3D Transmission Format

Neil Trevett
Khronos President
Vice President Mobile Content, NVIDIA
June 2013
Topics

- Industry need for transmission format
- Related Khronos activities - COLLADA and glTF
- Need for wider industry cooperation for complete transmission format solution
- Brainstorm ideas for cooperation next steps
3D Needs a Transmission Format!

• Compression and streaming of 3D assets becoming essential
  - Mobile and connected devices need access to increasingly large asset databases

• 3D is the last media type to define a compressed format
  - 3D is more complex - diverse asset types and use cases

• Needs to be royalty-free
  - Avoid an ‘internet video codec war’ scenario

• Eventually enable hardware implementations of successful codecs
  - High-performance and low power - but pragmatic adoption strategy is key

<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>?</td>
</tr>
</tbody>
</table>

An effective and widely adopted codec ignites previously unimagined opportunities for a media type
COLLADA - Authoring Interchange of 3D Assets

• COLLADA is an ISO-standard XML-based database schema for 3D assets
  - Can hold everything to do with a scene: geometry with full skinning, advanced material and visual effects, animation, physical properties and collisions

• COLLADA can be used to transport 3D assets between tools
  - Enables binding of diverse DCC and 3D processing tools into a production pipeline

• COLLADA is an open, archive-grade format that retains meta information
  - Retains all information - even multiple versions of the same asset

• COLLADA is NOT an transport format
  - Conditioning pipelines optimize the asset database for a target device or use case
COLLADA Conditioners Directory

- Encourage community to share COLLADA conditioners
OpenCOLLADA High Quality Import/Export

• OpenCOLLADA upgraded and released as open source on Github
  - High-quality, full functionality COLLADA importer/exporter
  - Available free-of-charge for tools vendors to integrate
  - https://github.com/KhronosGroup/OpenCOLLADA

• COLLADA conformance test suite released on GitHub
  - Enables tools vendors, and their users, to test for conformant COLLADA IO
  - https://github.com/KhronosGroup/COLLADA-CTS

No excuses left for missing or low quality COLLADA import/export! 😊
glTF - Transmission Format for WebGL

Binary 3D asset transmission format
Connects Authoring and Playback Ecosystems
Foundational - for use by any app or run-time

Authoring
Industry’s most portable and reliable
3D asset authoring interchange format

Playback
Industry’s most pervasive and secure
3D execution platform

A standards-based content pipeline for rich native and Web 3D applications
glTF Goals

• Binary file format for efficient transmission for 3D assets
  - Reduce network bandwidth and minimize client processing overhead

• Run-time neutral - DO NOT IMPLY OR MANDATE ANY RUN-TIME BEHAVIOR
  - Can be used by any app or run-time - usually WebGL accelerated

• Scalable to handle compression and streaming
  - Though baseline format does not include compression

• ‘Direct load efficiency’ for WebGL
  - Little or NO processing to drop glTF data into WebGL client

• Carry conditioned data from any authoring format
  - Prototype and optimize efficient handling of COLLADA assets
glTF Structure

- JSON used to describe node hierarchy
  - Platform neutral, run-time neutral, many processing libraries available
- Node hierarchy refers to EXTERNAL binary asset blobs
  - Geometry, Textures and Materials
- Non-compressed asset blob format enables WebGL ‘direct load’
  - E.g. Use typed arrays format directly
- Extensible to asset blob formats with streaming and compression
  - Extensions in Khronos registry define how to handle a new blob format

Careful design of glTF uncompressed binary is giving good loading performance boost - even before compression

Khrnos wishes to leverage and coordinate with industry-wide initiatives for streaming and compression
Building glTF

• glTF specification being driven at COLLADA working group
  - Draft is openly available - no Khronos membership needed
  - https://github.com/KhronosGroup/glTF
  - Industry input and feedback very welcome

• Open Source COLLADA to glTF Convertor - COLLADA2GLTF - on Github page
  - Uses OpenCOLLADA to convert COLLADA assets to JSON for use in WebGL
  - High-performance model loading and animations
  - Shader generation from COLLADA materials
Building a Complete Transmission Solution

