12/07/22 Weekly Meeting 85 Notes #323

mrmetaverse started this conversation in **General**

mrmetaverse on Dec 7, 2022 Maintainer

Weekly Meeting 85 is happening on Wednesday December 7, 2022 from 9:00 AM PST(17:00 UTC) - 10:00 AM PST (18:00 UTC)

You can add it to your calendar here.

We'll be meeting again in the omi-weekly-meeting channel of the AngellXR/OMI Discord.
We regularly use the # omi-meeting-chat channel for text chat.

Who we are and why we're here:

The Open Metaverse Interoperability Group (OMI) is focused on bridging virtual worlds by designing and promoting protocols for identity, social graphs, inventory, and more. Our members include businesses and individuals working towards this common goal.

If you're an observer or here for the first time, this is a meeting where people who work on the OMI group initiatives discuss updates and contribution opportunities for current projects and goals. This is your chance to jump in and help!

We welcome you to introduce yourself in #omi-meeting-chat as we dive into the agenda items. There will be an open floor in the last 15 minutes of the meeting (agenda permitting) where we can discuss any items you feel are relevant to the current or near future OMI goals.

Agenda

- Volunteer to take notes? Volunteer to record?
- Welcome newcomers

Working Group Updates:

- MSF delegates (setting up the repo)
- UX Research
- Scripting group
  - OMI developer hour*
- glTF Extensions Group
  - Progress on OMI_ref and authoring extensions

- Media Group
- M3 avatar interop
- OMI Website update
- any others?

Discussions:

- DEMO! :)
- OMC updates next week
- This is a time to build. What can we build?
  - Adding OMI to the Metaverse Street Jin demoed
  - M3 recently started a shop for wearables. This is a great example of things we could be doing and participating in.

Next week:

Demoing the OMI collider and physics body (in Godot!)

Future demo themes:

- `OMI-audio-emitter` exploration and demo
- `OMI-collider` exploration and demo (December 7!)
- `IPSME` and Me
- `Generative and AI tooling`
- Any other suggestions?

↑ 1

1 comment · 3 replies

Oldest | Newest | Top

mrmetaverse on Dec 7, 2022 Maintainer Author

Demo time!

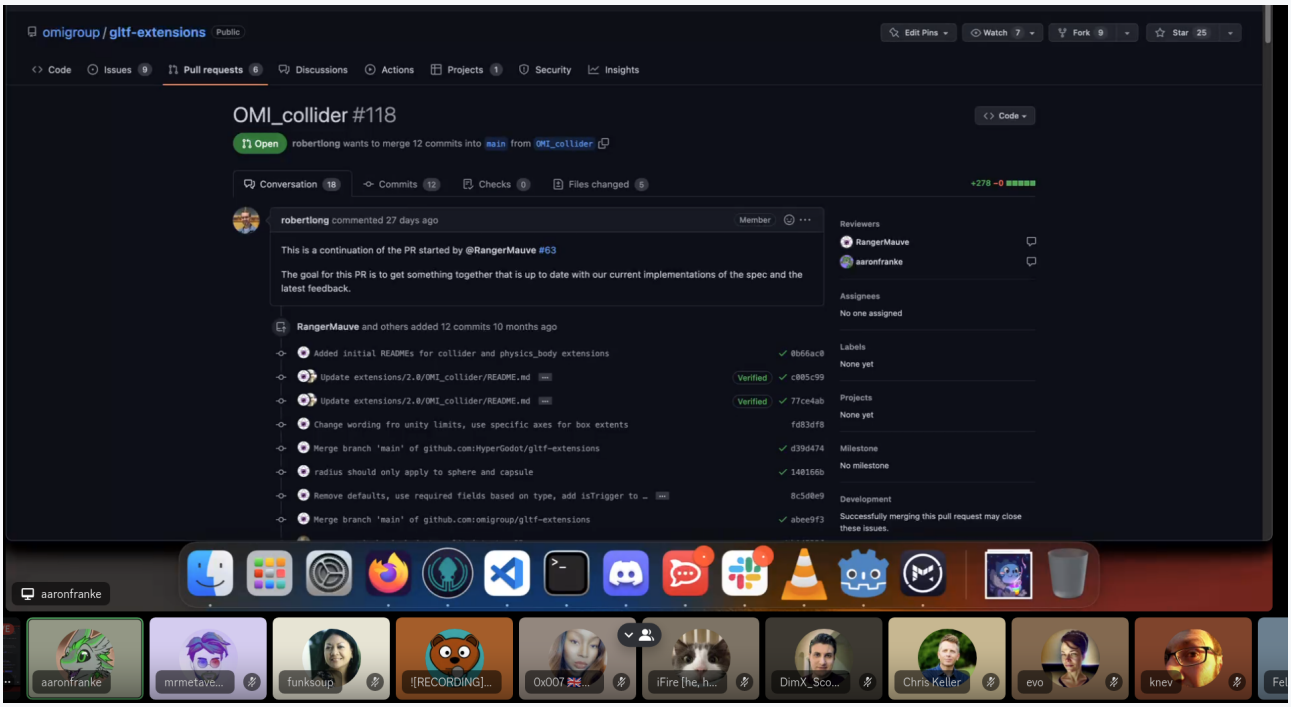
↑ 1

3 replies

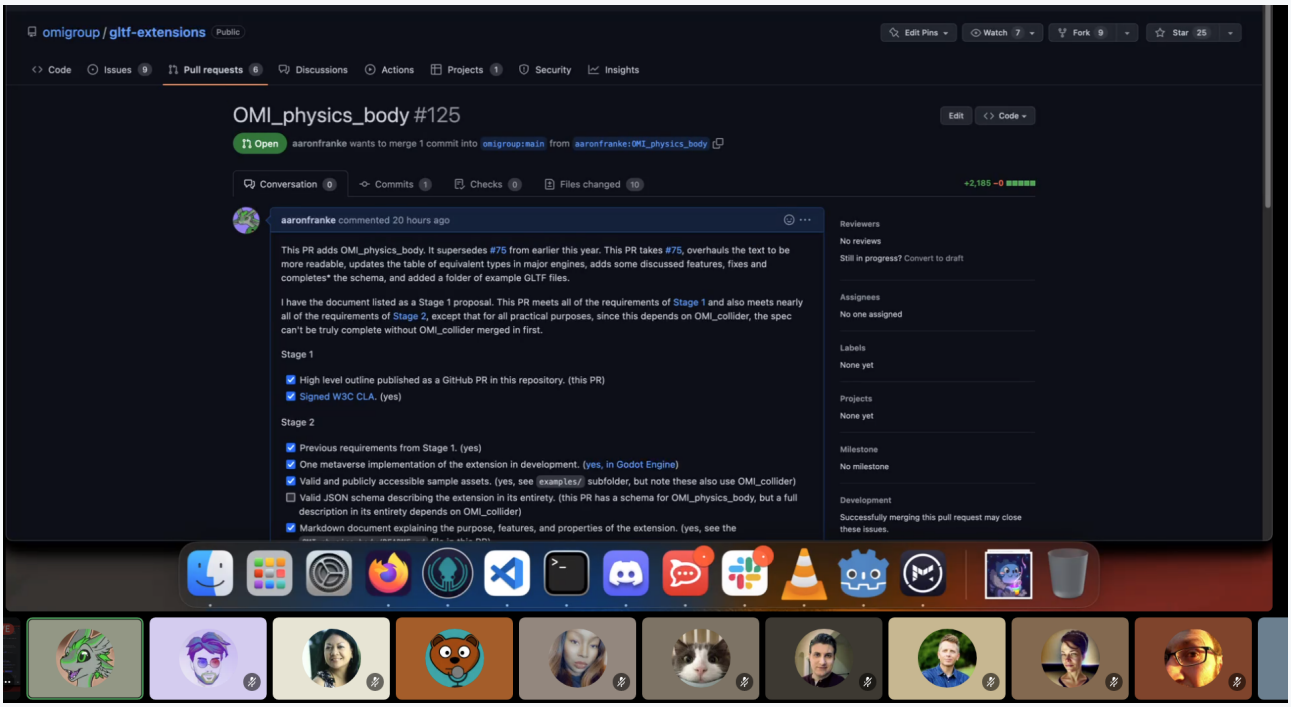
mrmetaverse on Dec 7, 2022 Maintainer Author edited

Two extensions here being designed by OMI meant to work together.

