OMI glTF Extensions Brainstorm #7

robertlong started this conversation in **General**

robertlong on Aug 12, 2021 Maintainer edited

Submit your ideas for OMI glTF extensions. These can be support for features from game engines such as physics or additional materials. They could be extensions specific to avatars or equipable items. Anything you'd like to see support for in the glTF ecosystem that you think the OMI community should push forward.

These don't have to be formal proposals. That will be a step later on. The goal of this conversation is to identify extensions that we all want.

Be sure to upvote if you'd like to see an extension developed further and comment if you'd be willing to help champion it.

↑ 2

12 comments · 1 reply

Oldest Newest Top

robertlong on Aug 18, 2021 Maintainer Author

Scene Backgrounds / Skyboxes

I think we should support scene backgrounds / skyboxes with equirectangular cubemaps. People already accomplish this by including a sky dome and the associated geometry in their scene. However, in practice, this can result in incorrect draw order, clipping in large scenes, complex geometry being used for the skydome, and more. It'd be much better if we could just specify a texture to use as the scene background.

↑ 3 0 replies

robertlong on Aug 18, 2021 Maintainer Author

Physics

This could be a hard one to specify since physics engines can represent things in pretty different ways, however if we could find a way to specify colliders, it'd mean that character controllers would be able to accurately collide with a scene. I'd ideally like to static, dynamic, and kinematic colliders. Such that we could have collision meshes for the scene, moving platforms, and dynamically movable objects.

↑ 2 0 replies

robertlong on Aug 18, 2021 Maintainer Author

Shadows

glTF has support for defining lights, but not shadows. We should add flags for whether or not an object casts or receives shadows. We also should add a flag for whether a light casts shadows. We could also add properties for lights such as shadow bias, normal bias, radius, resolution, etc.

↑ 2 0 replies

robertlong on Aug 18, 2021 Maintainer Author

Visibility

This is a simple one. You may create an object in a glTF scene and not want to initially show it. A visibility extension would have a single boolean telling the runtime to hide the object.

↑ 3 0 replies

robertlong on Aug 18, 2021 Maintainer Author

Animation

Animation graphs and timelines can quickly get complex. It'd be nice at least to be able to specify that an animation should auto play and loop on load.

↑ 3 0 replies

robertlong on Aug 18, 2021 Maintainer Author

Light Maps

The Mozilla Hubs team put effort into making this easy already. Maybe we can work with them to formalize their spec?

↑ 1 0 replies

robertlong on Aug 18, 2021 Maintainer Author

Referenced glTF

It'd be great to be able to compose a glTF that references external glTF scenes. Perhaps we could design the spec in such a way where you can pull out pieces of a referenced glTF and reuse them multiple times. Like an asset collection and being able to reuse an asset in a scene multiple times. Another usecase would be to dynamically generate a referenced glTF asset at request time.

↑ 2 1 reply

mikeskydev on Aug 18, 2021

I think this will be really important for creating larger scenes from pre-existing assets. I'd love to use an extension like this to represent gallery spaces.

robertlong on Aug 18, 2021 Maintainer Author

Video Textures

You should be able to use a video file or stream as a texture.

↑ 2 0 replies

robertlong on Aug 18, 2021 Maintainer Author

Triggers

This would be an extension for providing basic behaviors to glTF assets. Animation keyframes, collision events, and runtime events could be observed, optionally passed through operators, and then passed to targets. This could start with basic operations on the glTF asset and extensions, but expand much further through the use of external events.

Observers

- Animation Observer
- Collision Observer
- Event Observer

Operators

- Boolean Operators
- State Operators
- Arithmetic Operators

Targets

- Animation Target
- Audio Target
- Transform Target
- Visibility Target
- Event Target
- Material Property Target

2

0 replies

robertlong on Aug 18, 2021 Maintainer Author

glTF IDs

The glTF spec uses indices as identifiers. Sometimes these indices change between edits to the file. Names are not guaranteed to exist or be unique, so you can't reliably use them as identifiers. This is fine when you know the exact content of your scene because you can refer to a node by index or name. However, it'd be nice to have an identifier that is specified as a unique identifier that is stable between edits. This extension would focus only on adding formally specified unique string identifiers.

