

SIMPLYGON™

Automagic 3D Optimization



Speed up your game using Simplygon



David Larsson
david@simplygon.com

Special Guest:
Balázs Török, CD Projekt RED

Some of our clients...



Overview

- Why LODs?
- Simplygon Feature Overview
- Case study: Simplygon in Witcher 3
- Simplygon 6

Why LODs

- Frame rate is critical for a good gaming experience
- Allow more complex environments
- Improves experience in almost any 3D based game

LODs in games

- Distance based LODs
- Performance based LODs
 - Hardware spec
 - Local spots

How LOD Techniques Help Performance

- Geometry stage
- Overdraw and Overshading
- Draw calls
- Shading complexity
- CPU – Bone Animation

LODs in a traditional pipeline

- Manually or semi-automatically created LODs
- Simple distance based LOD selection

Issues

- Tedious manual work
- Costly, man-years for AAA/MMO game
- $4\text{h/asset} * 10000 \text{ assets} = 20 \text{ man-years}$
- Longer iteration times, even longer with outsourced work
 - Hard to know how your game plays during development without LODs

Issues

- Tedious manual work
- Costly, man-years for AAA
- $4\text{h/asset} * 10000 \text{ assets}$
- Longer iteration times with outsourced work
 - Hard to know how you're doing development without LOD's

Simplygon

- Automates LOD and Proxy generation
- Automates porting of content to lower end devices

Simplygon

Maya, Max, Softimage, UE

GUI

API

Optimizer

Simplygon

Maya, Max, Softimage, UE

GUI

API

MeshLOD

ProxyLOD

MaterialLOD

BoneLOD

Simplygon

Maya, Max, Softimage, UE

GUI

API

MeshLOD

ProxyLOD

MaterialLOD

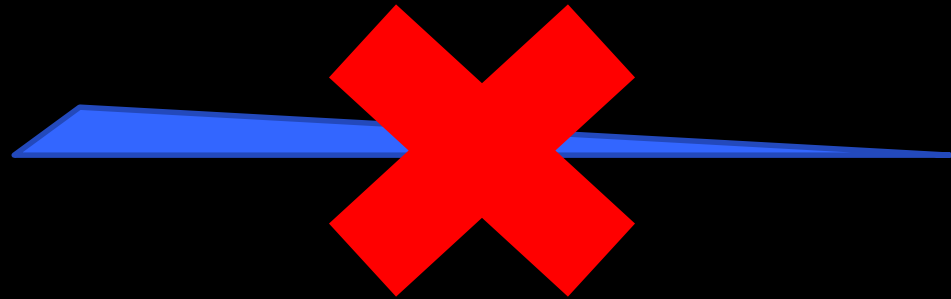
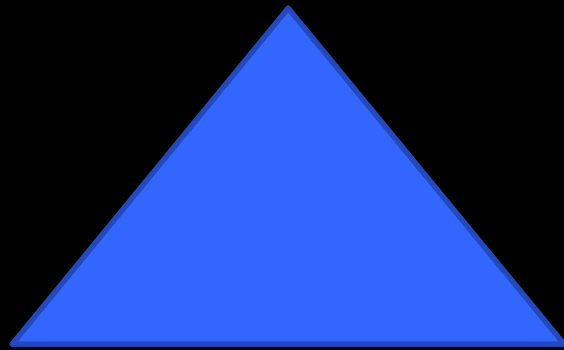
BoneLOD

Geometry Stage

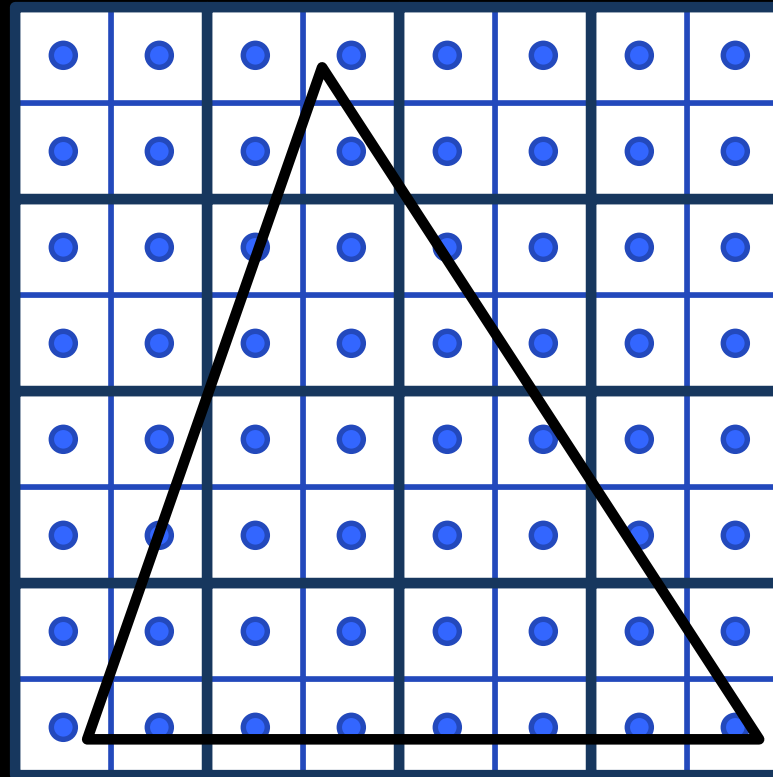
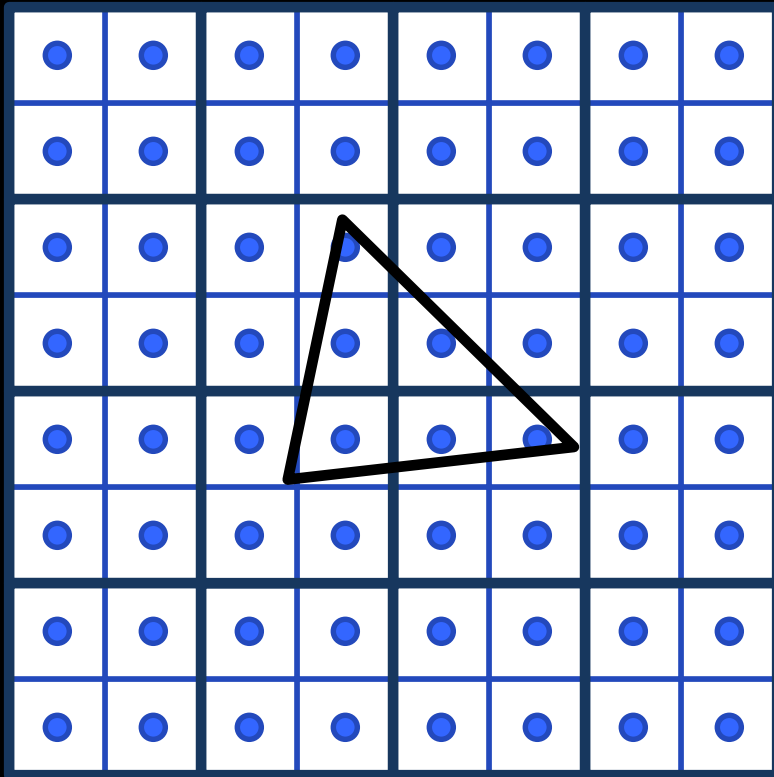
- LODs with fewer triangles, vertices
- Vertex shader, T & L, projection, (clipping)

Overshading – Quad utilization

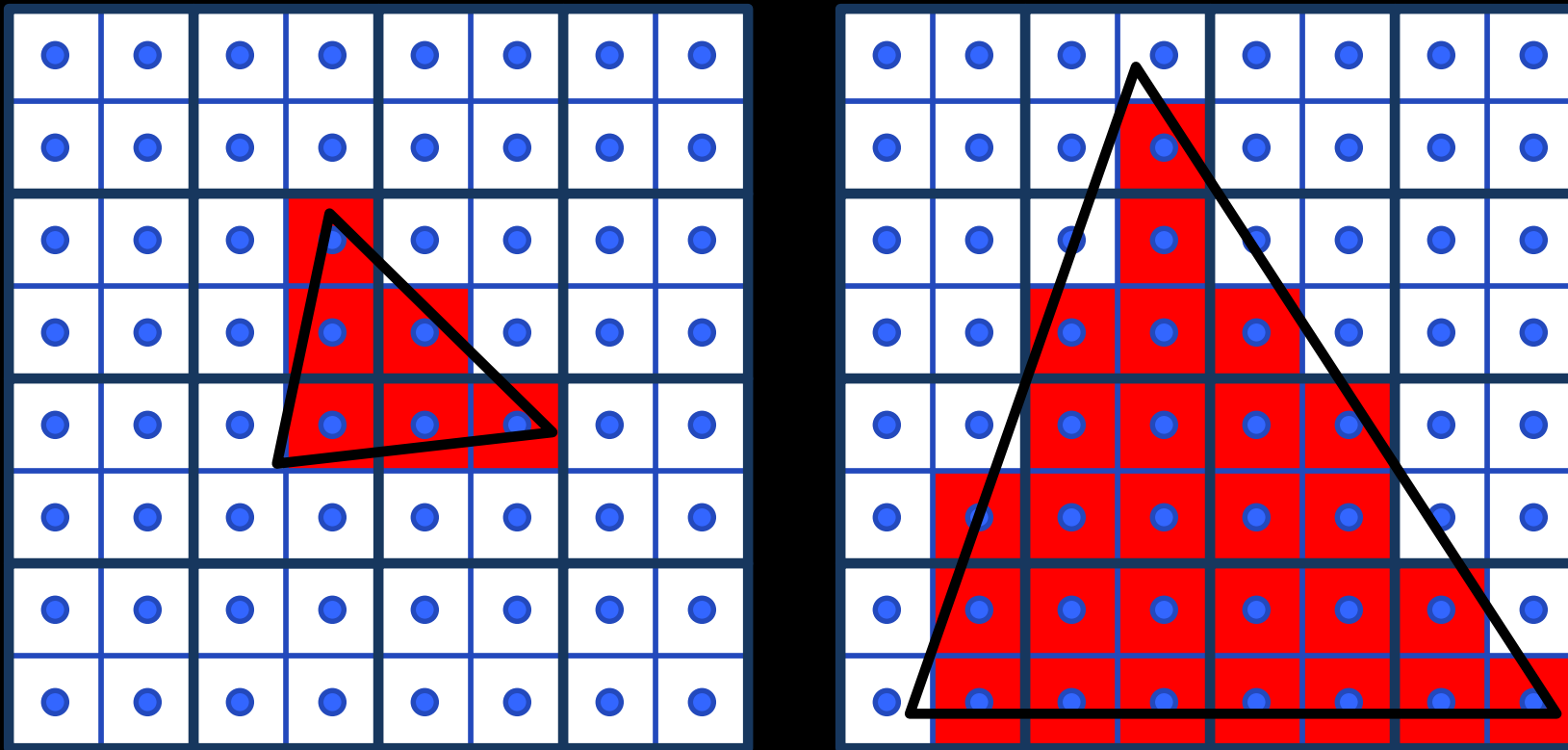
- LODs with larger (relative) and better shaped triangles
 - Quad utilization, should be at least ~16 pixels on most HW
- Wellformed triangles preferred
 - Sliver triangles have long edges, small area



