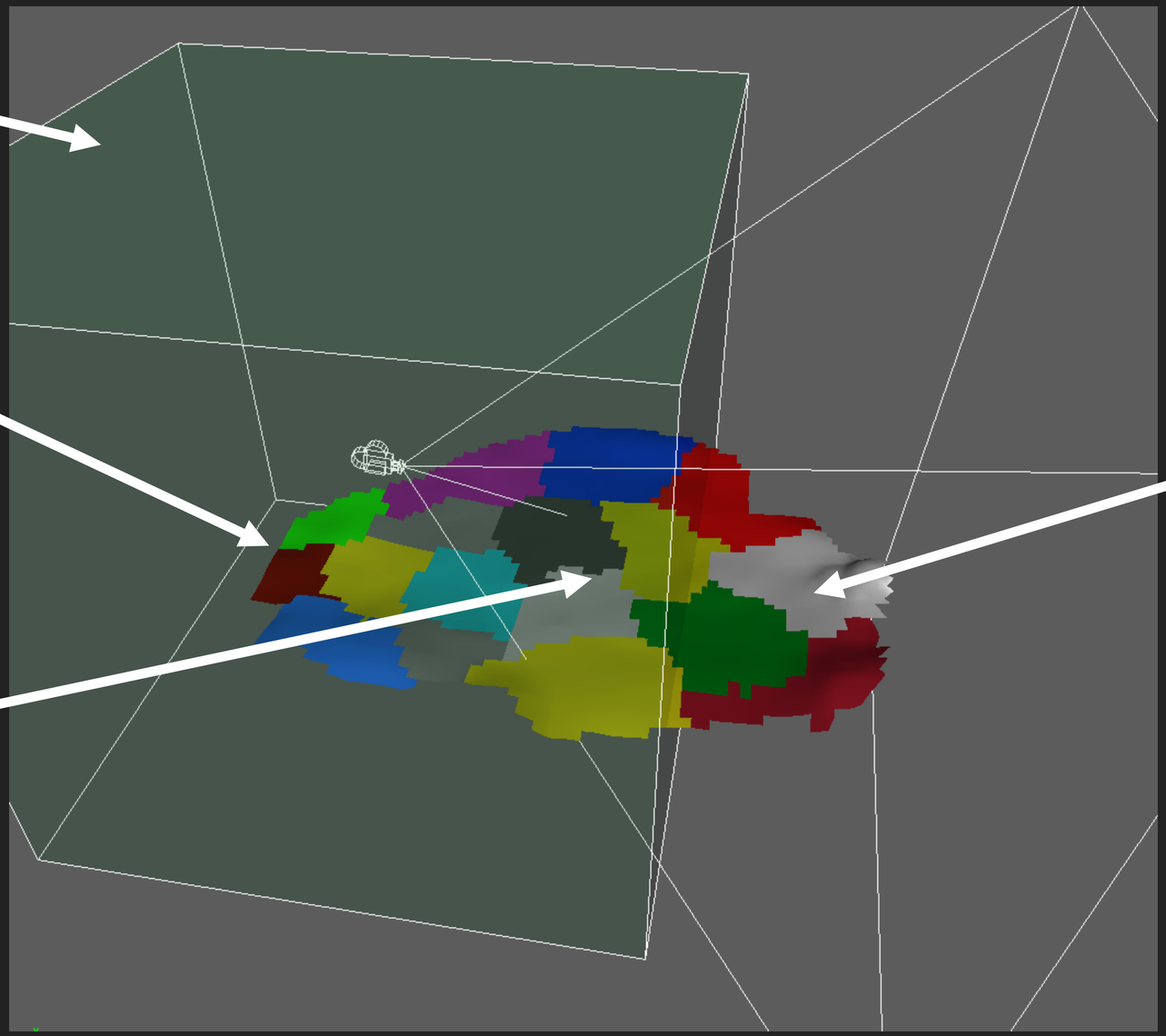


```
[
    {
        indexOffset = 0;
        indexCount = 315;
    },
    {
        indexOffset = 800;
        indexCount = 300;
    },
    {
        indexOffset = 1400;
        indexCount = 303;
    },

    ...

    {
        indexOffset = 15000;
        indexCount = 297;
    }
]
```

```
maxMeshletIndexCount = 315
```

