

**K H R O N O S**<sup>®</sup>  
G R O U P

**Khronos**  **Tools**

**Norbert Nopper**  
**30<sup>th</sup> January 2020**



# Content

- Introduction
- Motivation
- Tools from the Khronos Group
  - Live session & slides
- Summary
- Questions & Answers

# Introduction

- Managing Director at UX3D



- Computer scientist
  - OpenGL, Vulkan, glTF ... anything GPU related
- Working more than two years on glTF with the 3D Formats group



<https://www.ux3d.io/>

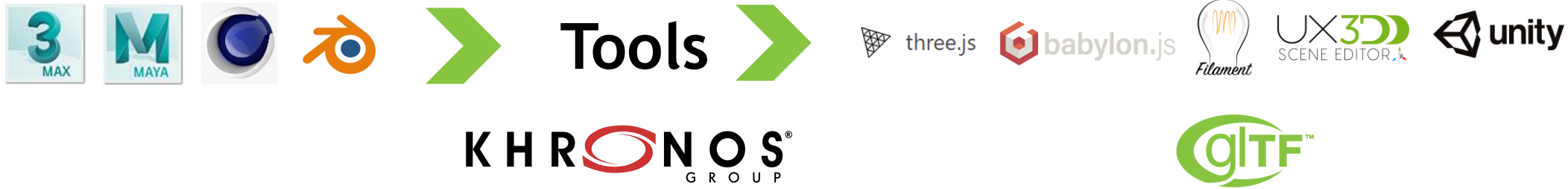
# Motivation



- Challenge from ...
  - Offline rendering
    - Ray / Path traced, Complex materials, High polygon models
- ... To
  - Real-time rendering
    - Rasterized, Baked materials, Low polygon models

Source: <https://blogs.nvidia.com/blog/2018/03/19/whats-difference-between-ray-tracing-rasterization/>  
<https://skfb.ly/6HwZR>

# Motivation (cont.)



- **Basic pipeline**
  - From DCC tools ...
    - 3DS Max, Maya, Cinema 4D, Blender etc.
  - ... to real-time 3D Engines
    - three.js, Babylon.js, Filament, UX3D, Unity etc.
- **Khronos provides free and open source tools!**

# Tools for Basic Pipeline

- glTF Blender glTF I/O
  - <https://github.com/KhronosGroup/glTF-Blender-IO>
  - Included in Blender since 2.80!
- glTF Sample Viewer
  - <https://github.com/KhronosGroup/glTF-Sample-Viewer>
  - Stable
  - <https://github.com/KhronosGroup/glTF-Sample-Viewer/tree/pbr-next>
  - WIP: Bug fixes, glTF PBR Next and KTX2 support
- glTF IBL Sampler
  - <https://github.com/KhronosGroup/glTF-IBL-Sampler>
- glTF Validator
  - <https://github.com/KhronosGroup/glTF-Validator>
- Much more
  - <https://github.com/KhronosGroup/> and look for glTF-\* projects