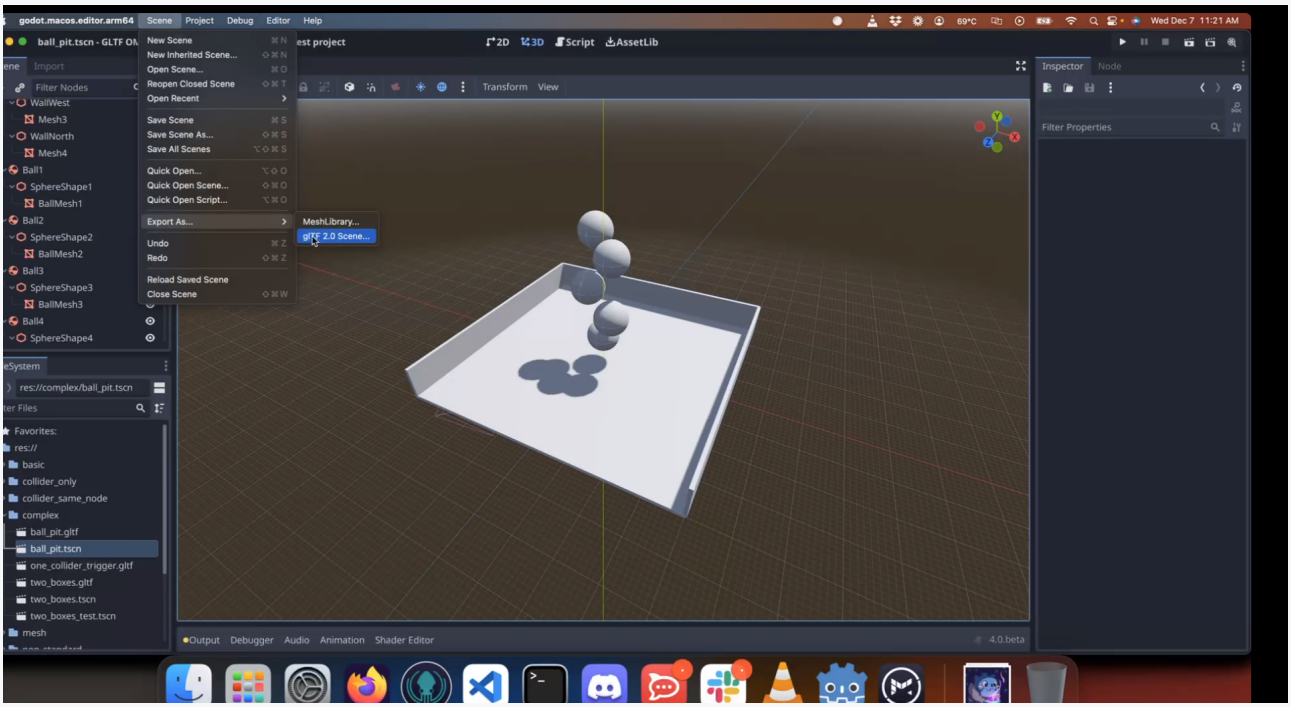
omi-collider



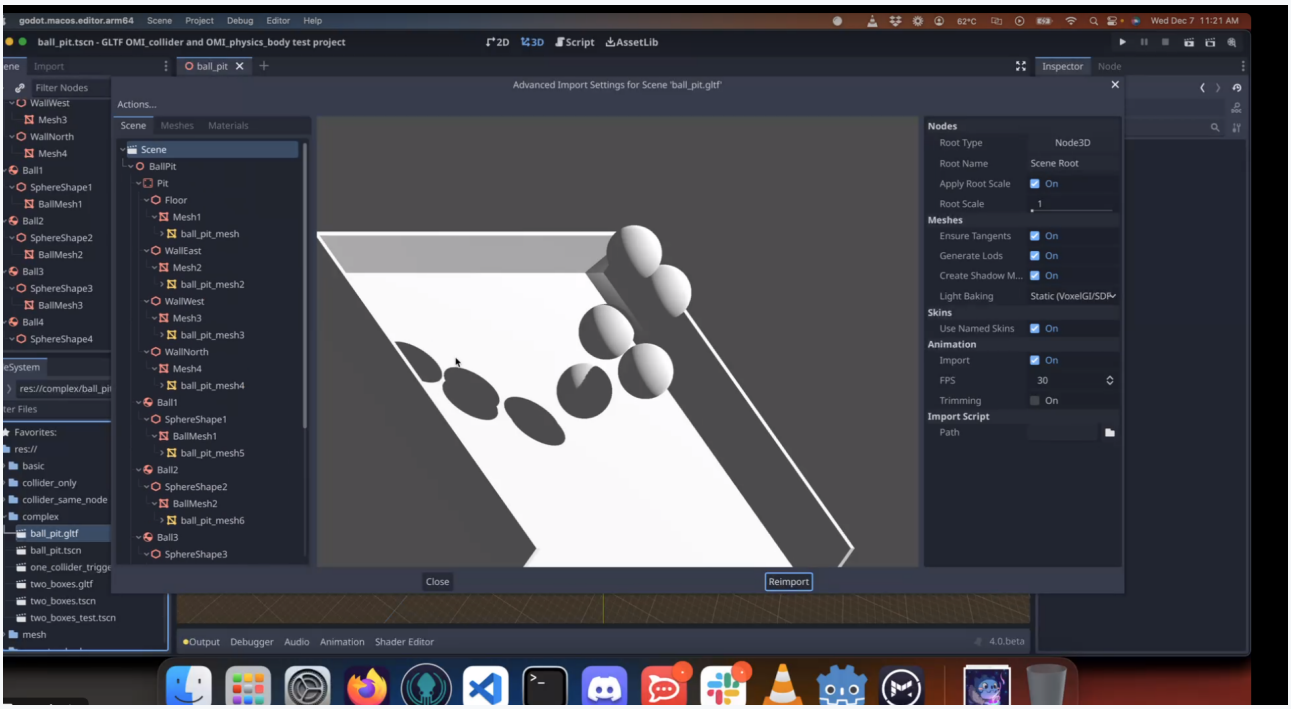
and omi physics body



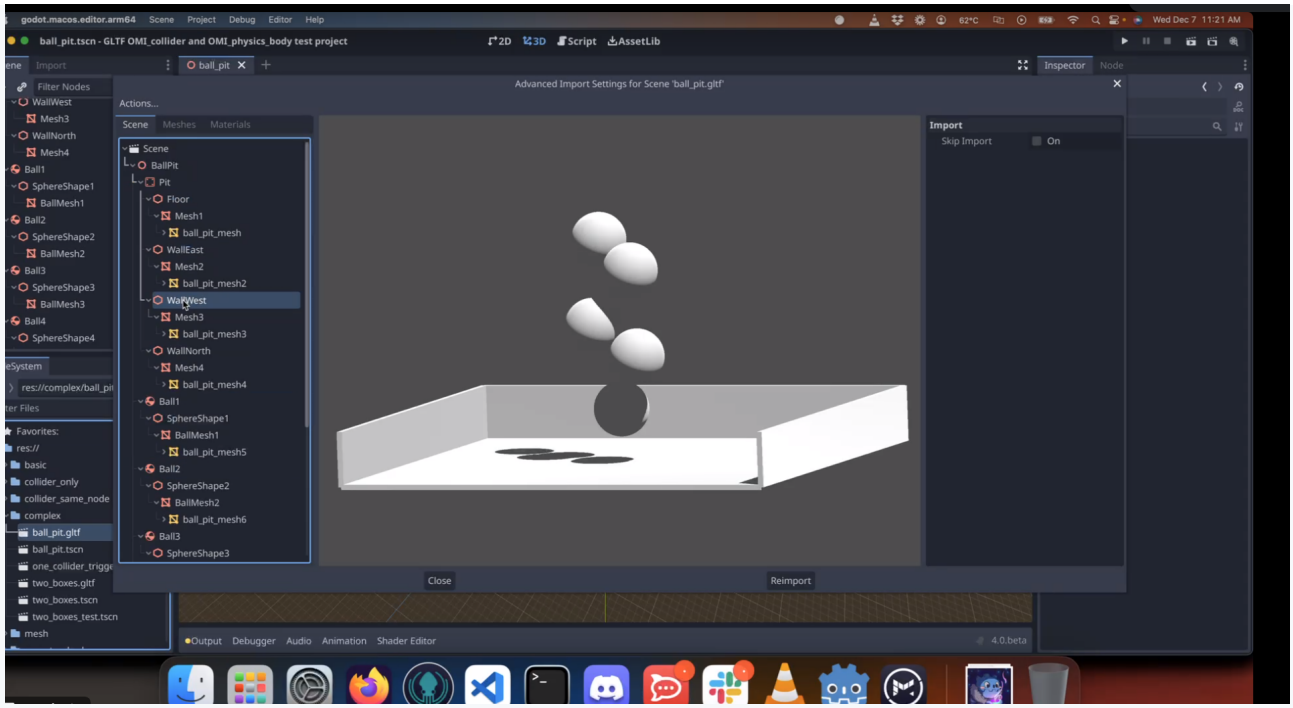
In godot we can export to gltf



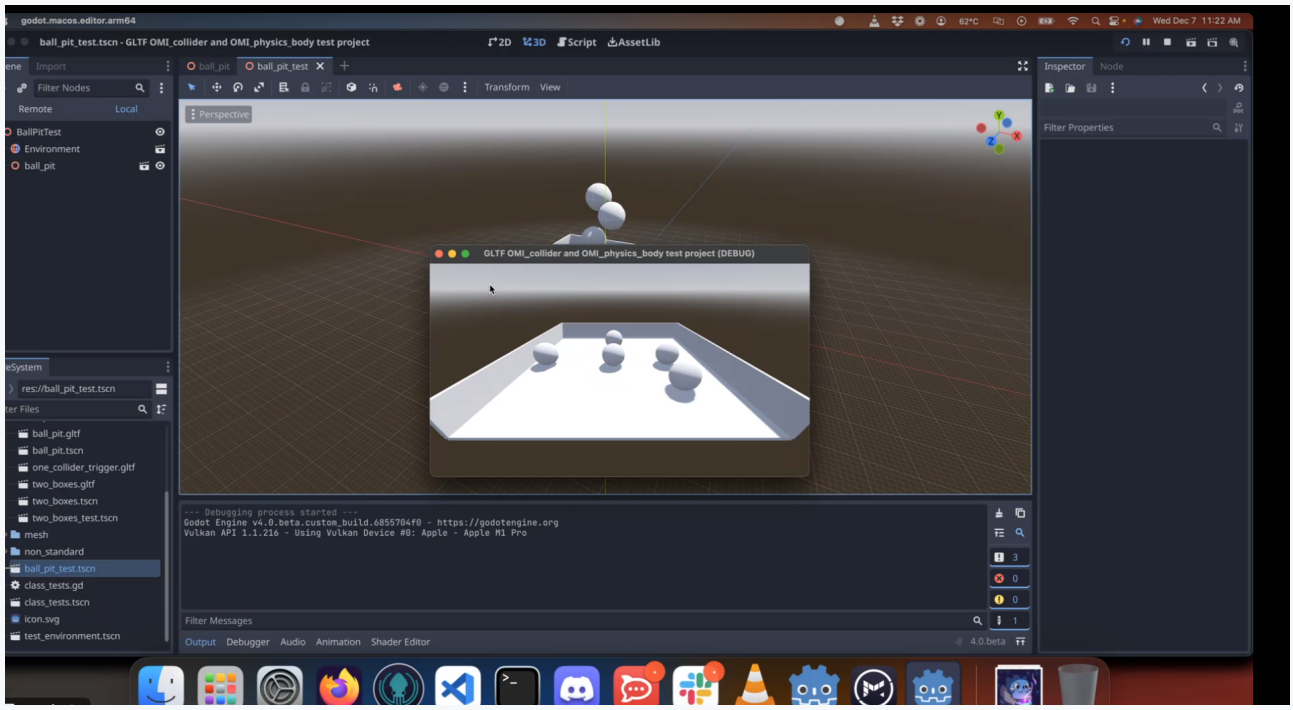
This creates this new gLTF file