Santa Monica Studio

How: Culling and Draw Args



Santa Monica Studio

How: Draw Call

CS

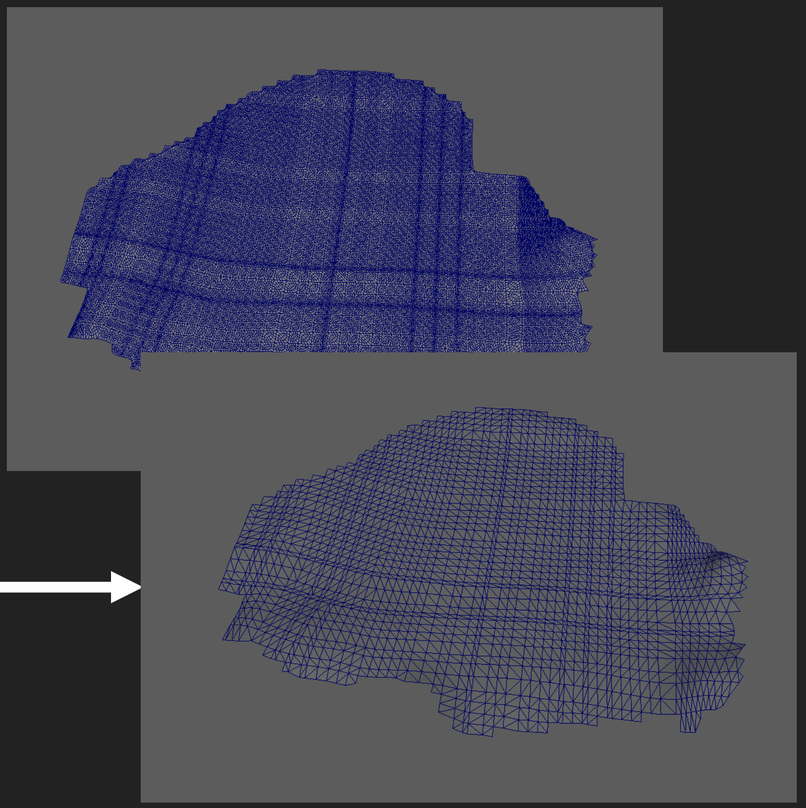
```
maxMeshletIndexCount * validMeshletCount
```

```
struct DrawIndirectArgs
{
    uint m_vertexCountPerInstance;
    uint m_instanceCount;
    uint m_startVertexLocation;
    uint m_startInstanceLocation;
};
```

HW Tess

```
struct DrawIndirectArgs
{
    uint m_vertexCountPerInstance;
    uint m_instanceCount;
    uint m_startVertexLocation;
    uint m_startInstanceLocation;
};
```

VS



Santa Monica Studio

How: Mesh VS

```
// Returns true if the given Input Assembler Index ID in a meshlet draw corresponds to a valid
// meshlet index, and in that case replaces it with the meshlet Index ID that can be used
// to read from the index buffer.
bool FetchMeshletIndexID(..., const bool isTessellatedDraw, inout uint indexID)
{
    uint maxMeshletIndexCount = constantBuffer.heightFieldMaxMeshletIndexCount;

    const uint indexIDInMeshlet = indexID % maxMeshletIndexCount;
    const uint meshletIDInDraw  = indexID / maxMeshletIndexCount;

    // Get meshlet ID, and base index offset
    uint meshletID, baseIndexOffset;
    if (isTessellatedDraw)
    {
        meshletID = meshletIDInDraw;
        baseIndexOffset = constantBuffer.tessellatedIndexOffset;
    }
    else
    {
        meshletID = constantBuffer.heightFieldMeshletCount - 1 - meshletIDInDraw;
        baseIndexOffset = 0;
    }

    const uint meshletIndex = constantBuffer.heightFieldMeshletIndices[meshletID];
    const uint2 fetchedMeshletInfo = constantBuffer.heightFieldMeshletInfo[meshletIndex];

    HeightFieldMeshletInfo meshletInfo;
    meshletInfo.indexOffset = fetchedMeshletInfo.x;
    meshletInfo.indexCount = fetchedMeshletInfo.y;

    // We use the maximum meshlet index count as index stride between meshlets
    // while the vertex buffers are compacted so the surplus indices must be discarded
    if (indexIDInMeshlet >= meshletInfo.indexCount)
        return false;

    indexID = indexIDInMeshlet + meshletInfo.indexOffset + baseIndexOffset;

    return true;
}
```

