glTF Overview
Efficient, reliable transmission of 3D Assets

Neil Trevett
Khronos President
VP Developer Ecosystems, NVIDIA
ntrevett@nvidia.com | @neilt3d
January 2021
# glTF - The JPEG of 3D!

## glTF 1.0 - December 2015
- Primarily for WebGL
- Uses GLSL for materials

## glTF 2.0 - June 2017
- Native AND Web APIs
- Physically Based Rendering
  - Metallic-Roughness and Specular-Glossiness

### Compact to Transmit
### Simple and Fast to Load
### Describes Full Scenes
### Runtime Neutral
### Open and Extensible

### Efficient, reliable transmission
Bring 3D assets into 1000s of apps and engines

### New market opportunities for 3D content creation and deployment!

<table>
<thead>
<tr>
<th>Audio</th>
<th>Video</th>
<th>Images</th>
<th>3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP3</td>
<td>H.264</td>
<td>JPEG</td>
<td>glTF</td>
</tr>
</tbody>
</table>

*This work is licensed under a Creative Commons Attribution 4.0 International License*
Core glTF 2.0 Asset Structure

- `.gltf (JSON)`
  - Node hierarchy, PBR material textures, cameras

- `.bin`
  - Geometry: vertices and indices
  - Animation: key-frames
  - Skins: inverse-bind matrices

- `.png`
  - Textures

- `.jpg`
  - Texture-based PBR materials

- `.ktx2`
  - PBR materials

---

PBR stands for “Physically-Based Rendering”

**Mandatory Metallic-Roughness Materials**
- Base Color (Albedo) | Metalness | Roughness
- Emission | Normal Map | Baked Ambient Occlusion

**Optional Specular-Glossiness Materials**
- Diffuse | Specular | Glossiness
This work is licensed under a Creative Commons Attribution 4.0 International License

https://github.khronos.org/glTF-Project-Explorer/
glTF Draco Mesh Compression Extension

- glTF extension for compressed geometry
  - Typically 10-25x geometry size reduction

- Google Draco technology - designed for decompression efficiency and speed
  - [https://github.com/google/draco](https://github.com/google/draco)

- Draco geometry encoders and decoders in open source
  - C++ source code encoder
  - C++ and JavaScript decoders
  - [https://github.com/google/draco/tree/gltf_2.0_draco_extension](https://github.com/google/draco/tree/gltf_2.0_draco_extension)

- glTF Draco compression adoption is growing in tools, applications and engines
  - glTF pipeline, FBX2glTF, AMD Compressonator and glTF sample models

![Mesh Compression Ratios](https://example.com/mesh-compression-ratios.png)
The Need for Universal Textures

How to create a glTF asset that can be used everywhere?
Without packing multiple copies of the textures (increases file size)
Without expensive conversion (consumes time and power)
glTF Universal Textures

- Compact, high-quality textures that are efficient to deploy on diverse target platforms
  - Supercompressed and transcodable on-the-fly to compressed GPU-accelerated formats

- Uses Binomial’s Basis Universal compression technology with two compression options
  - Highest Quality: RDO-encoded, block-compressed UASTC with optional zstd compression
  - JPEG-sized textures: Block-compressed ETC1S with custom LZ supercompression

- KTX 2.0 container for consistent, cross-vendor asset distribution
  - Includes access to separate mipmap levels to enable LOD streaming

- Open-source KTX tools
  - For texture creation, validation, inspection and efficient transcoding
**glTF Universal Textures: Compression Ratios**

**FlightHelmet_baseColor**
2048 x 2048, RGB

- **PNG and JPEG must be fully decompressed into GPU memory**

- **Universal textures can be directly transcoded to compressed GPU textures**

<table>
<thead>
<tr>
<th>Format</th>
<th>File Size</th>
<th>GPU Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncompressed</td>
<td>12,582,912</td>
<td></td>
</tr>
<tr>
<td>PNG</td>
<td>2,778,518</td>
<td></td>
</tr>
<tr>
<td>JPEG</td>
<td>335,619</td>
<td></td>
</tr>
<tr>
<td>ETC1S</td>
<td>2,097,152</td>
<td>232,104</td>
</tr>
<tr>
<td>ETC1S-basisLZ</td>
<td>232,104</td>
<td>0</td>
</tr>
</tbody>
</table>

This work is licensed under a Creative Commons Attribution 4.0 International License
glTF PBR Materials Roadmap

Creating a rich physically-based material framework for the glTF ecosystem

- glTF extensions add PBR material parameters that integrate with existing materials
- Building consensus on interoperable PBR deployable on diverse platforms and devices

Water Bottle sample is CC0, by Microsoft

June 2017
Core glTF 2.0
Mandatory Metallic-Roughness
Optional Specular-Glossiness

December 2020
First Wave glTF PBR Extensions
Clear Coat
Transmission
Sheen

Future Waves of glTF PBR Extensions
Subsurface Scattering, Attenuation, Index of Refraction (IOR), Thickness, Specular Color, Anisotropy, Translucency, Thin Film (iridescence) and more...

This work is licensed under a Creative Commons Attribution 4.0 International License
New Wave PBR Ecosystem Forming Quickly

Tools to Create and View PBR Assets

AUTODESK®
3DS MAX®
Adobe
blender

Online Model using Babylon.js

PBR Samples

glTF Sample Viewer

glTF®
Validator

PBR Capable Viewers and Applications

three.js
Microsoft
babylon.js
UX3D
GESTALTOR
emersya
AGI
STK

Input welcome on GitHub from developers and artists
Let us know what PBR materials are important to you!
Get Involved!

• glTF specification and resources
  - https://github.com/KhronosGroup/glTF

• glTF specification feedback on GitHub
  - https://github.com/KhronosGroup/glTF/issues

• Explore the glTF ecosystem
  - https://github.khronos.org/glTF-Project-Explorer/

• We want to know what you need next from glTF on the Khronos Forums!
  - https://community.khronos.org/c/gltf-general/45

• Engage with Khronos and help glTF evolve
  - Join as a Khronos member for a voice and a vote in any Khronos standard
  - https://www.khronos.org/members/

• Neil Trevett
  - ntrevett@nvidia.com
  - @neilt3d