1

0 replies

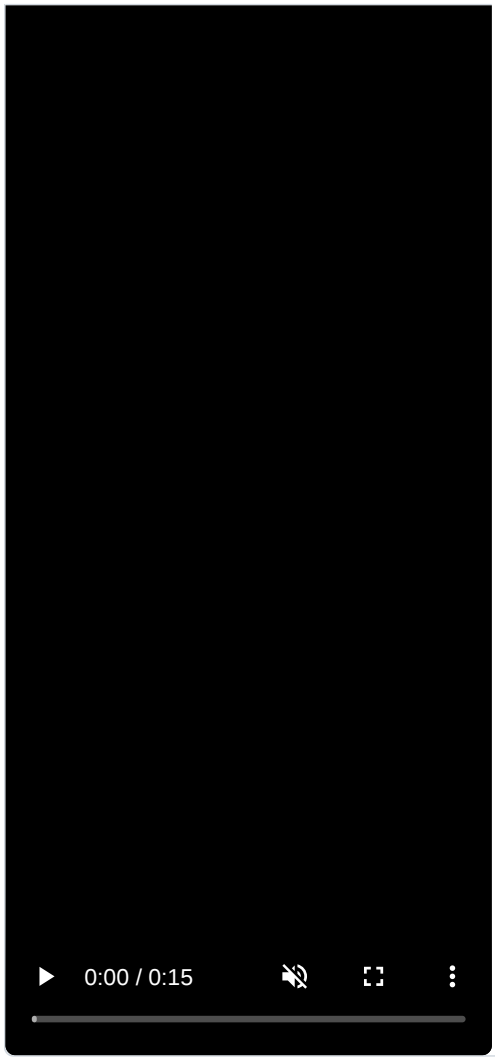
adampaigge on Aug 18, 2021 edited

glTF NFTs

Not sure about how this would be implemented but, a metadata tag to track token contracts (currently working on BSV, but also tracking IPFS hashes) so that files can have referenced ownership based on some key or token in the tag would be useful for the crypto art community to start standardising the way things can be pulled into experiences.

Would also allow experiences to reference a NFT token or contract address to actually fetch the file from IPFS or from a chain directly in the case of bitcoinfiles. (see link for more information on how this implemented in practice)