Draw index count is
meshletMax * validChunksCount

Not “real” index

Clip invalid vertices

```
// We use the maximum meshlet index count as index stride between meshlets
// while the vertex buffers are compacted so the surplus indices must be discarded
if (indexIDInMeshlet >= meshletInfo.indexCount)
    return false;

indexID = indexIDInMeshlet + meshletInfo.indexOffset + baseIndexOffset;

return true;
```

Santa Monica Studio

How: result

Santa Monica Studio

Art very happy

No need of user optimization -> better game

Santa Monica Studio

Be ready for more

New lead, more people
more wiggle room in schedule

Core is shippable

Go back to visual goal: add more details

Santa Monica Studio

More polish

Opaque particles/geo collision

Santa Monica Studio

More polish

Added dynamic model spawning around displacement

Santa Monica Studio

Art even happier

Very simple parameter driven system around height displacement

Art gets taste of in-game no-authoring pipeline

Santa Monica Studio

Greater than the sum of its parts

All thanks to reusable components

You have to ship it!

Height field tech, not just snow

Santa Monica Studio

Concept vs In-Game

Santa Monica Studio

We didn’t forget about Ps5

Santa Monica Studio

In Closing - Requirements

No terrain system; arbitrary meshes ✓

Highly customizable with material features ✓

Minimal setup ✓

Aim for good average perf, no fixed memory cost ✓

Must run well/look good on PS4, scale up on PS5 ✓

Santa Monica Studio

In Closing – The Process

We shipped on features on time

We were able to expand and reuse

No major re-writes/issues

Santa Monica Studio

In Closing – The Future

We have tech that solved a problem,
but can/will change easily

Code can change, but we won't re-invent the wheel since the wheel ~~is the friends we made~~ is the knowledge we gained along the way

Santa Monica Studio

In Closing – Takeaways

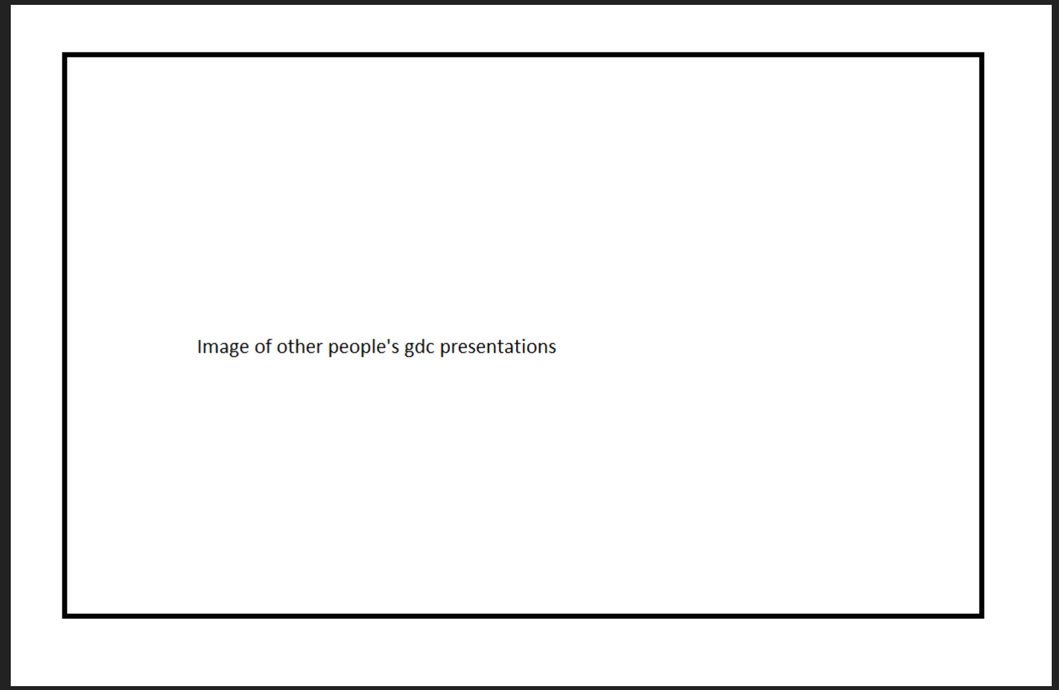
Start from product, back of napkin math

No “Core Pillar”

Be prepared

Santa Monica Studio

Thank you!



Santa Monica Studio

Thank you!

paolo.surricchio@sony.com

twitter: @rollingshark
mastodon: @therollingshark@mastodon.social

We are hiring: https://sms.playstation.com/careers

Santa Monica Studio