glTF Update
webGL/webVR/glTF Meetup
February 2017

Tony Parisi
Patrick Cozzi
Saurabh Bhatia
glTF - The Standardized 3D Delivery Format

Audio
MP3

Video
H.264

Images
JPEG

3D

New market opportunities for 3D content creation and deployment!

model/gltf+json MIME type Approved by IANA
Compact to Transmit
Fast to Load
Describes Full Scenes
Runtime Neutral
Extensible
Strong glTF Momentum

Oculus Executive Calls For 3D Equivalent Of JPEG To Build The Metaverse

A new standard for 3D scenes is gaining momentum with support from graphics industry leader, potentially laying the groundwork for science fiction's "metaverse" to be realized.

The GL Transmission Format (gltF) from The Khronos Group, a computer graphics industry standards body, could also put magnitudes more 3D content on the Internet. The Khronos Group is responsible for a variety of technologies critical to the 3D industry today, such as OpenGL and Vulkan. With the growing interest in virtual and augmented reality, the glTF standard provides a compelling solution for streaming 3D models and scenes over the Internet. By establishing a shared standard for 3D content, the glTF format helps accelerate the development of immersive experiences that could redefine how we interact with digital content.
glTF Milestones

- **2012 thru 2014**: Design iteration and Multiple Implementations
  - Original motivation: standardized way to deliver 3D into WebGL applications

- **Dec 2015**: glTF 1.0 Spec Ratified and Released
  - glTF 2.0 Target Spec Finalization
  - Significant Industry Adoption

- **Oct 2016**: Validator Project

- **Spring 2017**: We are here! Seeking feedback on developer preview glTF 2.0 before finalization
  - glTF 2.0 adds Physically Based Rendering for higher-quality materials and API independence
glTF 2.0

- Graphics API neutral
  - Proven by implementations using WebGL, Vulkan and Direct3D
  - GLSL materials moved to extension — enabled by Physically Based Rendering (PBR) in core
- Major new functionality in glTF 2.0 core
  - PBR — for higher quality materials AND API independence
  - Binary glTF
  - Morph targets
  - Textures can be accessed from binary buffers, not just images
  - Sparse buffer views
- Enhanced Performance
  - Changed top-level object properties to arrays — better performance
  - Removed indirection when possible, e.g., animation, skins
- Has breaking changes!
  - Community input and significant implementation experience
  - Tightening up corner cases through careful validator implementation
  - Community told us ‘better now than later’ — it's a better spec
glTF 2.0 Scene Description Structure

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

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

- `.jpg`
  ... Textures

- Geometry
- Texture based PBR materials
glTF 2.0 Ecosystem Updates

• Industry moving quickly to glTF 2.0 — lots of early adopters
  - BabylonJS, three.js, Cesium, xeogl, instant3Dhub

• gltf-pipeline includes glTF 2.0 updates — including converting glTF 1.0 models to 2.0
  - Use this to support both glTF 1.0 and 2.0 or move your users to 2.0!

• Converters and Translators
  - COLLADA2GLTF glTF 2.0 updates nearly ready
  - Major refactoring for easier code contributions
  - obj2gltf glTF 2.0 updates are in progress

• Tests and Validators
  - Validator is being updated for glTF 2.0 support
  - Gltf-test @cx20 is adapting to glTF 2.0

• Tutorials and Samples
  - glTF 2.0 sample models with PBR are emerging
  - Extensive glTF tutorial series in draft

• glTF 2.0 being integrated into other formats / standards
  - 3D Tiles stream massive heterogeneous 3D datasets
  - Spatial data structure + glTF + declarative styling
  - Proposed OGC Community Standard
Exporters!  Blender glTF 2.0 RFQ

- Still need more direct exporter support!
  - Tools vendors – give your product a market advantage – support glTF 2.0 export!

- Khronos Blender RFQ
  - [https://www.khronos.org/rfq/](https://www.khronos.org/rfq/)
  - Blender, the open source 3D authoring tool, has started development of an open-source exporter of assets in the glTF file format: [https://github.com/Kupoman/blendergltf](https://github.com/Kupoman/blendergltf).
  The goal of this project is to build on, and expand, the current exporter so that it robustly supports full glTF 2.0. The resultant exporter code is to be contributed, royalty-free to the Blender open source project

- RFQ Milestones
  - 1. February 27th — Khronos Releases RFQ
  - 2. March 17th — RFQ responses received by Khronos
  - 3. March 24th — Contractor selected and notified
  - 4. March 29th — Contract executed and start of work

---

*glTF 2.0 PBR Rendering - Image courtesy Fraunhofer*
glTF Roadmap Discussions

- Extensions for API and language specifics
  - DX12, Metal, GLSL, HLSL, SPIR-V, Metal C++

- Mesh Compression
  - Talking to Draco team at Google

- Geometry Streaming
  - Fraunhofer SRC?

