glTF and Mobile VR:
Inclusive standards for a 3D world

Amanda Watson
Oculus Mobile SDK
Choosing a Scene Format for Mobile VR

• For content creators, mobile VR can be an exciting new medium, but a challenging new platform
  - Severe resource constraints, even for graphics devs
  - Absence of vetted tools/best practices
• Formats for mobile VR should be chosen to minimize its barrier to entry
  - Support inclusive standards that make it easy to integrate with major packages
  - Prioritize community tools that can open up development to a range of artists, engineers and contributors.
Creating a Scene Format for Mobile VR

ovrscene: Oculus scene format
- High performance run times, rendering
- Designed, deployed, maintained internally
- Deployed to developers via FBXConvert

- Significant advantages to choosing external, community-driven (open) standard over an in-house format:
  - Interoperability
  - Support from the greater community
  - Step towards a healthy ecosystem

The *inclusive* 3D formats aren’t open
- Industry standards tend to be ones that are heavily tailored to a platform

The *open* 3D formats aren’t performant
- Probably where all those other standards came from
Creating a Scene Format for Mobile VR

models.json
- Ordered list of “surfaces”/materials
- Indices into models.bin

models.bin
- Geometry: vertices and indices (sorted, pre-filtered)

.gltf
- JSON describes node hierarchy, materials, cameras

.models.bin
- Geometry: vertices and indices (sorted, pre-filtered)

.models.json
- Ordered list of “surfaces”/materials
- Indices into models.bin

.models.bin
- Geometry: vertices and indices (sorted, pre-filtered)

.models.json
- Ordered list of “surfaces”/materials
- Indices into models.bin

.models.bin
- Geometry: vertices and indices (sorted, pre-filtered)

.models.json
- Ordered list of “surfaces”/materials
- Indices into models.bin

.models.bin
- Geometry: vertices and indices (sorted, pre-filtered)

.models.json
- Ordered list of “surfaces”/materials
- Indices into models.bin

.models.bin
- Geometry: vertices and indices (sorted, pre-filtered)

.models.json
- Ordered list of “surfaces”/materials
- Indices into models.bin

.models.bin
- Geometry: vertices and indices (sorted, pre-filtered)

.models.json
- Ordered list of “surfaces”/materials
- Indices into models.bin

.models.bin
- Geometry: vertices and indices (sorted, pre-filtered)
FBX2glTF: built with VR in mind

- Conversion tool based on in-house FBX->.ovrscene converter
- Definitely a *subset* of glTF (for now)

- **Merging meshes**
  - Draw calls incur substantial driver overhead on mobile
  - Merges meshes that use the same material into a single surface.

- **Pre-filtering vertices**
  - Removing attributes unnecessary for rendering reduces the data set and improves the cost of lookups and storage

- **Remapping textures to adjust LOD**
  - Static scenes can simplify their textures by baking in the optimal LOD for a fixed vantage point

- **Pre-compressing textures**
  - Image files compressed with GPU compression formats

- **Sorting geometry**
  - Rendering front-to-back is optimal on modern GPUs
  - Perfect sorting for scenes with fixed/limited vantage points

- **Texture Atlas**

---

On The Roadmap

- Open Source/Khronos Release
- Collision detection/gaze selection
- Support for animations
Most Thrilling Demo You Will See at SIGGRAPH

Loading with glTF

Loading with ovrscene