Includes physics bodies and shapes imported as Godot nodes

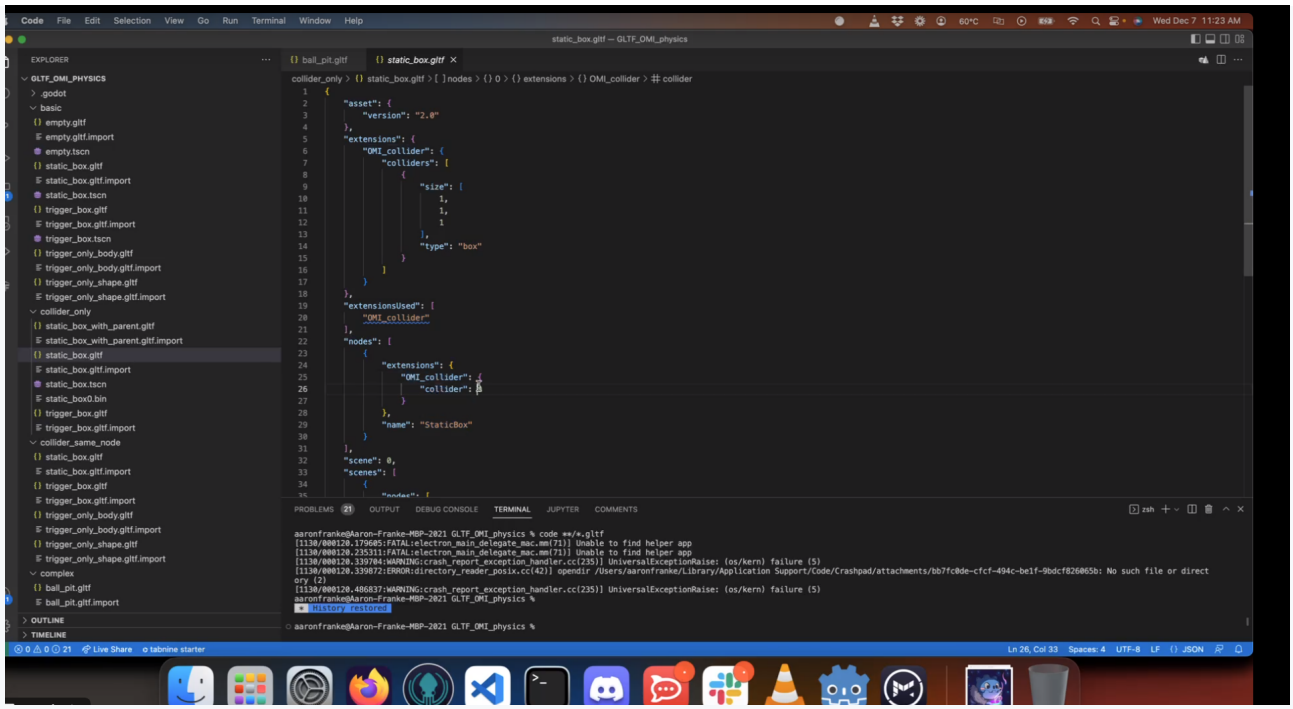


We can run this and the balls collide exactly as they should and we can bring into *any* game engine we want!



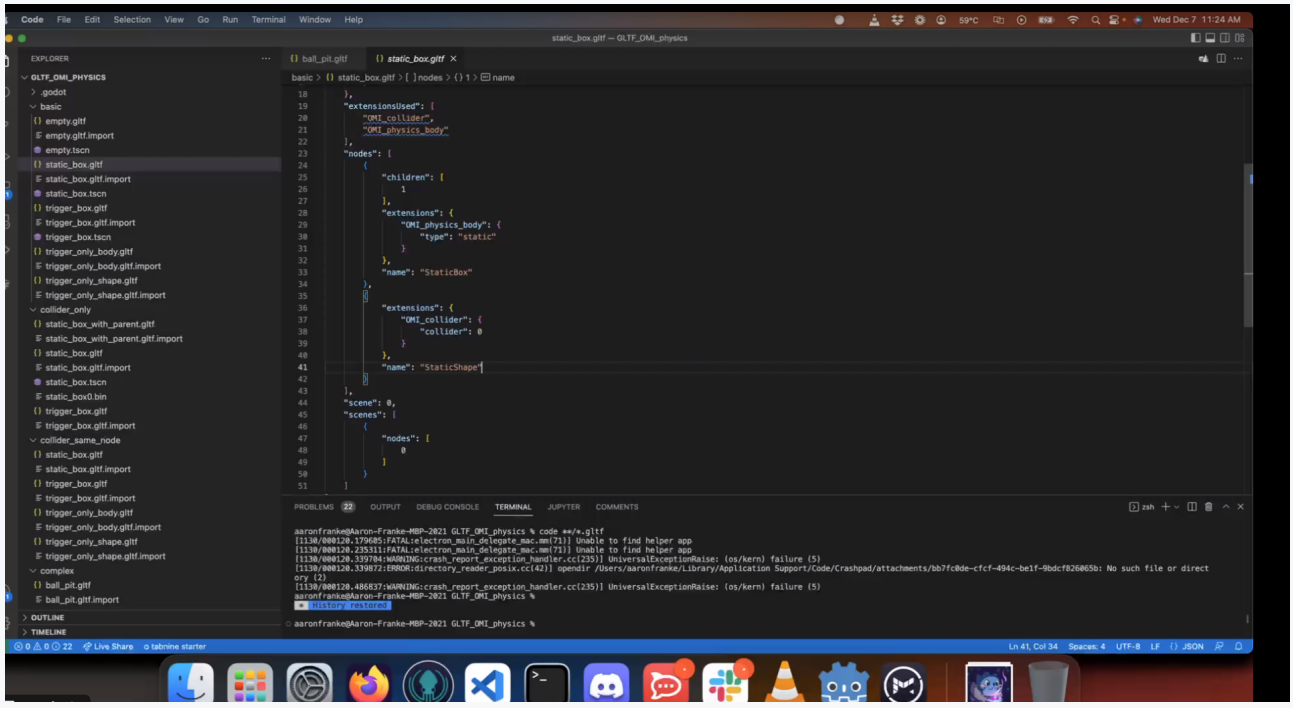
Other examples:

We want to handle the case where there is only an OMI_Collider. That is a simpler system that will work individually. This also aligns with one of our core values for modularity and severability.

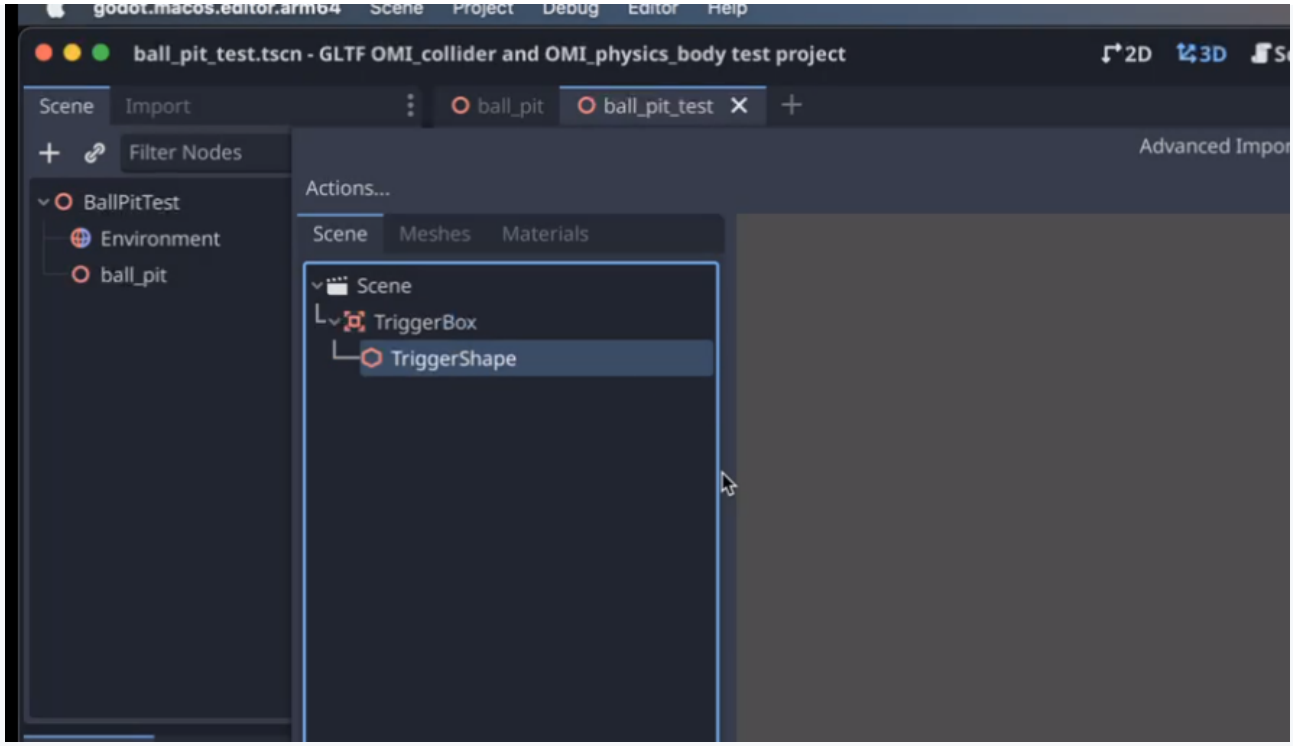


We strive to make this as simple as it can be, and then it can be used in any other engine.