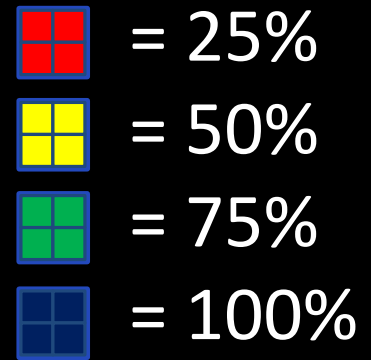
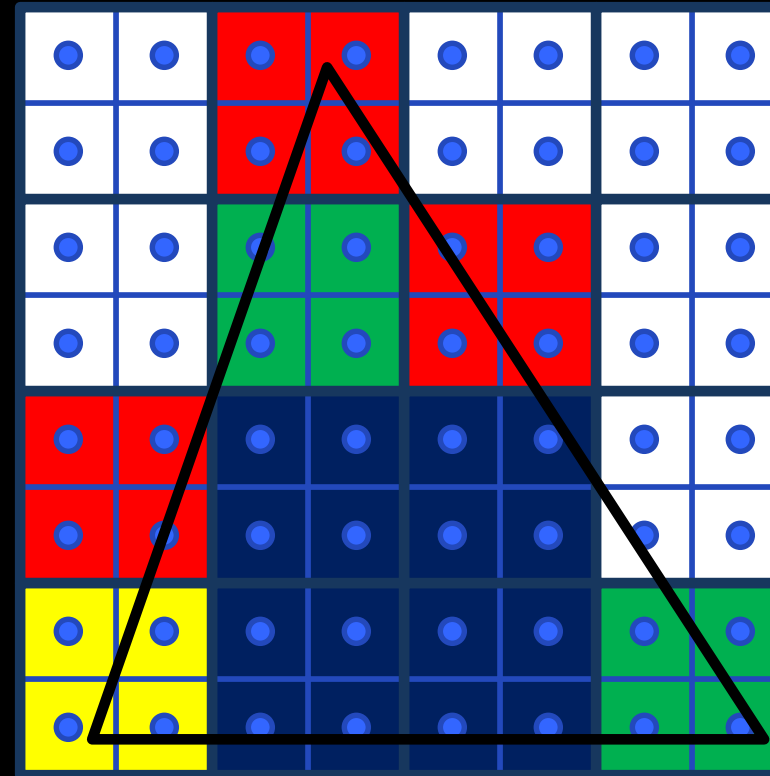
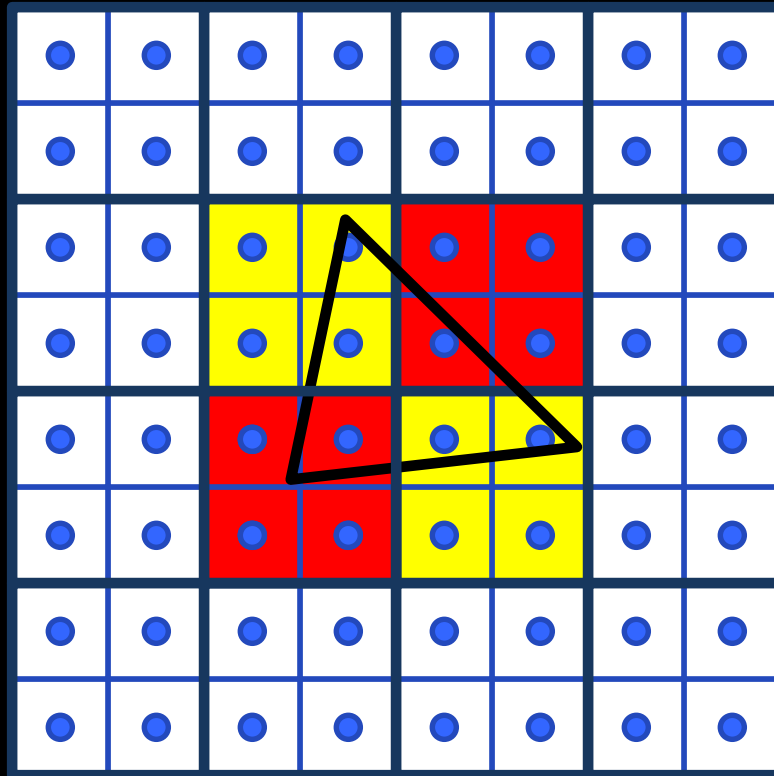
Overshading– Quad utilization



