glTF Update and Roadmap

July 27, 2016

Tony Parisi
glTF Specification Co-Editor

tparisi@gmail.com @auraduluxe
Background and Motivation

- OpenGL ES and WebGL have led to a proliferation of 3D, but no standard way to deliver data into applications
  - Efficient transmission
  - Full scene information
  - Vendor- and runtime-neutral

  **OBJ - too simple - COLLADA - too bulky - FBX - vendor-specific**

- As a result, content and app creators were developing a new pipeline for each project
  - Huge inefficiencies
  - Limited opportunities for sharing data among applications
The “JPEG of 3D”

- Compact to Transmit ✓
- Fast to Load ✓
- Runtime Neutral ✓
- Extensible ✓
glTF Structure

- **.gltf**: JSON describes node hierarchy, materials, cameras
- **.bin**: Geometry: vertices and indices, Animation: key-frames, Skins: inverse-bind matrices
- **.glsl**: Shaders
- **.png**, **.jpg**: Textures

Describes full scenes—not just meshes

NORAD’s Santa Tracker
Extensions

- Syntax and name registry for extending the base specification
  - KHR_binary_gltF - binary container format, single payload - ratified
  - KHR_materials_common - common fixed function materials and lights e.g.
  - Vendor extensions e.g. CESIUM_RTC, WEB3D_quantized_attributes

- Keeps the base spec small while allow for experimentation and domain-specific use cases

- Popular extension can potentially be promoted to the base spec

File declares extensions used up front

```
  "extensionsUsed" : [ "KHR_binary_gltF"
  ]
```

“extensions” property contains the data

```
  "a_shader" : {
    "extensions" : {
      "binary_gltF" : {
        "bufferView" : // ...
      }
    }
  }
```

Adoption

Publicly Stated Support for glTF
With New Formats Come New Opportunities!

<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>

- napster
- YouTube
- Facebook
- ?!
glTF Ecosystem News!

Drag and Drop FBX -> glTF
(coming soon)
http://gltf.autodesk.io/

Autodesk FBX -> glTF
AssImp
OBJ2GLTF
glTF Pipeline
COLLADA2GLTF
Cesium converter

Drag and drop COLLADA -> glTF
http://cesiumjs.org/convertmodel.html

Tools

Export

Validator

Validate

Translators

Convert | Optimize

glTF

Apps & Engines

Import

model/gltf+json MIME type Approved by IANA!

glTF 1.0.1 Spec in Review and glTF Validator in open source!
http://github.khronos.org/glTF-Validator/

glTF Ecosystem Page
https://github.com/KhronosGroup/glTF-glTF-tools
glTF 1.0.1 Validator

- glTF 1.0.1 tightens specification
  - For robust validation and interoperability
    https://github.com/KhronosGroup/glTF/issues/605

- Validator in open source on GitHub
  - Khronos Validator project RFQ awarded to Alexey Knyazev - doing awesome work!
  - 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)
  - Shipping today as client-side drag-n-drop and command-line wrapper
  - Client-side JavaScript library coming soon
  - Extensible - validation plugins for extensions - output can be integrated into the validation report

Please give us feedback on GitHub!
Roadmap Discussion Topics

- Physically Based Rendering
  Modern, compact, scalable
  Fraunhofer, NVIDIA MDL

- Streaming and Mesh Compression
  MPEG 3DGC (royalty-free), Fraunhofer SRC

- Enhanced API Support
  Make efficient use of
  WebGL 2.0 and Vulkan

- Enhanced Animation
  Morph Targets

- Advanced Surfaces
  Pixar’s OpenSubdiv?

- Increased Efficiency
  Improved parsing, arrays, bounding boxes, spatial constructs

Must avoid the complexity trap!
Core glTF must remain efficient and straightforward to use
Extensions for domain specific functionality

Come to the glTF Community on GitHub
https://github.com/KhronosGroup/gltf
Or join Khronos to get directly involved!
T-Shirts!

Efficiently describe and transmit your 3D scenes!