8b0976a20f21e26b126612b0cc550f3cb665143f5dc5409bc85e9a18d21bf6ff.mp4



1

0 replies

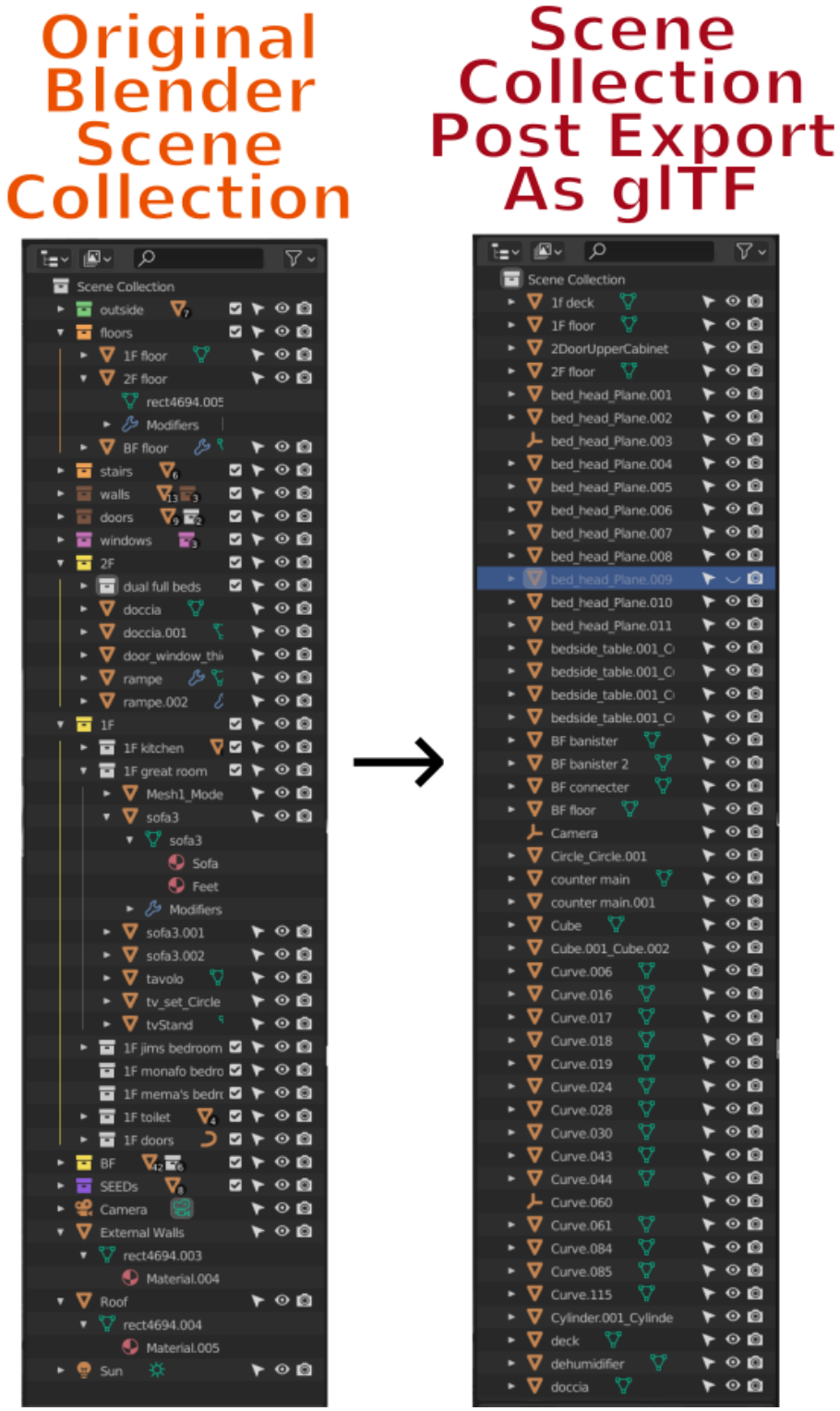
humbletim on Aug 25, 2021 Maintainer

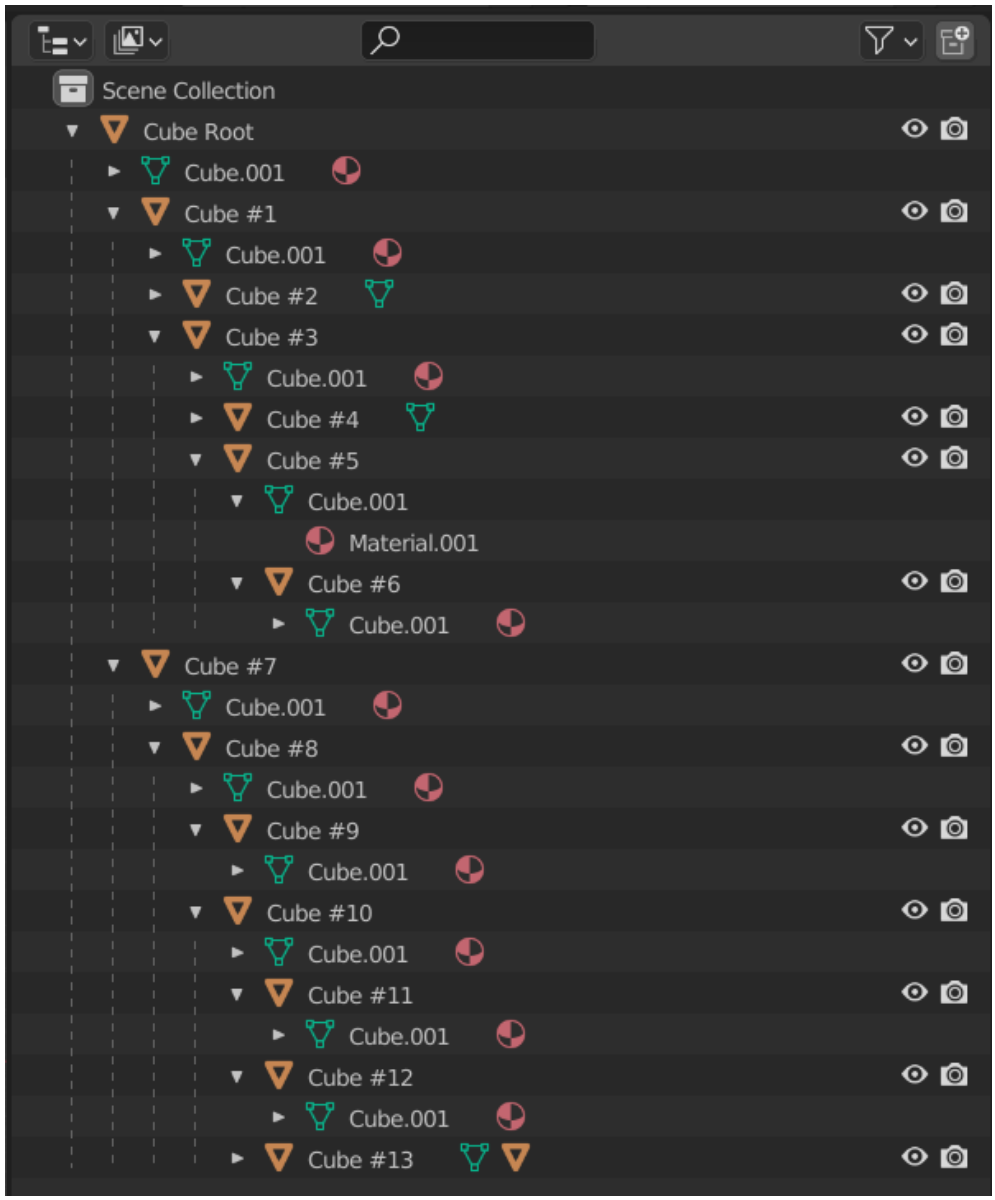
edited

Preserve Scene Hierarchy

on 2021-08-24, community member **@dysbulic** suggested this on the omi-gltf-extensions discord :

> I have long been frustrated by the fact that when a file is exported from Blender to glTF, it flattens the collections hierarchy to a single level.
>
> I have created a simple example with nested objects: https://github.com/dysbulic/stay/blob/react/public/cubes.gltf. The extension field is valid on any glTF object. ¿Can we not use that to mark that nodes are Collections rather than Objects and create the appropriate type when deserializing?
>
> There is a Gitcoin Grants Round coming up Sept. 8th. I'd very much like to identify a GitHub issue to try and rally support for it in the funding process.
>
> This, KhronosGroup/glTF-Blender-IO#1378, is my best guess so far. I've tried inquiring if it will fix the flattening issue, but, frustratingly, gotten no response.
>
> My ultimate interest is a system for medium (weeks long) term rentals where 3D maps partitioned using trees are the sources of truth as to what's available.