Overshading– Quad utilization

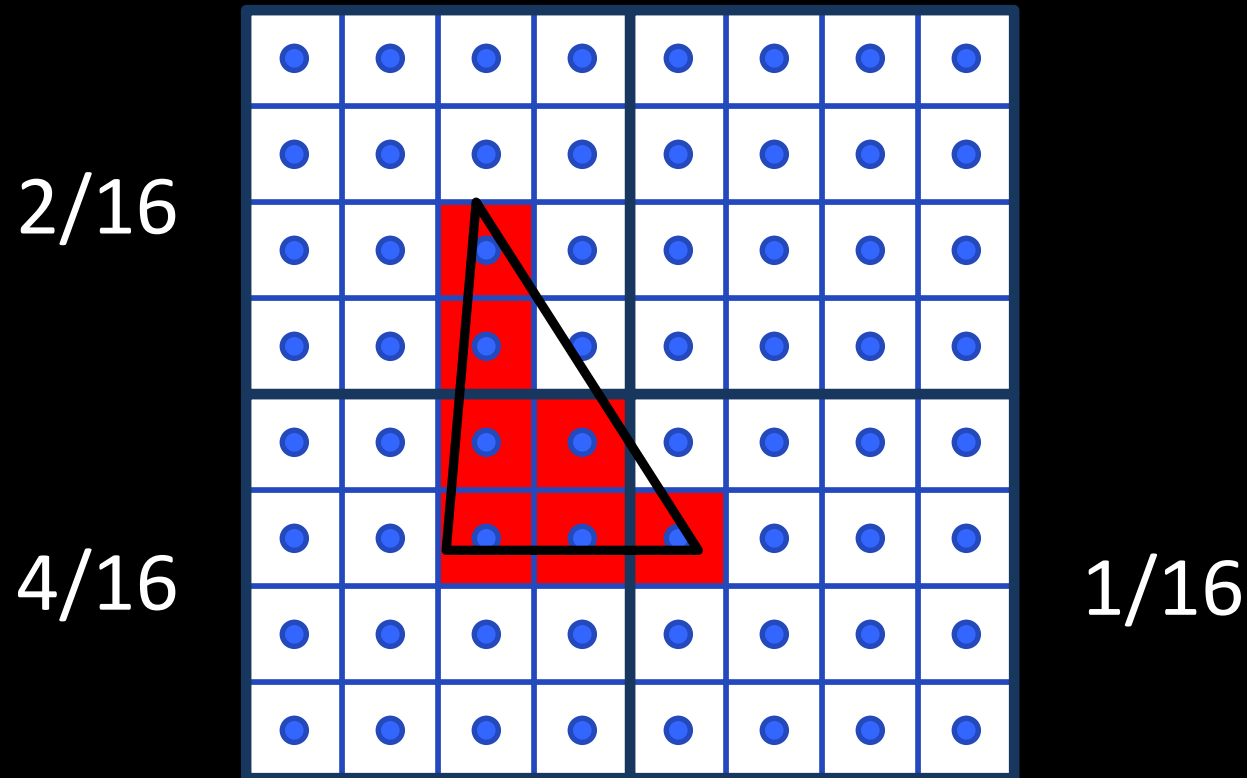


Overshading– Quad utilization

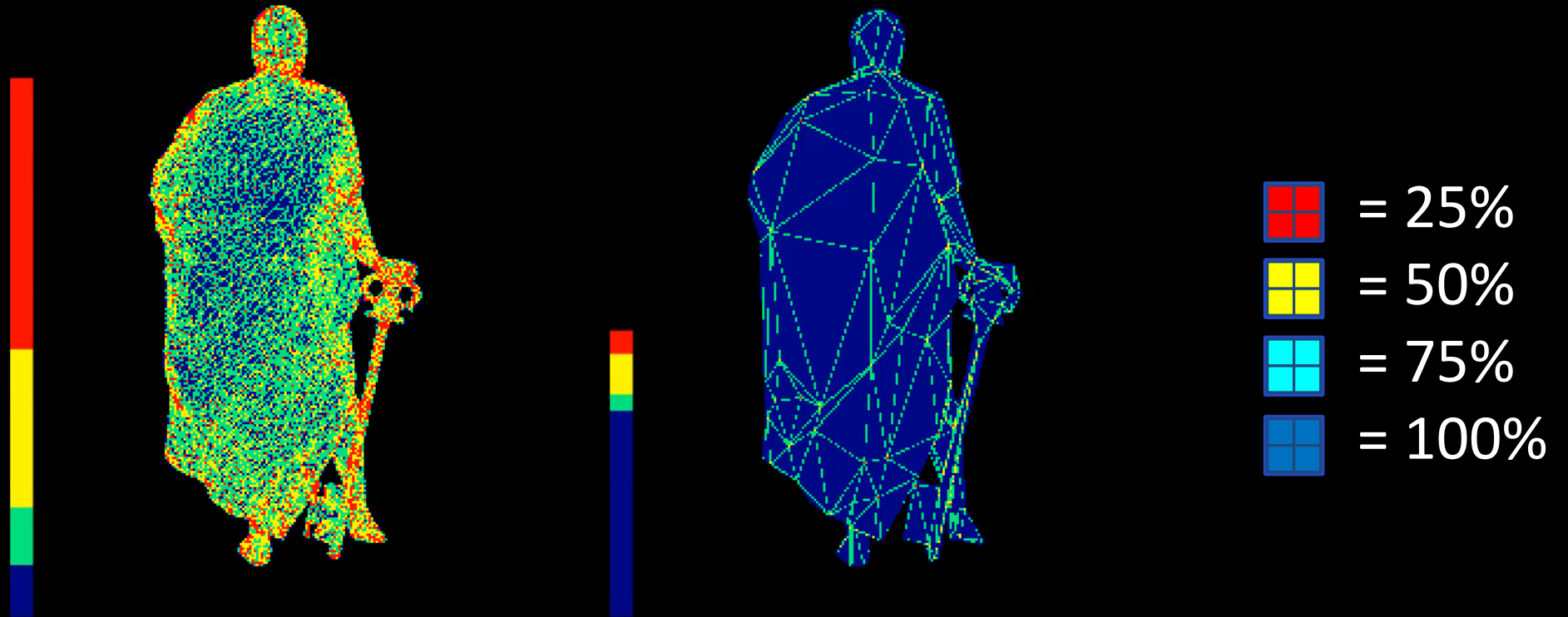


Overshading – Larger quads

- Most GPUs have even wider SIMD units, like 8, 16 or 32 pixels



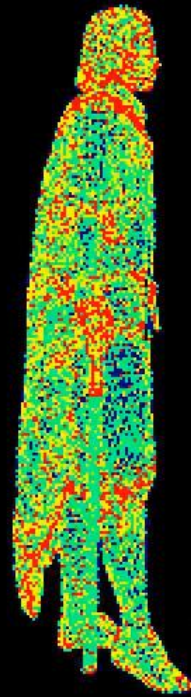
Overshading



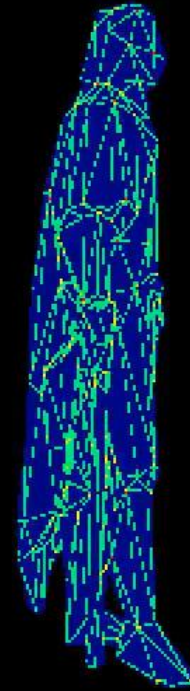
Original

Optimized

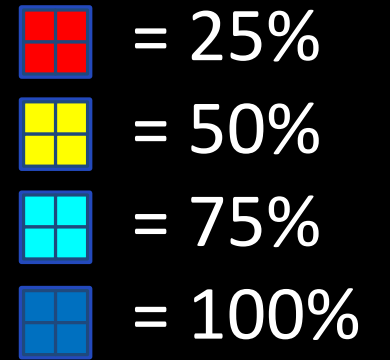
Overshading



Original



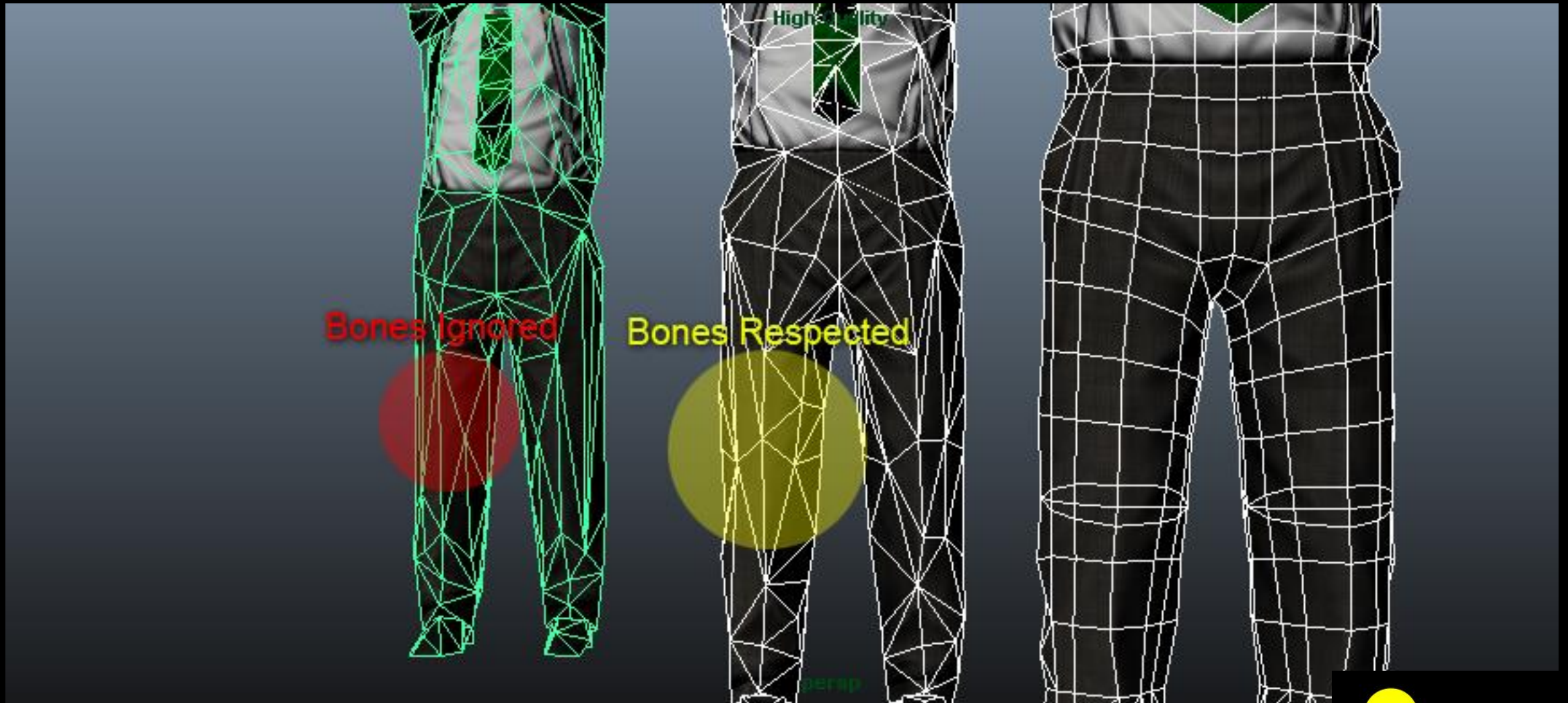
LOD Chain



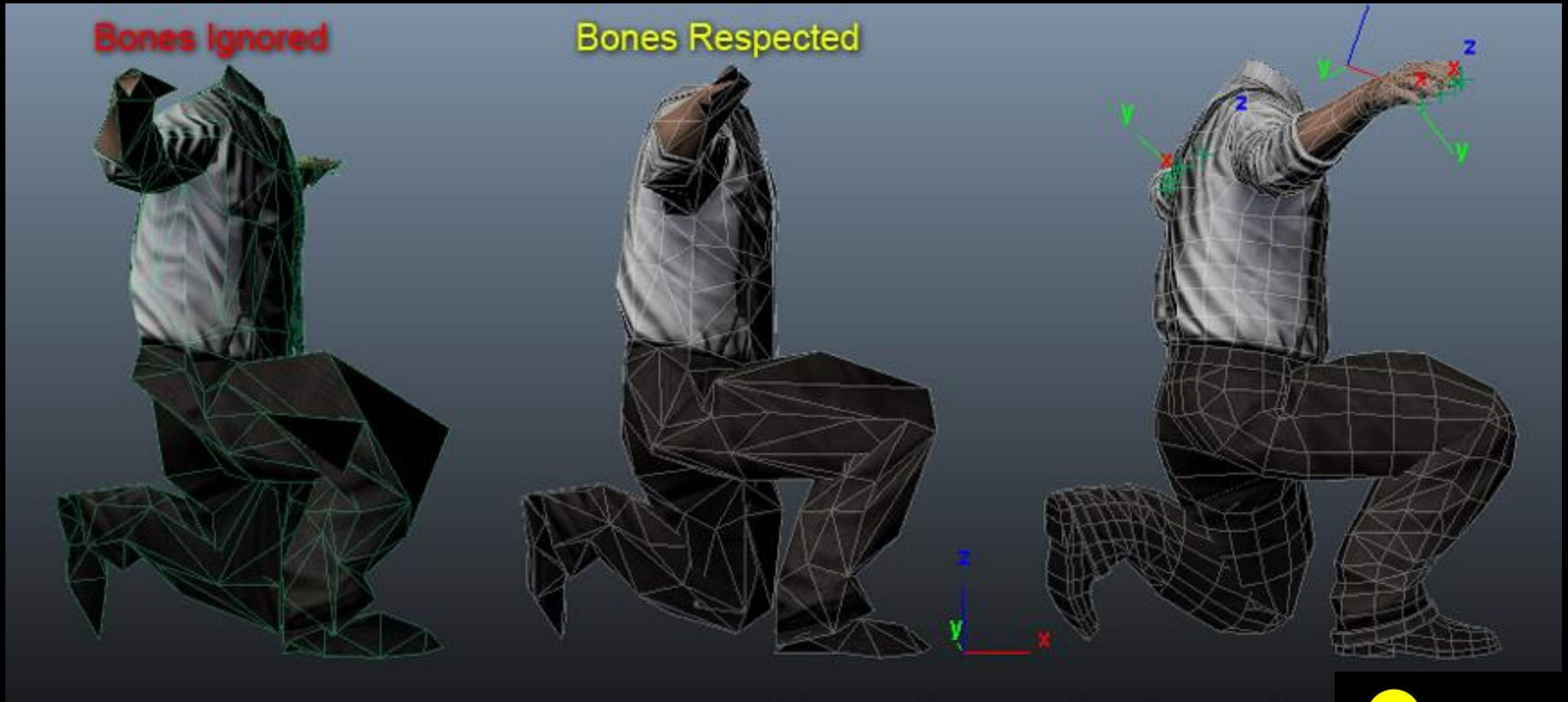
Simplygon MeshLOD

- State of the art triangle reduction
- Made for game assets
 - 256 UVs, 256 colors, 16 bone skinning etc
 - Full LOD chain directly
- On-screen size / triangle percent
 - Make a LOD for a specific size, or
 - Tell where the LOD would switch well
- Supports non-manifolds

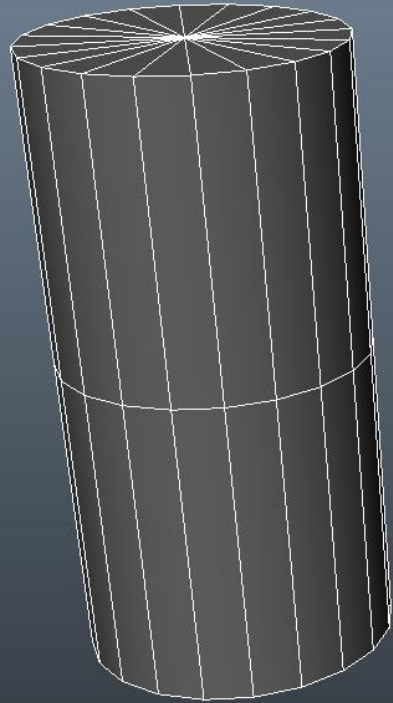
Full skinning support – up to 16 bones/vtx



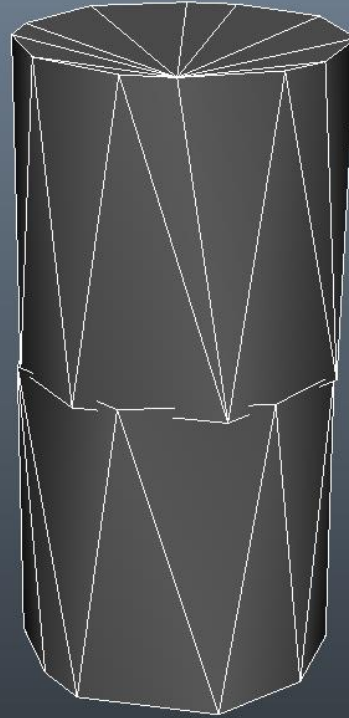