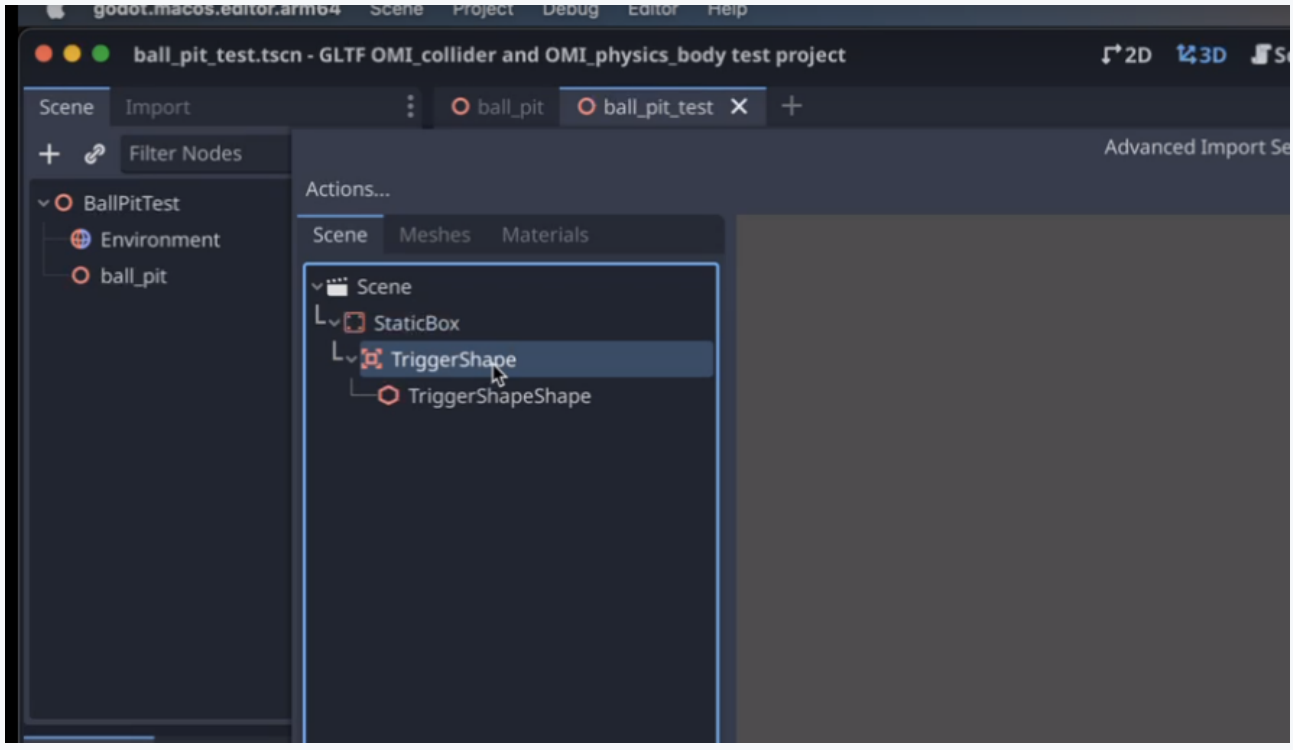
Another example has both the collider and physics body



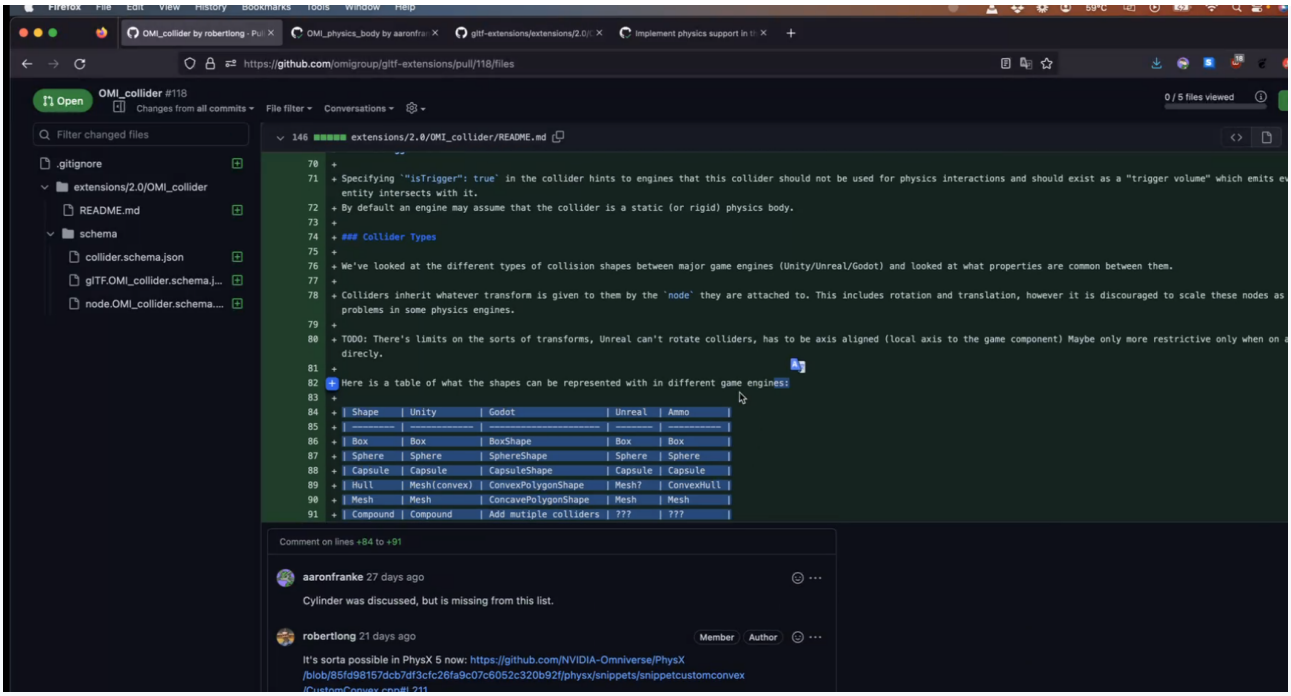
In unity you only need the triggershape. But we want to make it more portable between Unity and other engines as well. Wheras, in godot we require a body for all shapes.