- Unified Compressed Texture Format for Transmission
  - Binomial Basis — optimized transmission format with local expansion to any GPU format

- Extension for lights
  - Enhanced lighting control

- Feedback on priorities is welcome!
Calls to Action

• Evaluate/implement glTF 2.0
  - glTF 2.0 draft spec: https://github.com/KhronosGroup/glTF/tree/2.0/specification/2.0
  - glTF 2.0 Sample Models https://github.com/KhronosGroup/glTF-Sample-Models

• Bid on the Blender exporter RFQ!
  - Help support the glTF exporter ecosystem https://www.khronos.org/rfq/

• Tell us what you are building with glTF
  - Send a pull request to add it to our list — let us help promote you work

• glTF Online Resources
  - Resource Hub https://www.khronos.org/gltf/
  - Tools Page https://github.com/KhronosGroup/glTF#gltf-tools
  - Validator Open Source http://github.khronos.org/glTF-Validator/

• Join Khronos!
  - Get directly involved in the glTF Working Group
glTF 2.0 Physically Based Rendering

- Standardize the BRDF inputs for common PBR workflows
  - Metallic-Roughness and Specular-Glossiness
- Incredible industry effort
  - Started by Fraunhofer and supported by Microsoft, Sketchfab, NVIDIA, Autodesk, Marmoset, University of Pennsylvania, and others

Sketchfab User: theblueturtle
https://sketchfab.com/models/b81008d513954185a063ff901f7abfe4
glTF 2.0 Physically Based Rendering

- In Core: Metallic-Roughness Material model
  - baseColor — base color
  - metallic — metalness
  - roughness — roughness
- Simple to implement with small resources
  - Can be everywhere

- Extension: Specular-Glossiness Material model
  - diffuse — reflected diffuse color
  - specular — specular color
  - glossiness — glossiness
- A little more resource heavy
  - Optional extension (e.g. on low-power devices)

- The two models can be combined
glTF 2.0 PBR materials in various engines

WebGL reference implementation
http://www.seas.upenn.edu/~moneimne/WebGL-PBR/

Laugh Engine running on Vulkan
https://github.com/jian-ru/laugh_engine
### glTF 2.0 Implementations

<table>
<thead>
<tr>
<th>Company, Institute</th>
<th>Software/Engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft</td>
<td>3D for everyone</td>
</tr>
<tr>
<td>Gary Hsu</td>
<td></td>
</tr>
<tr>
<td>Fraunhofer</td>
<td>Instant3Dhub, InstantUV</td>
</tr>
<tr>
<td>Max Limper</td>
<td></td>
</tr>
<tr>
<td>Sketchfab</td>
<td>Sketchfab</td>
</tr>
<tr>
<td>Cedric Pinson</td>
<td></td>
</tr>
<tr>
<td>University of Pennsylvania</td>
<td>WebGL reference implementation</td>
</tr>
<tr>
<td>Mohamad Moneimne</td>
<td></td>
</tr>
<tr>
<td>University of Pennsylvania</td>
<td>Laugh engine on Vulkan</td>
</tr>
<tr>
<td>Jian Ru</td>
<td></td>
</tr>
</tbody>
</table>
Bonus Slides
glTF Validator

- Started out as glTF 1.0 project
  - Ended up informing how to clean-up glTF 2.0
  - Will handle glTF 2.0 once spec is finalized
- In open source on GitHub
  - Khronos funded open source project
- Rigorous checking for correctly formed glTF files
  - Checks JSON syntax, all property details, GL parameter combinations etc. etc.
- Built using Dart (easy API level integration)
  - Extensible - validation plugins for extensions
  - Output can be integrated into the validation report
- Client-side drag-n-drop and command-line wrapper
  - Client-side JavaScript library coming soon

Please give us feedback on GitHub!
glTF Ecosystem

Tools
- Blender DIRECT export
- glTF open source Validator (will support glTF 2.0)

Translators
- Unity to Sketchfab Exporter
- Drag and Drop FBX -> glTF (coming soon)
  - http://gltf.autodesk.io/
- Other Translators
  - AssImp
  - OBJ2GLTF
  - glTF Pipeline
  - COLLADA2GLTF
  - Cesium converter
- Drag and drop COLLADA -> glTF
  - http://cesiumjs.org/convertmodel.html

Validator
- glTF open source Validator

Apps & Engines
- A-FRAME
- BABYLON
- Cesium
- three.js
- PEX
- xeoEngine

Import
- Blender DIRECT export
- Export
- Convert | Optimize
- Validate