Full skinning support – up to 16 bones/vtx



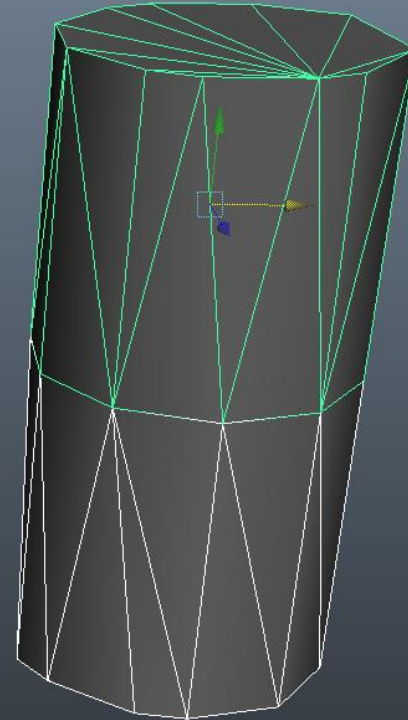
Object Boundaries



Original



Reduced in
Isolation



Reduced
together

Mesh LOD Demo



Simplygon

Maya, Max, Softimage, UE

GUI

API

MeshLOD

ProxyLOD

MaterialLOD

BoneLOD

Object, topology & material reduction

- Overdraw
- Draw calls
- Material complexity

Draw Calls

- We want lot of objects
- We want to reuse components
- Results: Lots of draw calls
- Lots of overdraw



Draw Calls

- Often a bottleneck
- Extra costly on:
 - PC (API, driver stack, costly kernel calls)
 - Mobile (Low perf. CPU)
- Partial or full flushes on other hardware

Proxy and Material LOD's to the Rescue

- LODs with Fewer objects, materials
 - Hierarchical LODs (HLODs)
- Simplify materials

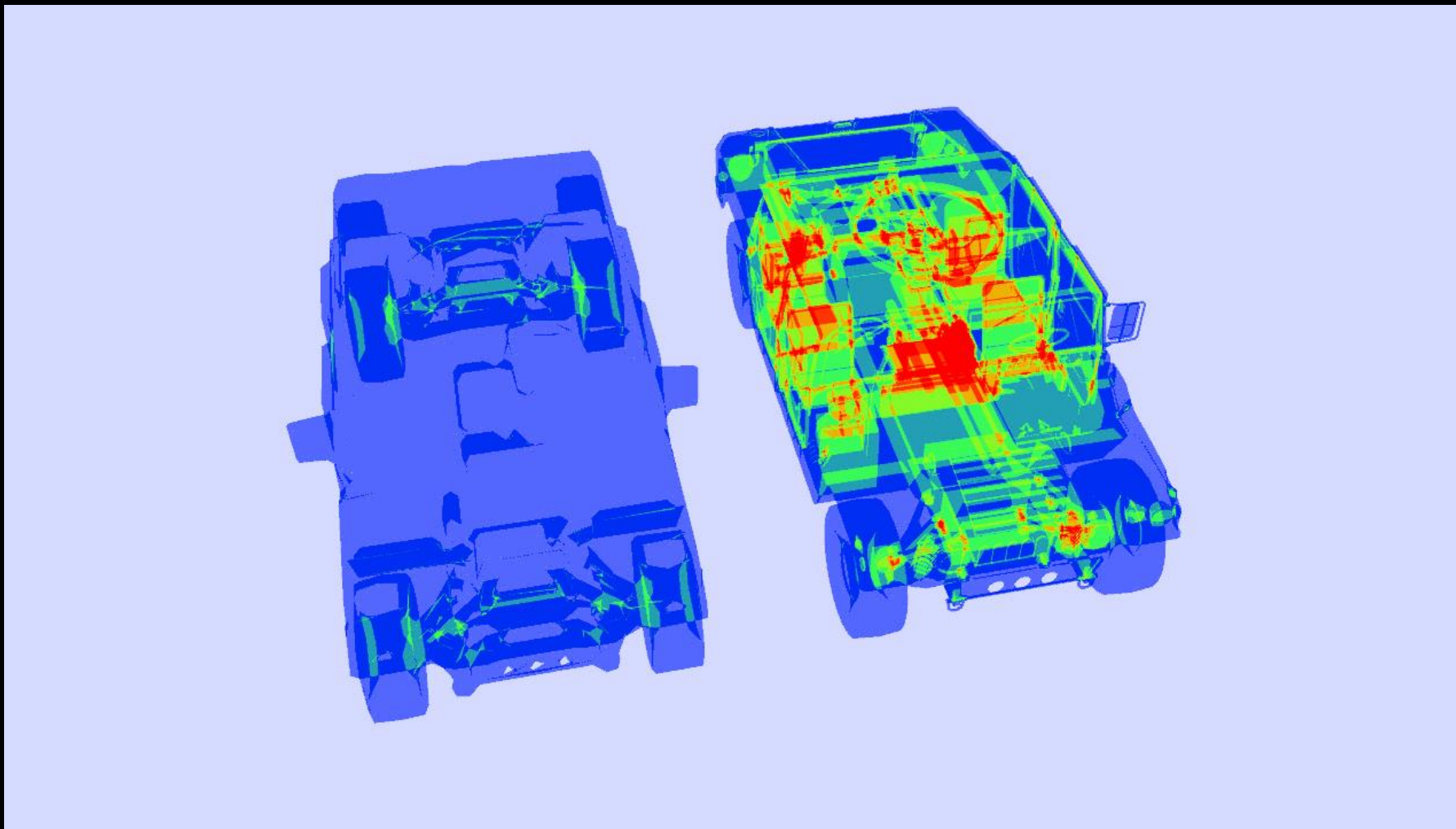
Proxy LOD

- “Shrinkwraps” and remeshes a set of objects
- Creates 2-manifolds
- Fills small holes
- Removes internal geometry