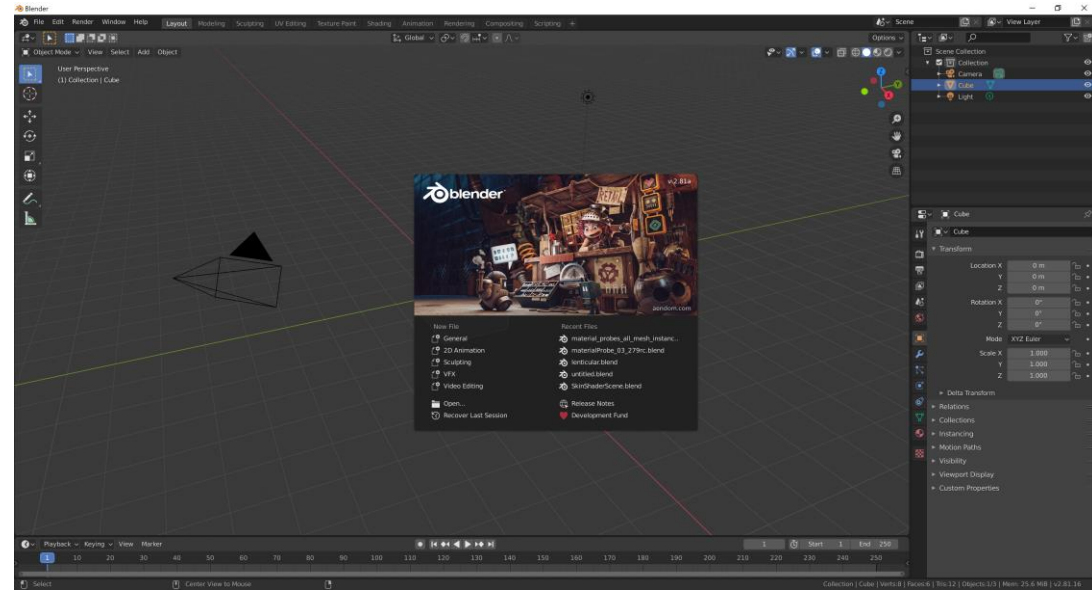
# Basic Pipeline



- Next: glTF Blender I/O

# glTF Blender I/O

- Live session
  - Import
    - \*.glTF
    - \*.obj
    - Assign materials
  - Camera
  - Lighting
    - Punctual
    - Caution: Light Power
    - Environment
  - Export
    - \*.glTF
    - Explain export settings





# Basic Pipeline (cont.)



- Next: glTF Validator

# glTF Validator

- Live session
  - Inspect glTF manually
  - Drag & drop into validator

KHRONOS<sup>®</sup>  
GROUP



Select glTF or GLB asset or drop it here.

*Validation is performed locally in your browser. Submitted assets are not uploaded.*

# Basic Pipeline (cont.)



- Next: glTF Sample Viewer

# glTF Sample Viewer

- Resources

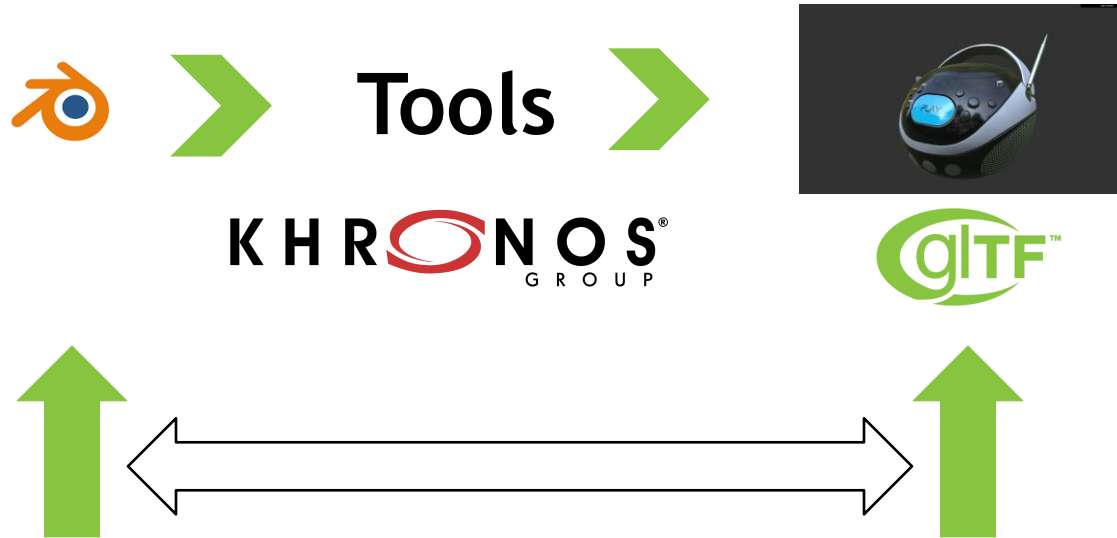
- <https://github.com/KhronosGroup/glTF-Sample-Viewer>
  - Stable
- <https://github.com/KhronosGroup/glTF-Sample-Viewer/tree/pbr-next>
  - WIP: Bug fixes, glTF PBR Next and KTX2 support
- Assets
  - <https://github.com/KhronosGroup/glTF-Sample-Models>
    - glTF models used in the glTF sample viewer
  - <https://github.com/ux3d/glTF-Sample-Environments>
    - Environments used for image based lighting (temporary)