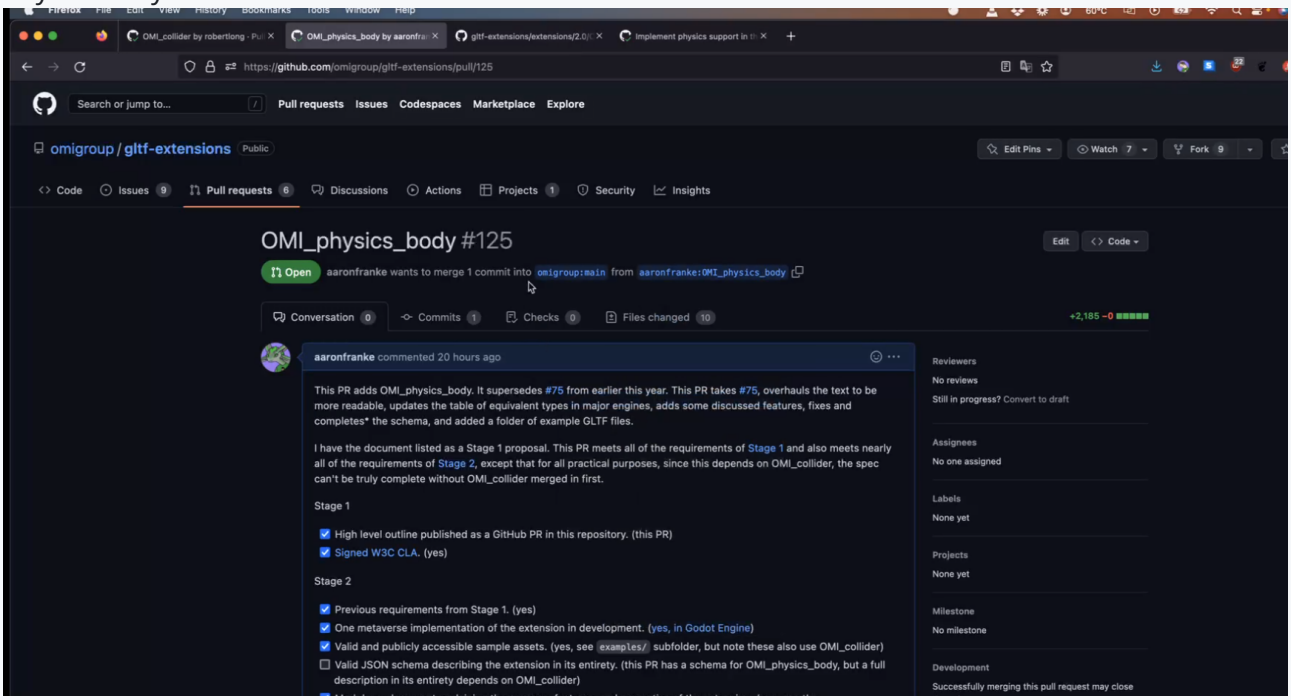
Setup so that godot creates a child body so the shape behaves as expected.



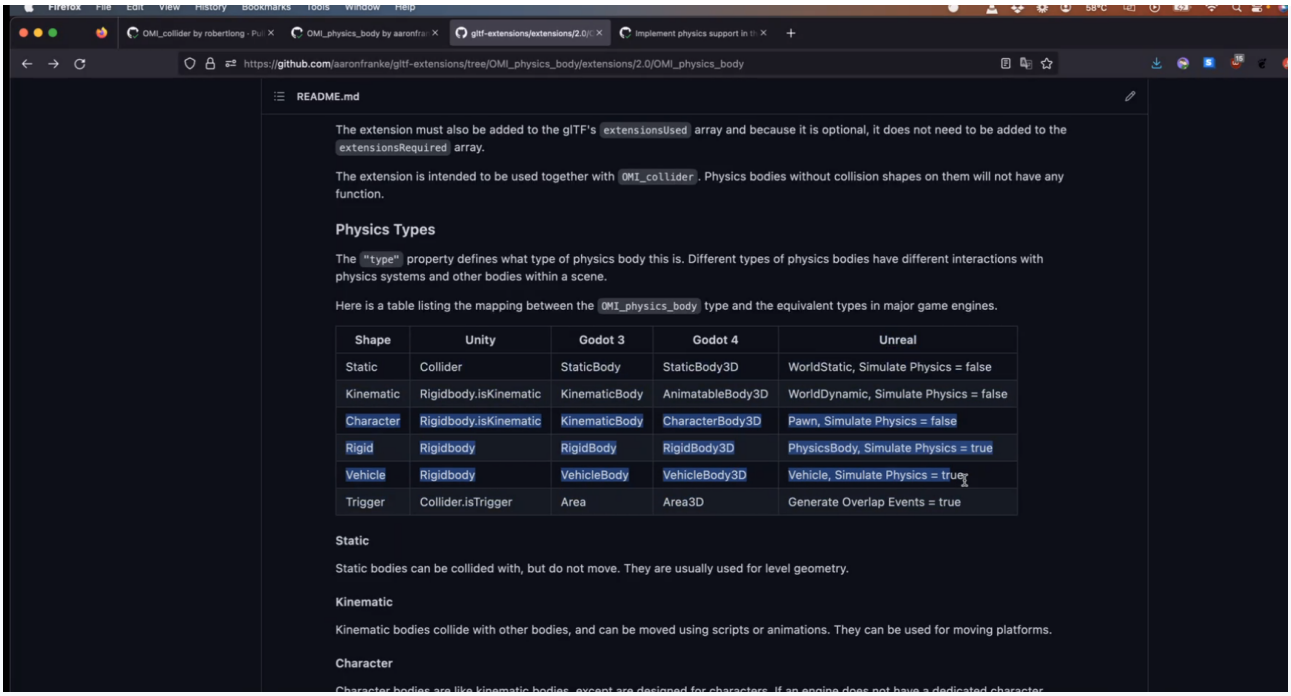
We have a PR submitted for OMI_Collider that calls out various examples and use-cases:



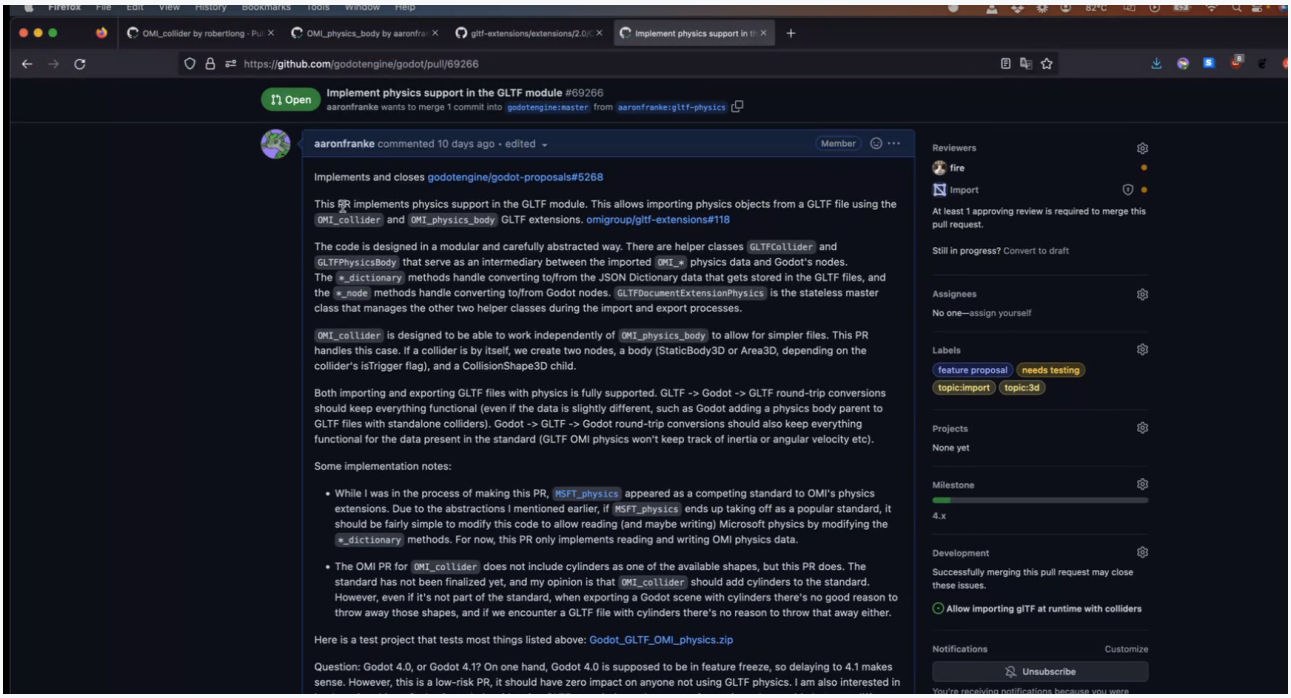
Physics Body has a new PR as well. Re-worked