Original Blender Scene Collection
Scene Collection Post Export As glTF
Scene Collection
outside
floors
1F floor
2F floor
rect4694.005
Modifiers
BF floor
stairs
walls
doors
windows
2F
dual full beds
doccia
doccia.001
door_window_thi
rampe
rampe.002
1F
1F kitchen
1F great room
Mesh1_Mode
sofa3
sofa3
Sofa
Feet
Modifiers
sofa3.001
sofa3.002
tavolo
tv_set_Circle
tvStand
1F jims bedroom
1F monafo bedro
1F mema's bedr
1F toilet
1F doors
BF
SEEDs
Camera
External Walls
rect4694.003
Material.004
Roof
rect4694.004
Material.005
Sun
Scene Collection
1f deck
1F floor
2DoorUpperCabinet
2F floor
bed_head_Plane.001
bed_head_Plane.002
bed_head_Plane.003
bed_head_Plane.004
bed_head_Plane.005
bed_head_Plane.006
bed_head_Plane.007
bed_head_Plane.008
bed_head_Plane.009
bed_head_Plane.010
bed_head_Plane.011
bedside_table.001_C
bedside_table.001_C
bedside_table.001_C
bedside_table.001_C
BF banister
BF banister 2
BF connecter
BF floor
Camera
Circle_Circle.001
counter main
counter main.001
Cube
Cube.001_Cube.002
Curve.006
Curve.016
Curve.017
Curve.018
Curve.019
Curve.024
Curve.028
Curve.030
Curve.043
Curve.044
Curve.060
Curve.061
Curve.084
Curve.085
Curve.115
Cylinder.001_Cylinde
deck
dehumidifier
doccia



1

0 replies

Category

General

Labels

None yet

4 participants