# glTF Sample Viewer (cont.)

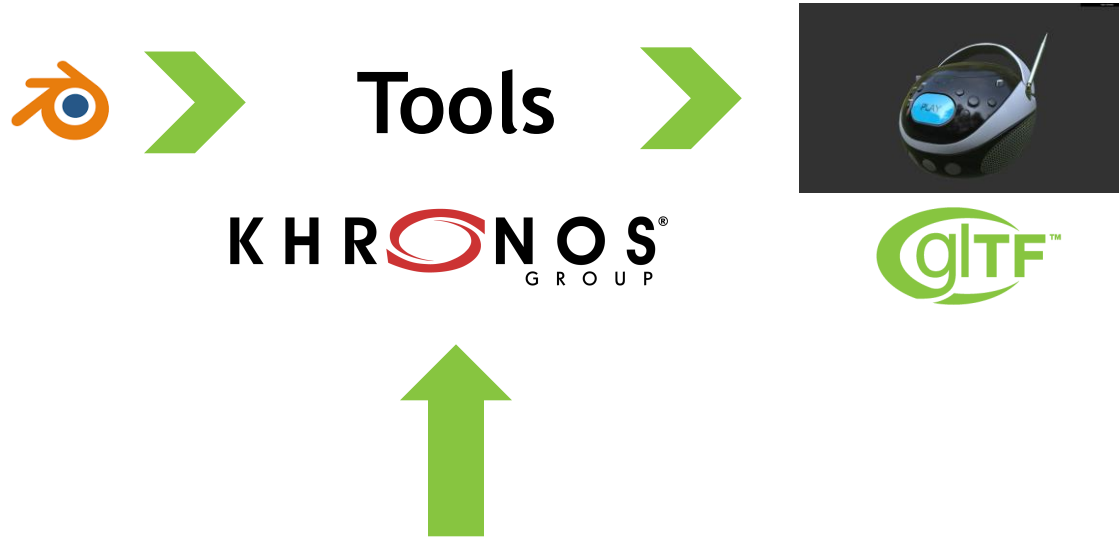
- Live Session
  - Main features
    - Using glTF
    - Camera
    - Light
      - Punctual
      - Image based
    - Animations
    - Debugging materials
  - Our exported glTF model
    - Drag & drop
    - Inspect & debug materials

# Basic Pipeline (cont.)



- Comparison: Blender vs. glTF Sample Viewer
  - Punctual light good
  - Environment light not (obvious)

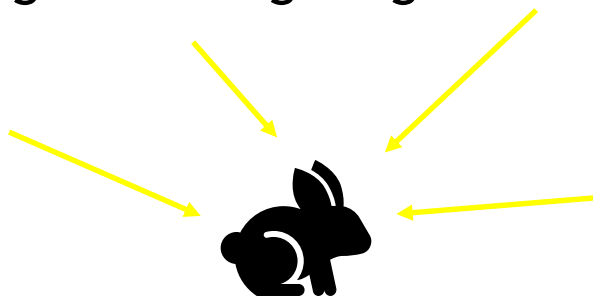
# Basic Pipeline (cont.)