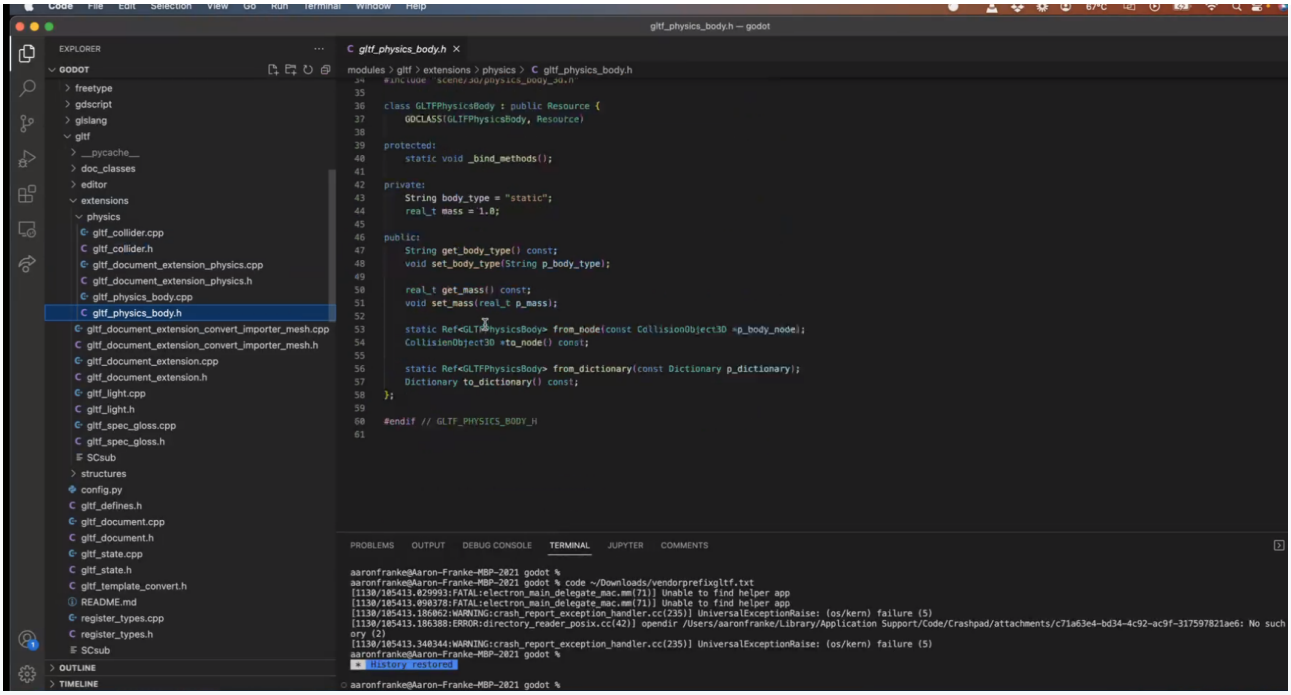
Each of these types has meaning in multiple game engines