• Three components need for complete solution
  - JSON Scenegraph encoding - glTF + ✓
  - REST3D APIs for Server/Client Negotiation +
  - 3D Asset payload compression/streaming

1. Send JSON encoded scene graph

2. Use REST to negotiate asset selection, compression and streaming

3. Compress and Stream Assets
Assets as Web Service - 3D Streaming Spectrum

- Many methods of serving assets
  - Best technique depends on capabilities of server AND client AND the use case

- Use Rest API to negotiate capabilities and preferred formats
  - Eliminates need to bake all use cases into complex monolithic format
  - BUT must minimize number of client/server round trips
  - Enables industry to incrementally roll-out new payload formats over time

- What REST standardization do we need?
  - Work with existing initiatives? E.g. Rest3D [https://github.com/amd/rest3d](https://github.com/amd/rest3d) or XML3DRepo?

Request serving order of geometry and texture assets
Request compression format best suited to use case

Serve Rendered Image for requested FOV

Compression and Streaming Formats:
- Request FOV -> entire scene
- Coarse geometry -> fine geometry
- Coarse textures -> fine textures

Serve complete asset database

Spectrum of Compression Possibilities

High Compression

Low Compression
Asset Compression/Streaming

• Want to work TOWARD hardware 3D CODEC implementations
  - But will probably need adoption through JavaScript implementations first

• Need range of geometry compression options - Lossy vs. Lossless
  - Gzip is baseline for compression efficiency

• Many, many initiatives in the industry
  - MPEG 3D Mesh Coding Progressive Streaming (3DMC), Bones Based Animation (BBA)
  - OCG 3D Portrayal - lossless png for geometry and REST APIs for request section of city model
  - OpenCTM - Compressed Triangle Mesh - http://opencmt.sourceforge.net/

• Should focus on common low-level operation for use from JavaScript Codecs?
  - E.g. INFLATE/DEFLATE and LZMA etc. etc...

• How many compression formats are needed?
  - What is the right balance between Darwinian choice and Fragmentation?

• The Web Way is to enable many JavaScript codec implementations
  - And let the market decide before native/hardware implementations
Industry Liaison Models

- Khronos originally suggested liaison for joint specification
- All participating SDOs would commit to ratifying and publishing agreed specification (royalty-free)

Single Specification for ‘3D Transmission’

- COORDINATE which SDOs should do which piece of the problem
  - Consider core competencies
  - Parallel development

SEPARABLE but Normatively LINKED Specifications

Learned more about problem topology

+ other SDOS
Industry Coordination Discussion

Rest APIs for 3D

XML3DRepo?
Rest3D?

Layered Run-time Stack

web3D consortium
x3dom

Asset Compression and Streaming

web3D consortium
UMPEG
OpenCTM
Google

Is glTF meeting industry needs? Have we tried x3dom loaders using glTF – any issues? Khronos establishing open mechanisms for feedback and design contributions. Is this enough?

Work with Rest3D or create different REST initiative? Where? How? What?

How do we try/propose/select compression technologies? Need royalty-free ratification at an SDO to be candidate? Do we need asset DRM? High-level or low-level standardized components? Industry-wide call for proposals? What is the selection process?
Liaison Proposal for Discussion

Khronos immediately open glTF registry to any extension format
Enables everyone to leverage glTF low-level foundation for WebGL clients/apps

• Period for widespread implementations and experimentation in apps/run-times (WebGL/X3D/X3DOM/MPEG)
  • Industry Cooperation Group/Wiki for collation/discussion of asset format proposals
    - Format proposals
      (with IP Royalty-free Provenance and link to open source JavaScript implementation)
      - Test data sets
      - Benchmark framework
      - Results matrix

Engage Rest community
Requirements and methodology

Promote multiple widely used, ratified formats to formal glTF extensions. Support in REST.
Silicon vendors begin hardware acceleration