Overdraw – Depth complexity



Overdraw – Depth complexity



Proxy LOD Example



Proxy Mesh Characters

- Generate proxy mesh, and transfer skin to new mesh
 - E.g. as last LOD in LOD chain
- Use for
 - Reduce draw calls, shader cost
 - Customizing characters

Original
~100,000 Tris
200 Objects
9 Material sets

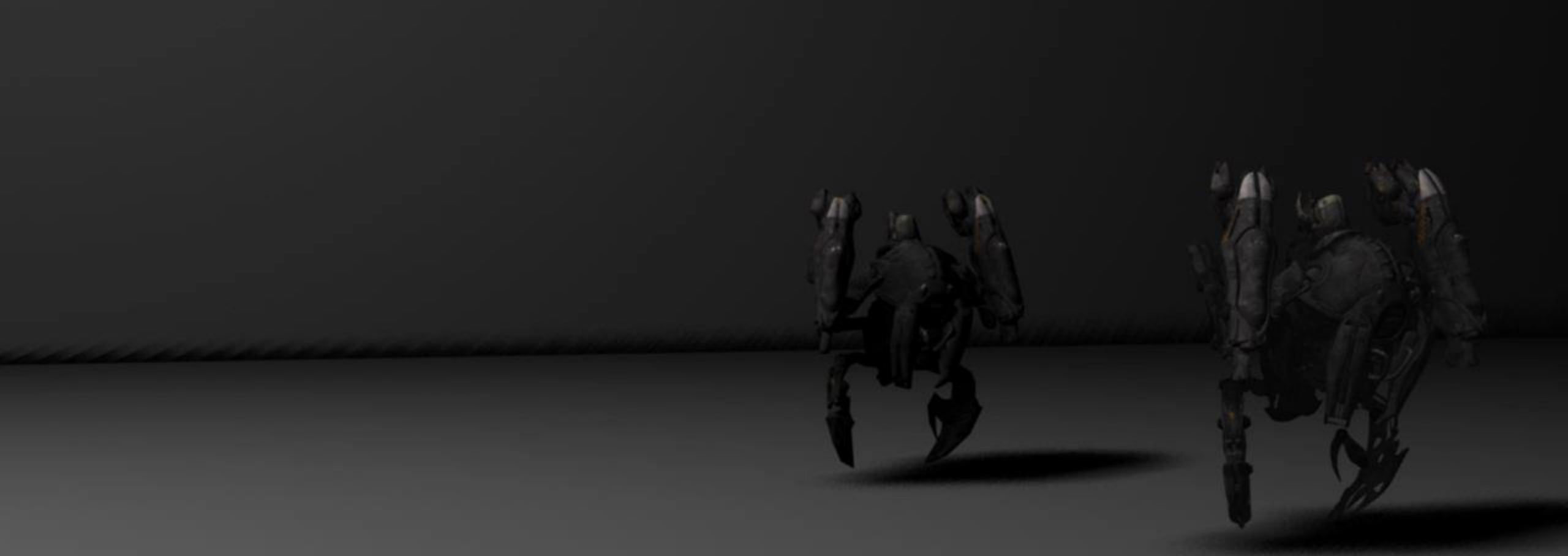
Proxy Mesh
~2000 Tris
1 Object
1 Material set

Proxy Mesh



Original





Samaritan Robot Asset © 2010-2013 Epic Games®, used with permission

Simplygon

Maya, Max, Softimage, UE

GUI

API

MeshLOD

ProxyLOD

MaterialLOD

BoneLOD

CPU – Setup/Animation

- LODs with fewer active bones
 - Fewer bones to animate, less CPU heavy
 - Fewer bones in matrix palette
 - Double win: Fewer bones needs fewer triangles.

Simplygon BoneLOD

- Automated solution for skeletal reduction
 - Automatic re-rigging of skinned mesh
 - Optionally lock or force removal of certain bones
- Available with MeshLOD & ProxyLOD
- Reduction based on number of bones or on-screen size
- Reduce max number of bones per vertex
 - Simpler vertex shader with guaranteed matrix palette size

Simplygon

Maya, Max, Softimage, UE

GUI

API

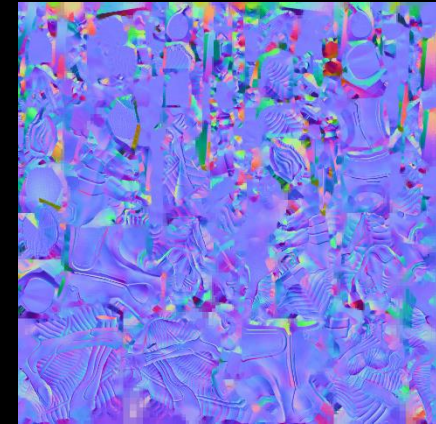
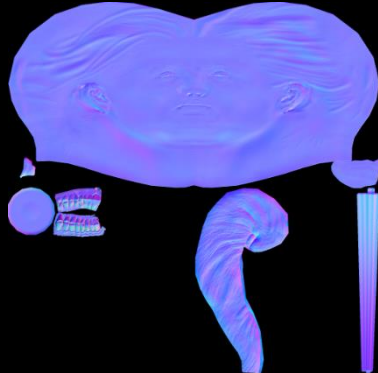
MeshLOD