- Next: glTF IBL Sampler

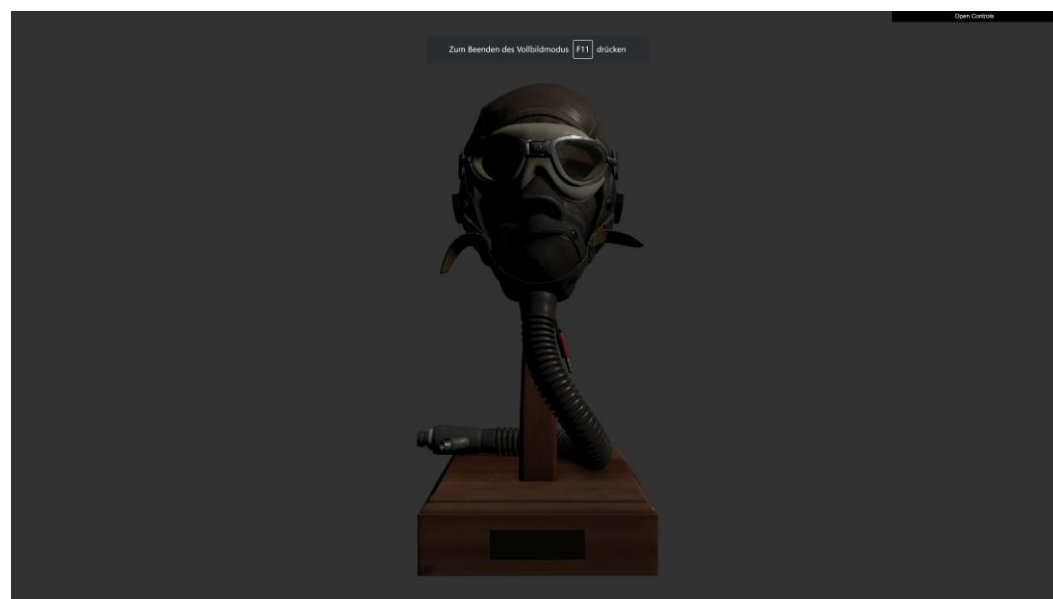
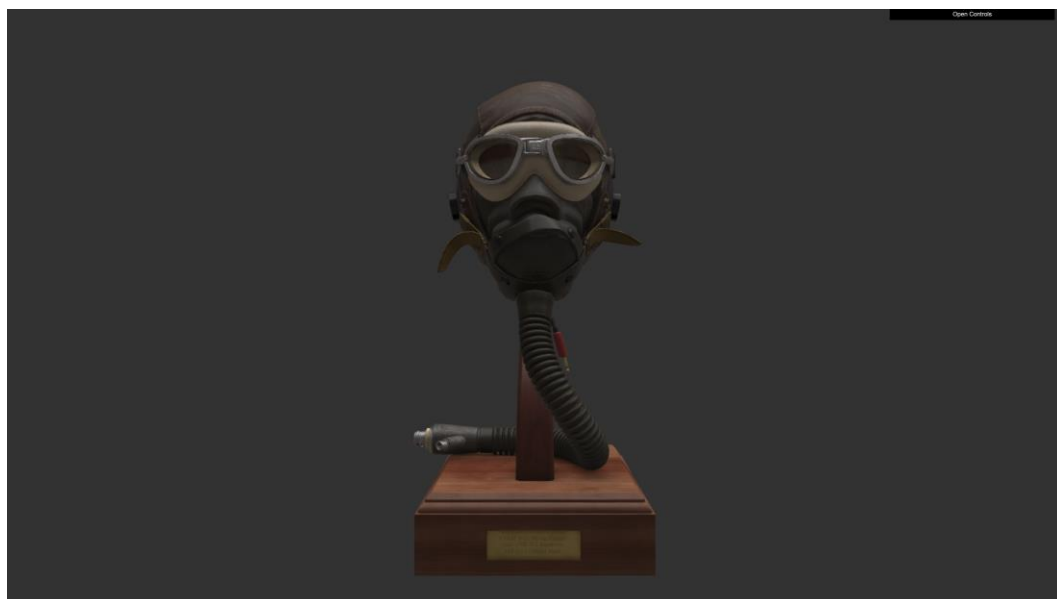
# glTF IBL Sampler

- Image Based Lighting (IBL)
  - Image based lighting



vs.

