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!

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

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

glTF spec development
on open GitHub - get involved!
https://github.com/KhronosGroup/glTF
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**
- **.jpg**
- **.ktx2**
  - Textures

**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

Texture based PBR materials

Geometry
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 Compressorator and glTF sample models
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

FlightHelmet_baseColor
2048 x 2048, RGB

PNG
JPEG
ETC1S
Basic Universal ETC2

File Size
GPU Size

12,582,912
2,776,518
315,619
2,097,152
232,104

14,000,000
10,500,000
7,000,000
3,500,000
0
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

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...

Water Bottle sample is CC0, by Microsoft

https://belcour.github.io/blog/research/2017/05/01/brdf-thin-film.html

Roadmap includes requirements from Khronos 3D Commerce Working Group

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

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](https://github.com/KhronosGroup/gltF)

- **gltF specification feedback on GitHub**
  - [https://github.com/KhronosGroup/gltF/issues](https://github.com/KhronosGroup/gltF/issues)

- **Explore the glTF ecosystem**
  - [https://github.khronos.org/gltF-Project-Explorer/](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](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/](https://www.khronos.org/members/)

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