ProxyLOD

MaterialLOD

BoneLOD

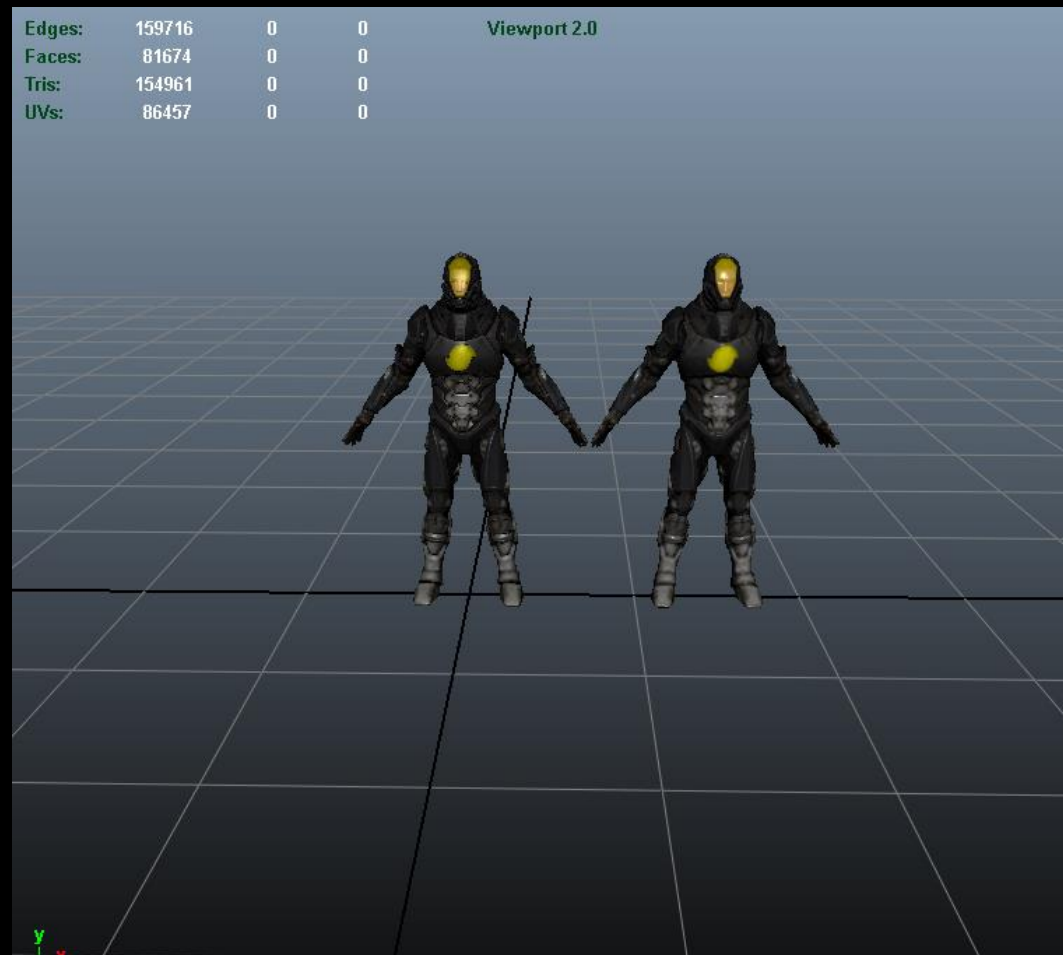
Material LOD



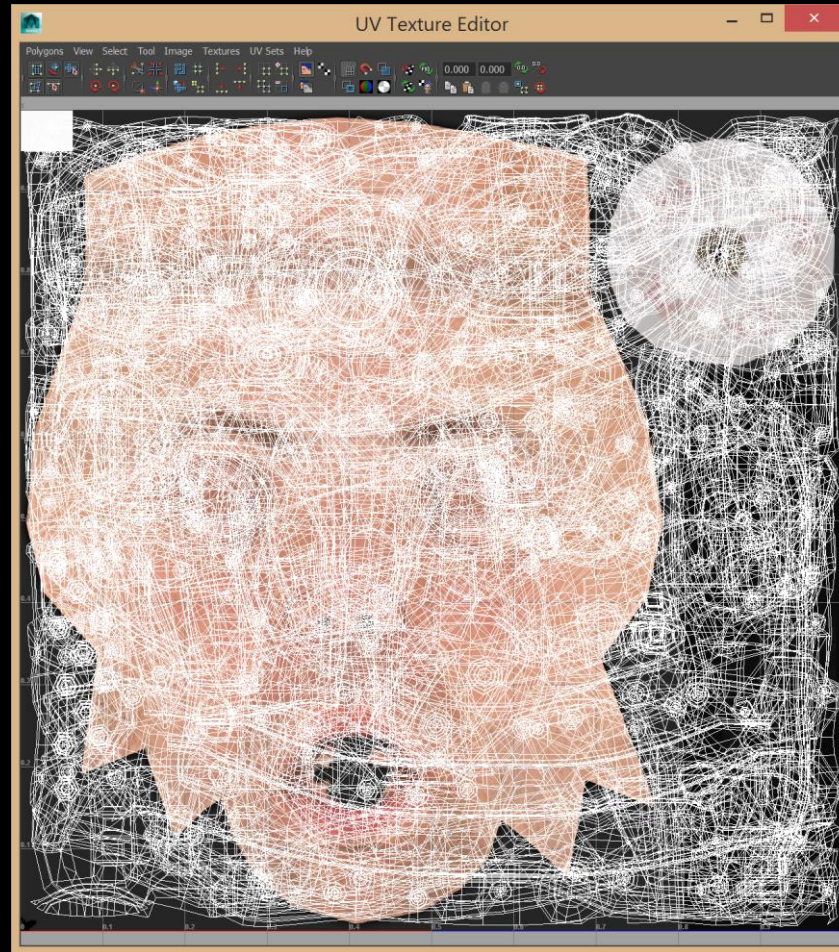
Original Textures

Merged Textures

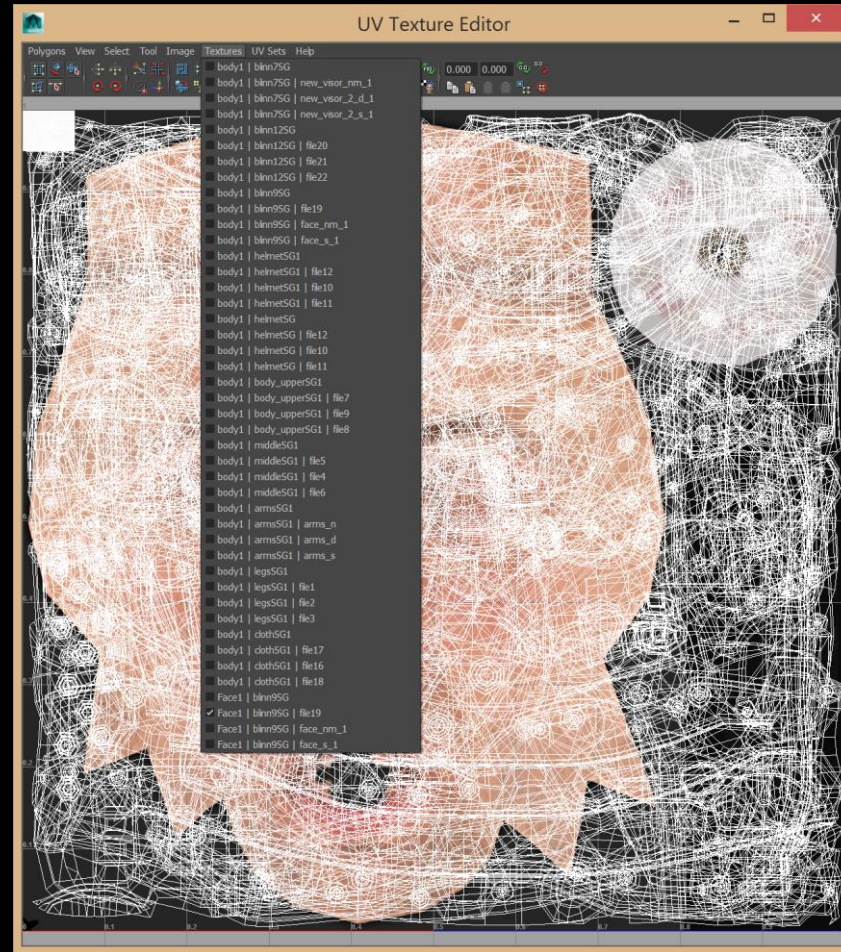
Material LOD



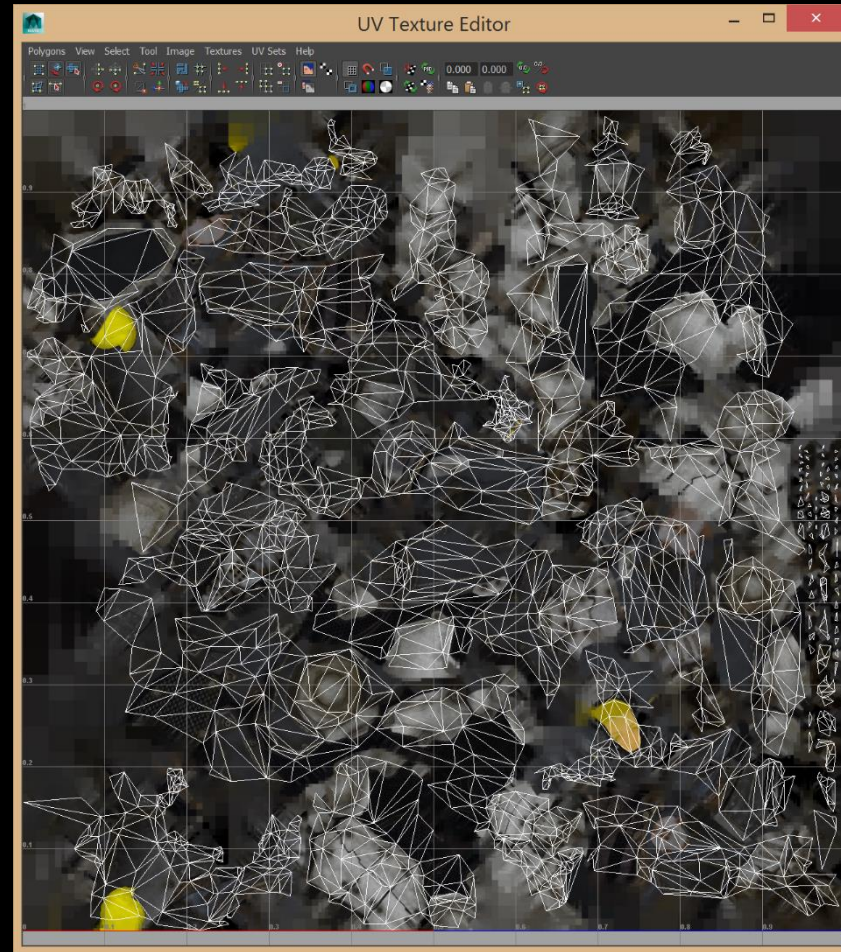
Material LOD Before



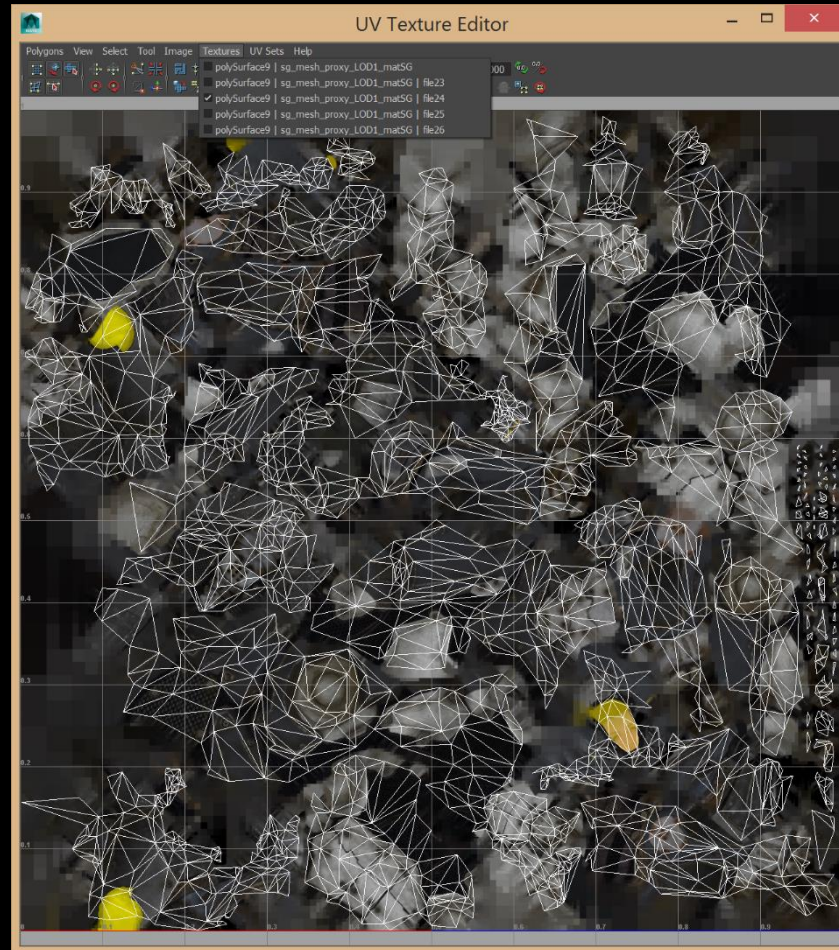
Material LOD Before



Material LOD Before



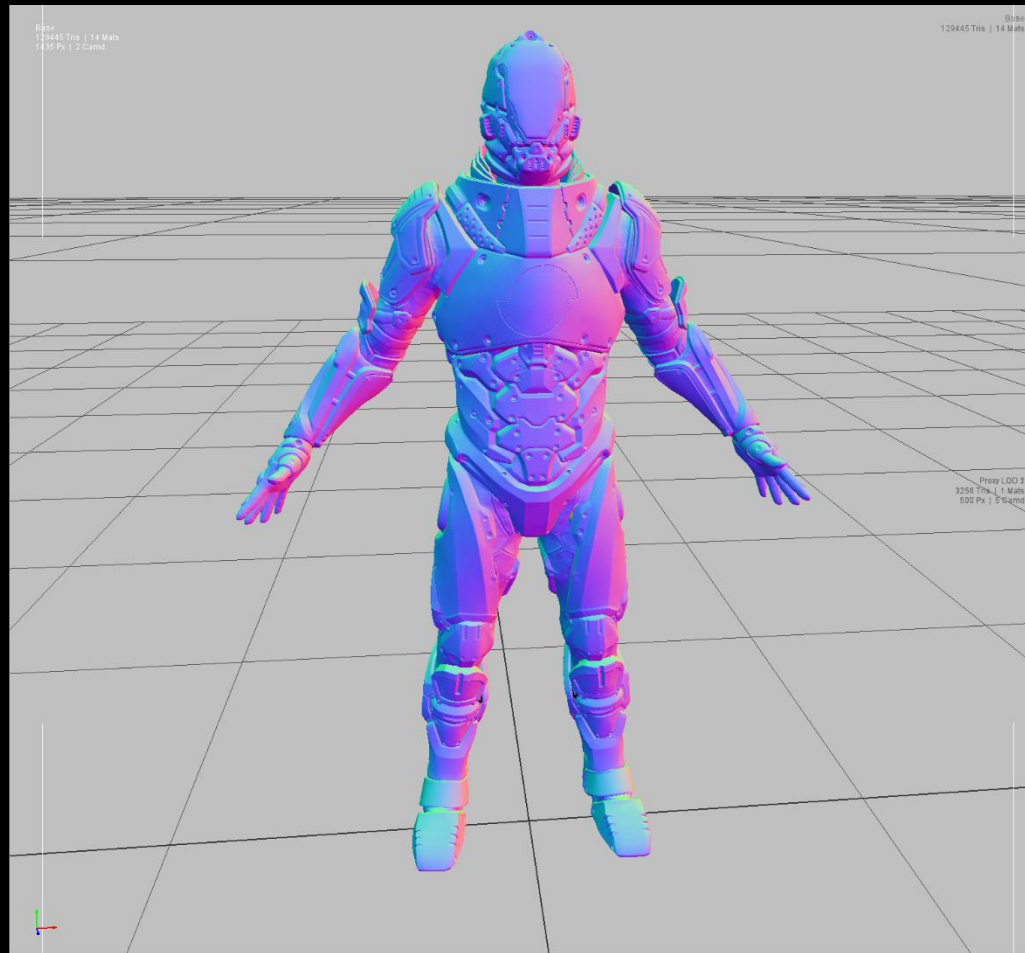
Material LOD



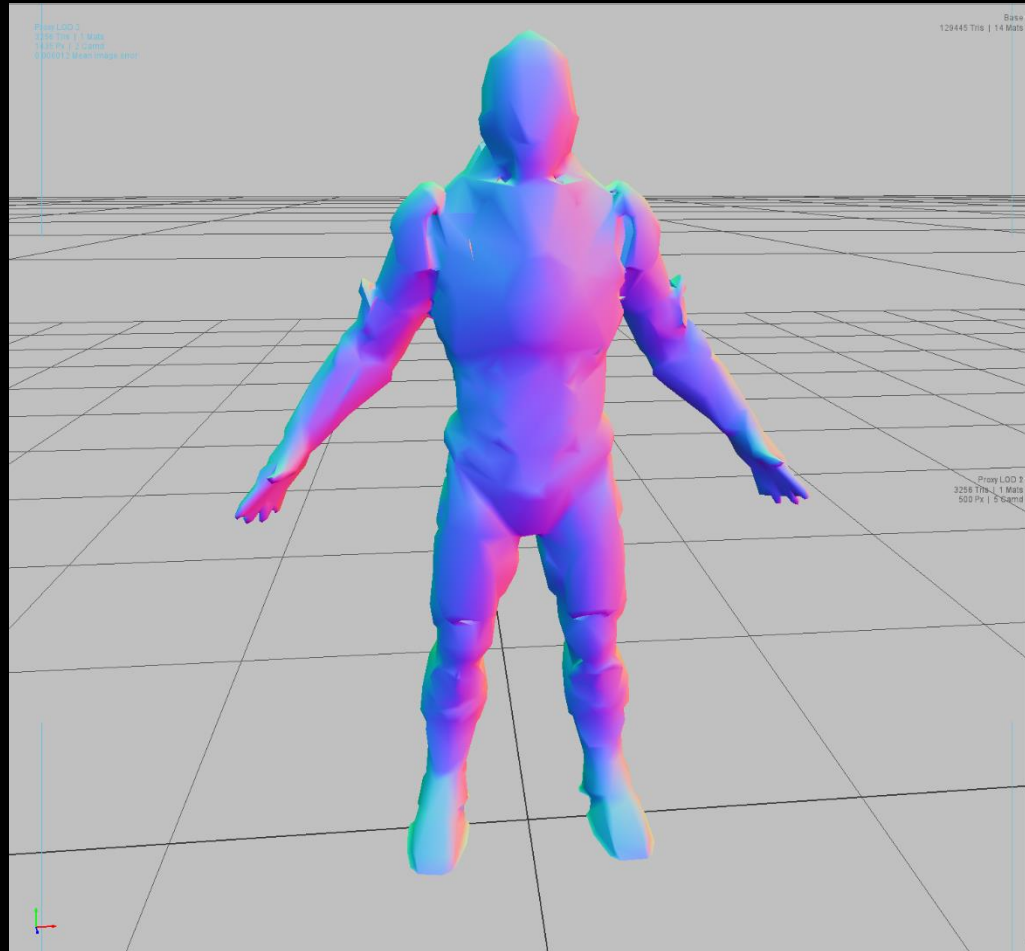
Normal Casting

- Can transfer details from original polygon content to normal map