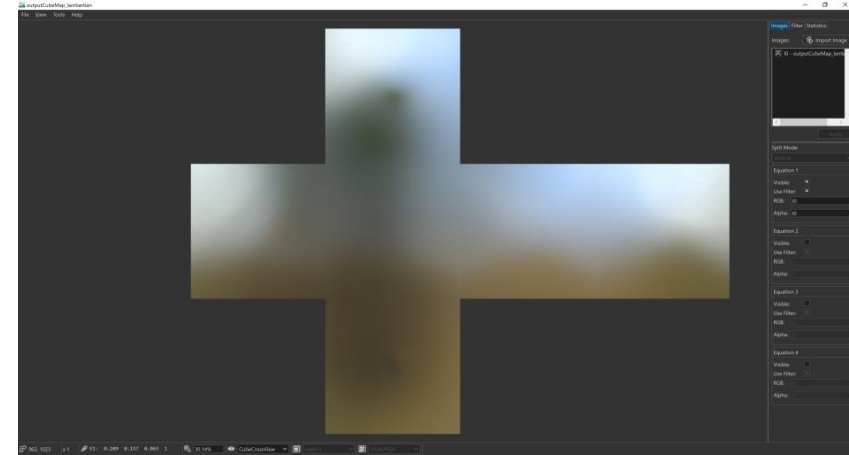
Punctual lighting



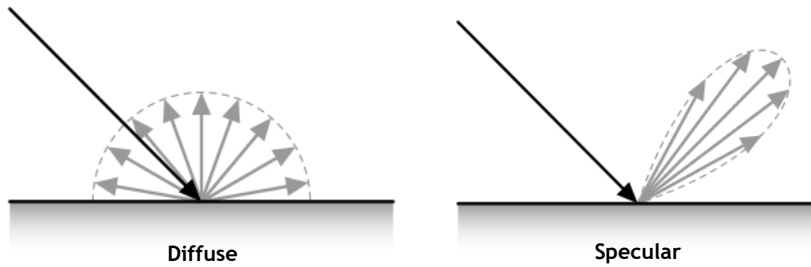


# glTF IBL Sampler (cont.)

- Image Based Lighting (IBL)
  - Source is a (static) image e.g. panorama photo



- Image need to be prefiltered

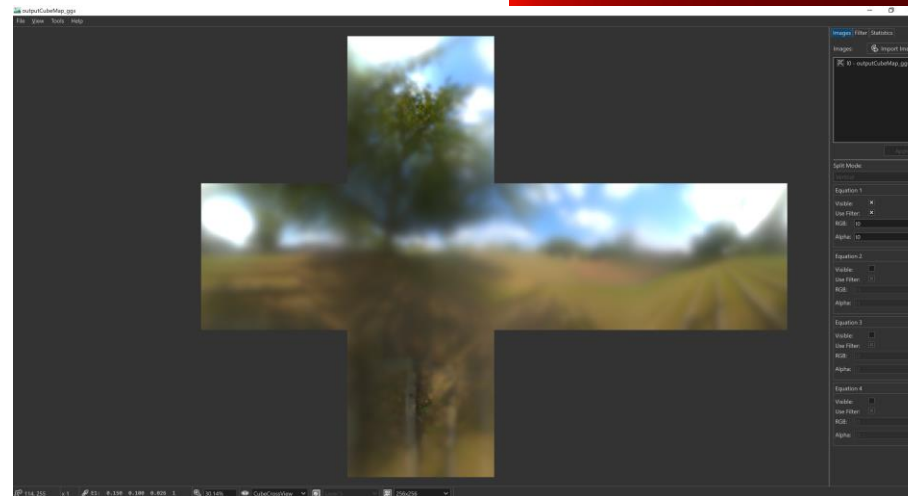
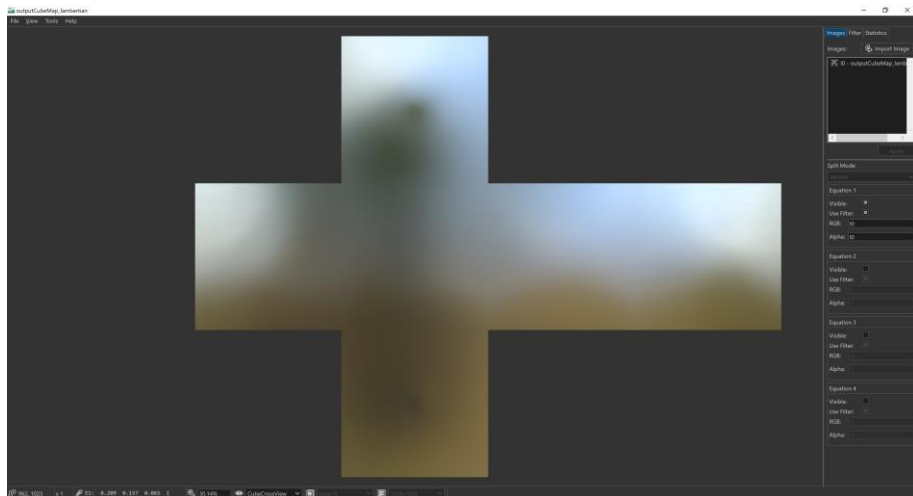
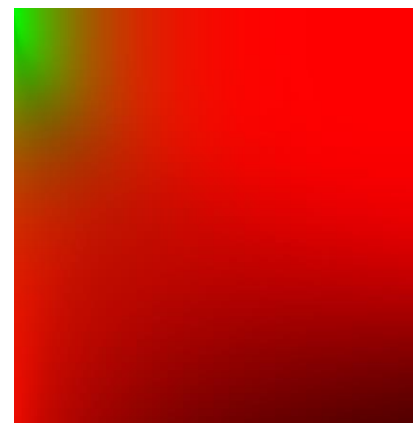


Source:

[https://en.wikipedia.org/wiki/Bidirectional\\_reflectance\\_distribution\\_function](https://en.wikipedia.org/wiki/Bidirectional_reflectance_distribution_function)  
<https://developer.playcanvas.com/en/user-manual/assets/cubemaps/>

# glTF IBL Sampler (cont.)

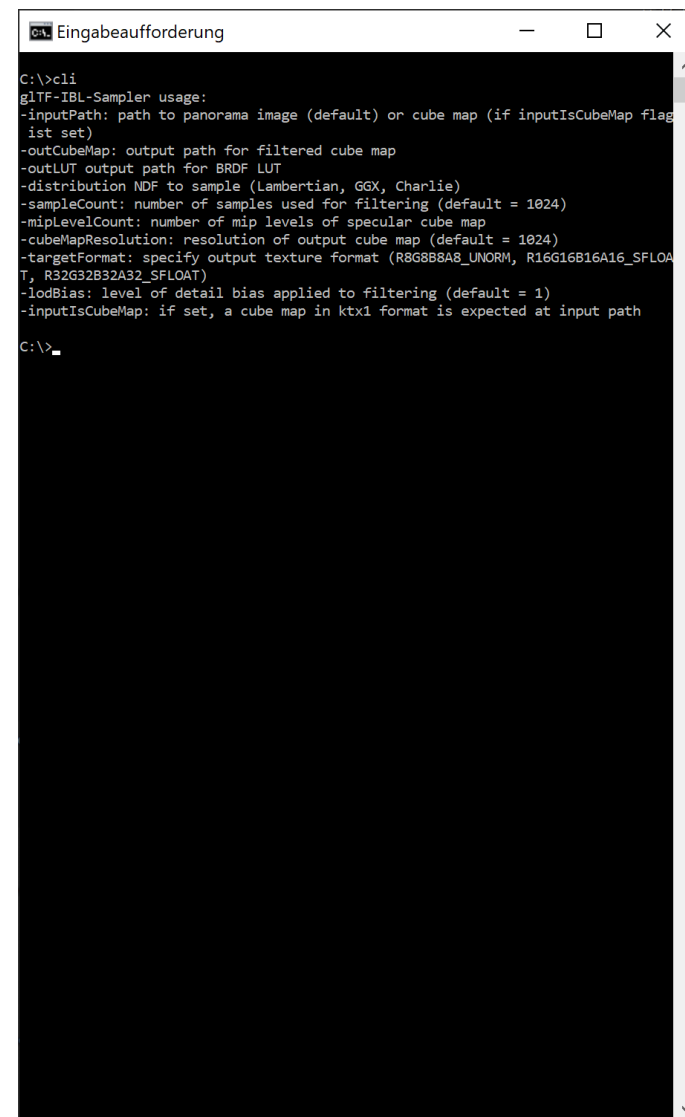
- Generates required prefiltered images and look up table
  - Diffuse
  - Specular
  - LUT
- Input \*.hdr and output is \*.ktx2



Source: [https://wiki.jmonkeyengine.org/jme3/advanced/pbr\\_part3.html](https://wiki.jmonkeyengine.org/jme3/advanced/pbr_part3.html)

# glTF IBL Sampler (cont.)

- Live session
  - Generating the output file



```
C:\>cli
glTF-IBL-Sampler usage:
-inputPath: path to panorama image (default) or cube map (if inputIsCubeMap flag is set)
-outCubeMap: output path for filtered cube map
-outLUT output path for BRDF LUT
-distribution NDF to sample (Lambertian, GGX, Charlie)
-sampleCount: number of samples used for filtering (default = 1024)
-mipLevelCount: number of mip levels of specular cube map
-cubeMapResolution: resolution of output cube map (default = 1024)
-targetFormat: specify output texture format (R8G8B8A8_UNORM, R16G16B16A16_SFLOAT, R32G32B32A32_SFLOAT)
-lodBias: level of detail bias applied to filtering (default = 1)
-inputIsCubeMap: if set, a cube map in ktx1 format is expected at input path

C:\>_
```

# Basic Pipeline (cont.)



- Next: glTF Sample Viewer (again)

# Deploying IBL

- **Either on server ...**
  - ... or integrate into glTF
  
- **Live session**
  - Manual integration in glTF

# glTF Sample Viewer (cont.)

- Live session
  - Drag and drop into glTF Sample Viewer again
  - Inspect again

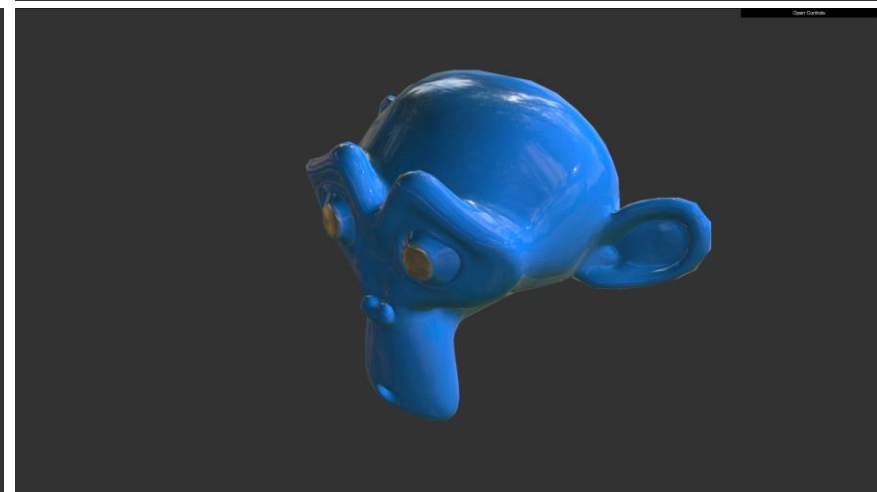
# Summary

- Comparison

Blender



Sample Viewer





# Questions and Answers

 @McNopper @UX3DGpuSoftware