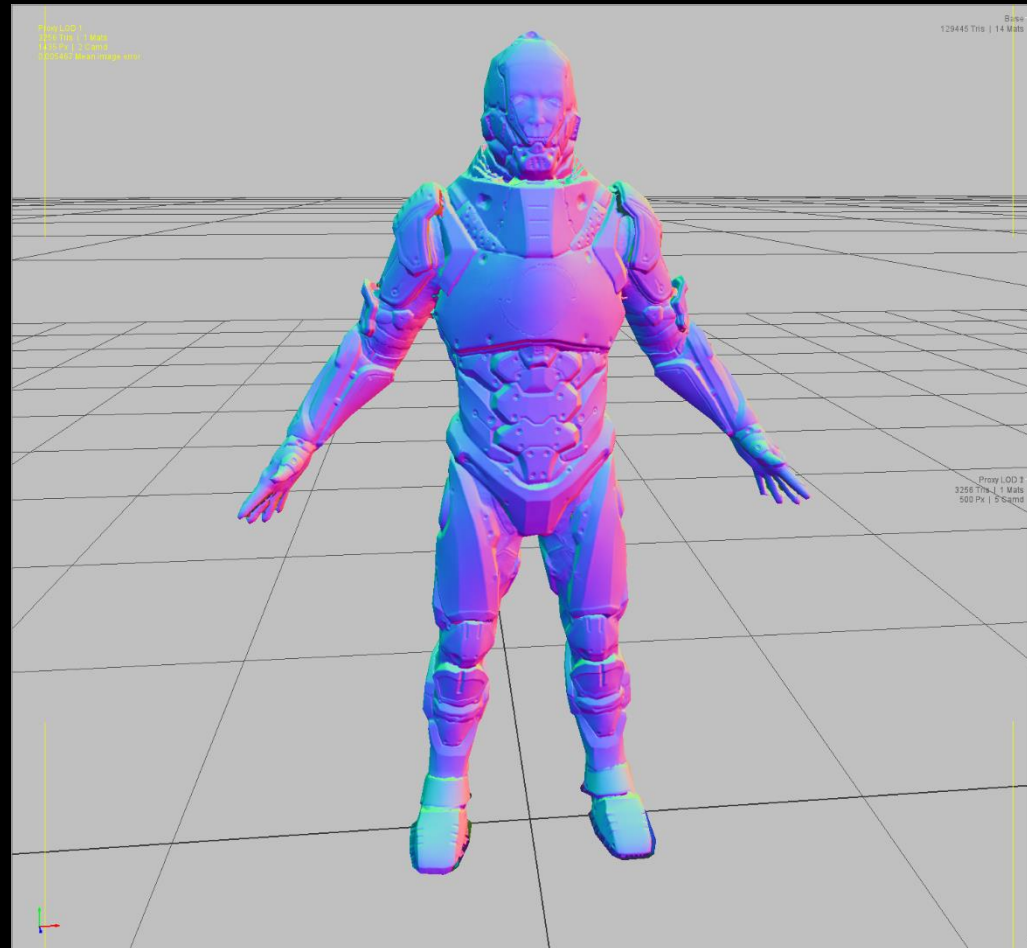
Normal Casting: Original



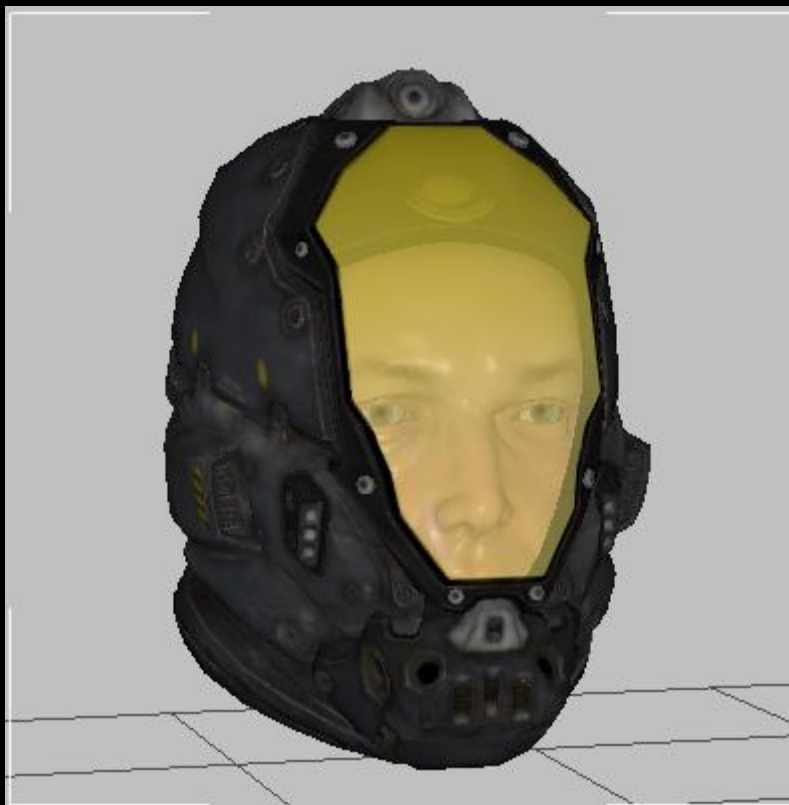
Normal Casting: Disabled



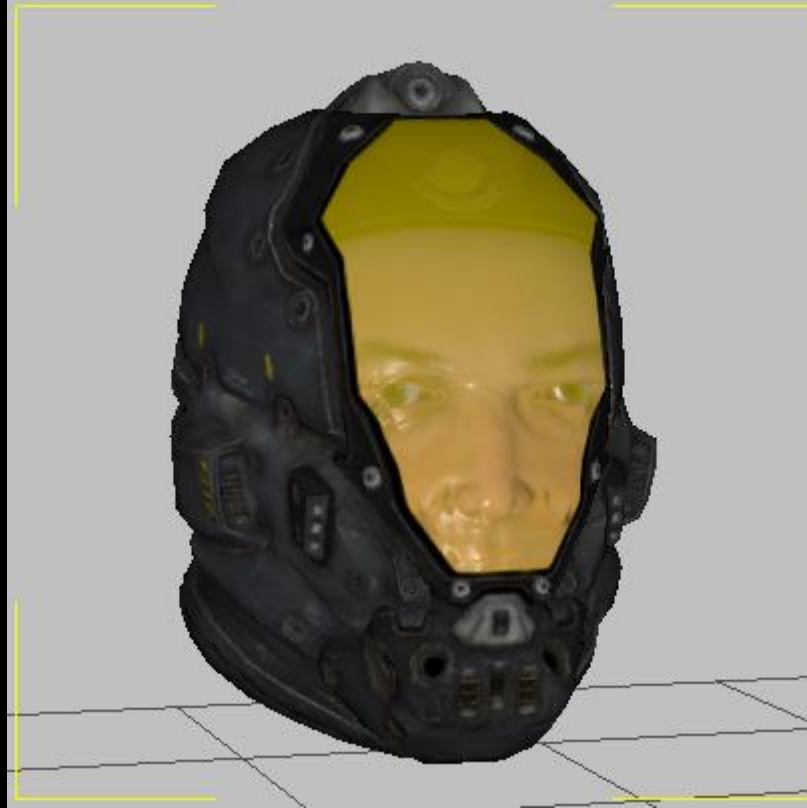
Normal Casting: Enabled



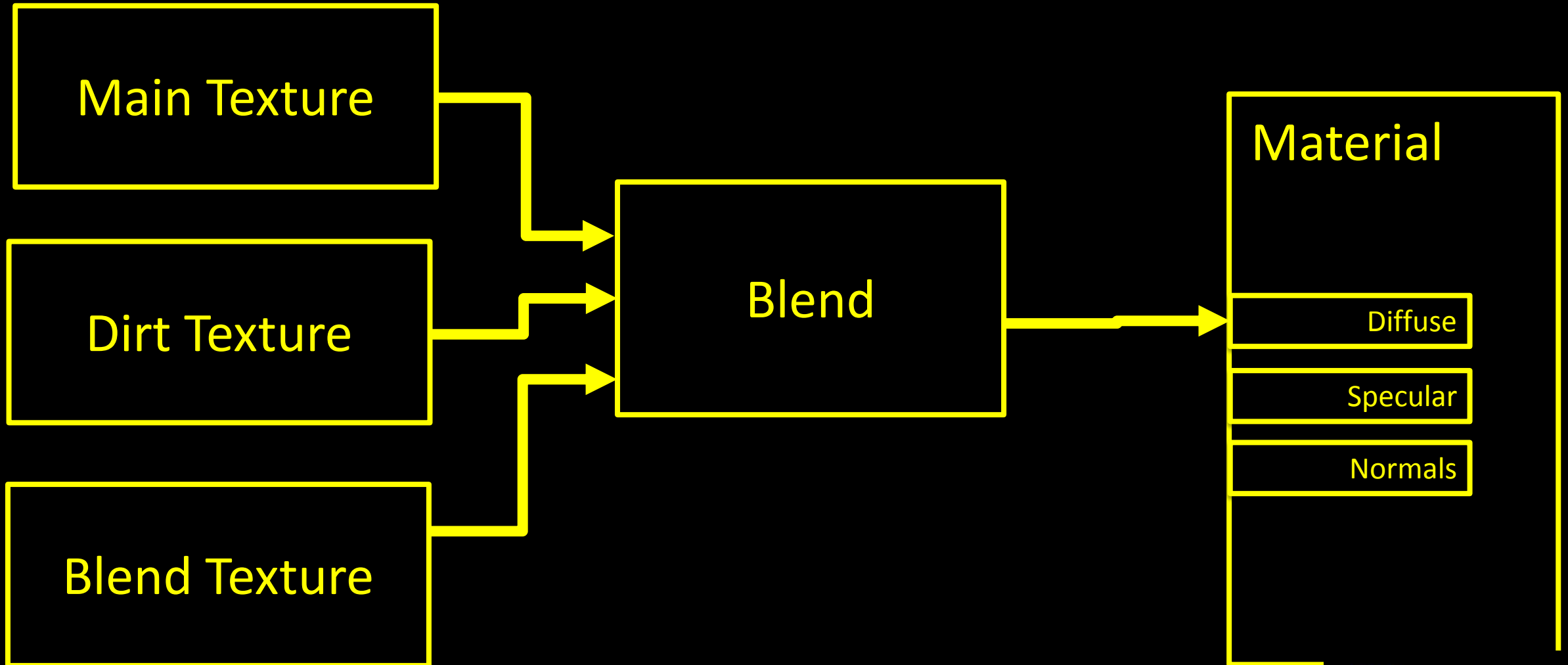
Transparency Projection: Original



Transparency Projection: LOD



Material System



Material System

- Node based system to express material blending
- Nodes includes
 - Texture fetch, vertex color, alpha blend etc
 - Allows creating advanced networks
- Fallback by providing new UV set with and per pixel mapping back to original asset
 - Flatten material channels yourself to our new shared UV

Special Guest

Balázs Török

Simplygon 6.0

- Material network baking
- Improved normal quality
- Automatic symmetry
- View-dependent processing
- Mesh aggregation





Optimize your 3D Assets with Simplygon

- Second Presentation
- Koshi from Simplygon
- Tramell Isaac from Sony Online Entertainment
- Thursday, March 20 | 11:30am-12:30pm
- Friday, March 21 | 10:00am-11:00am

Thank you!
Questions?