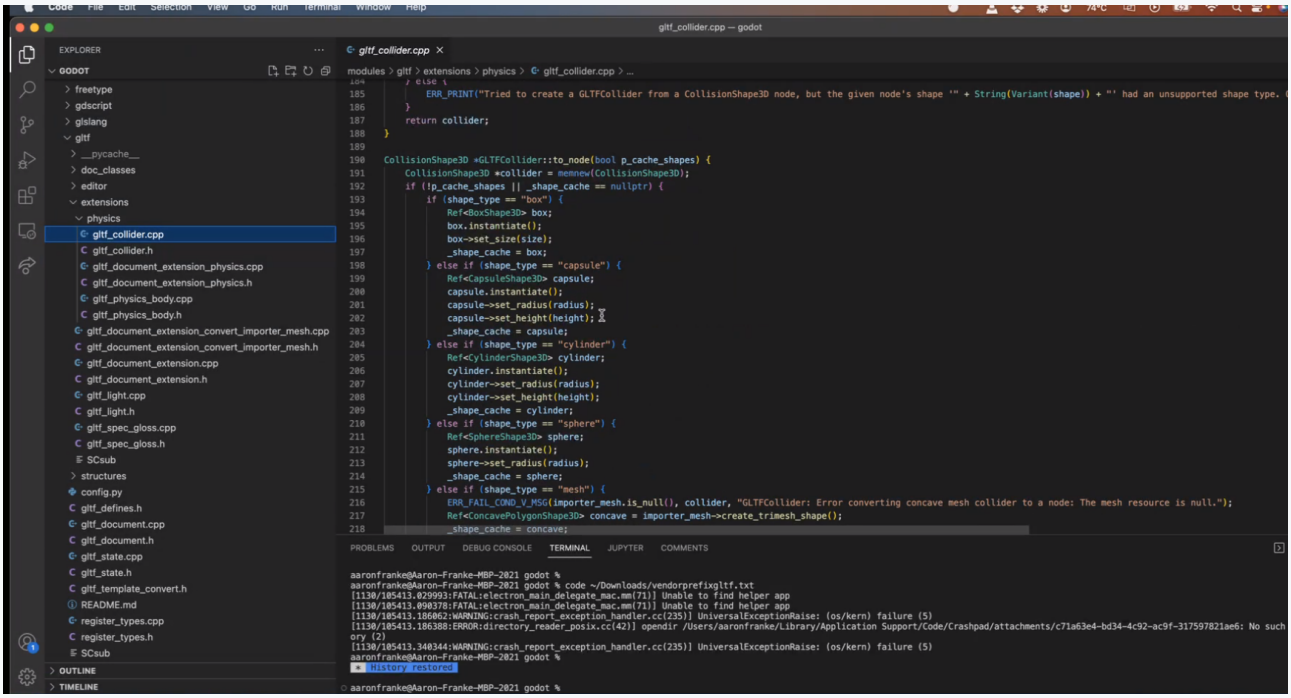


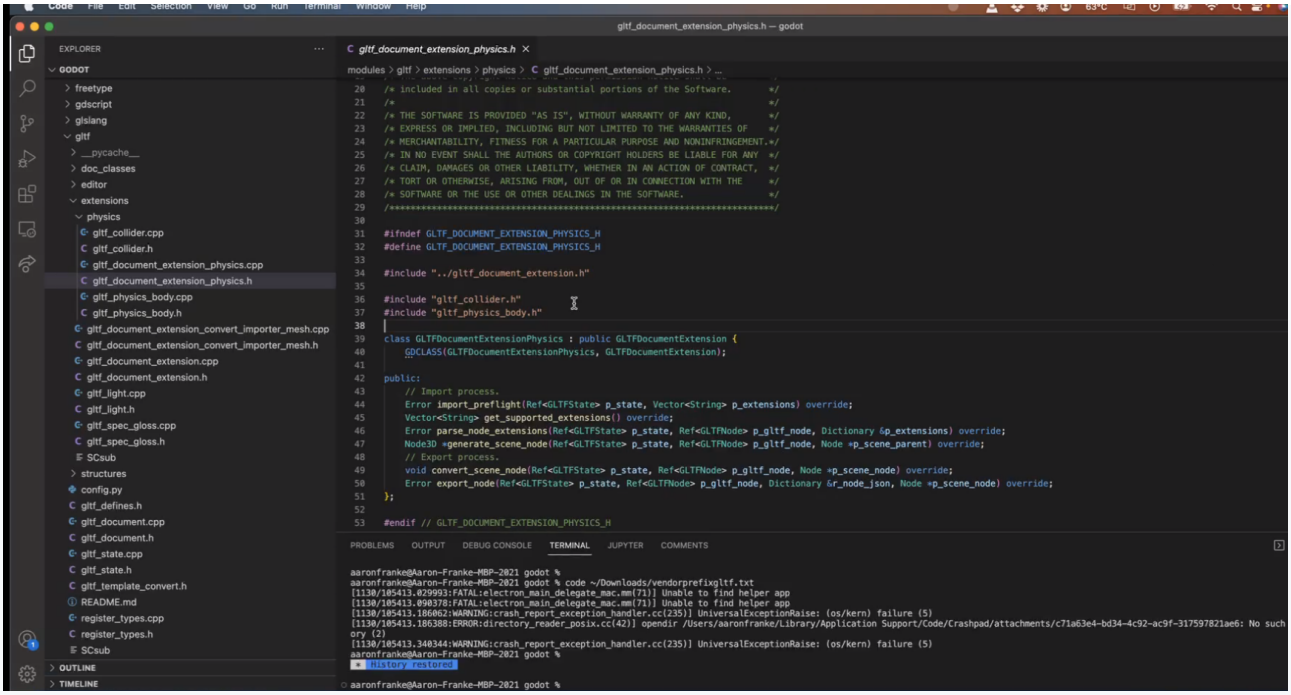
We have this PR to Godot engine discussing implementation



Inside of godots code we have these little methods for converting as well:

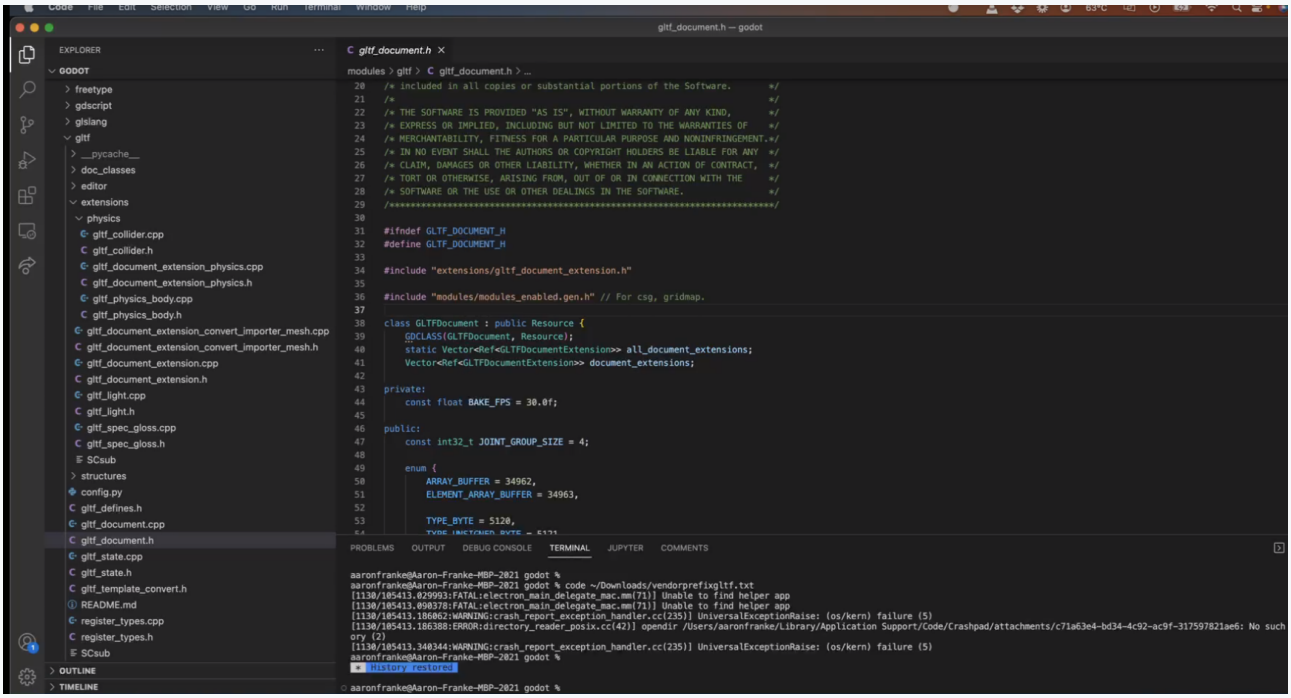

gltf_physics_body.h — godot
EXPLORER
GODOT
modules > gltf > extensions > physics > gltf_physics_body.h
class GLTFPhysicsBody : public Resource {
GDCLASS(GLTFPhysicsBody, Resource)
protected:
static void _bind_methods();
private:
String body_type = "static";
real_t mass = 1.0;
public:
String get_body_type() const;
void set_body_type(String p_body_type);
real_t get_mass() const;
void set_mass(real_t p_mass);
static Ref<GLTFPhysicsBody> from_node(const CollisionObject3D *p_body_node);
CollisionObject3D *to_node() const;
static Ref<GLTFPhysicsBody> from_dictionary(const Dictionary p_dictionary);
Dictionary to_dictionary() const;
};
#endif // GLTF_PHYSICS_BODY_H
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER COMMENTS
aaronfranke@Aaron-Franke-MBP-2021 godot %
aaronfranke@Aaron-Franke-MBP-2021 godot % code ~/Downloads/vendorprefixgltf.txt
[1130/105413.029993:FATAL:electron_main_delegate_mac.mm(71)] Unable to find helper app
[1130/105413.090378:FATAL:electron_main_delegate_mac.mm(71)] Unable to find helper app
[1130/105413.186062:WARNING:crash_report_exception_handler.cc(235)] UniversalExceptionRaise: (os/kern) failure (5)
[1130/105413.186388:ERROR:directory_reader_posix.cc(42)] opendir /Users/aaronfranke/Library/Application Support/Code/Crashpad/attachments/c71a63e4-bd34-4c92-ac9f-317597821ae6: No such
ory (2)
[1130/105413.340344:WARNING:crash_report_exception_handler.cc(235)] UniversalExceptionRaise: (os/kern) failure (5)
aaronfranke@Aaron-Franke-MBP-2021 godot %
History restored
aaronfranke@Aaron-Franke-MBP-2021 godot %


gltf_collider.cpp — godot
EXPLORER
GODOT
modules > gltf > extensions > physics > gltf_collider.cpp > ...
ERR_PRINT("Tried to create a GLTFCollider from a CollisionShape3D node, but the given node's shape '" + String(Variant(shape)) + "' had an unsupported shape type.
}
return collider;
}
CollisionShape3D *GLTFCollider::to_node(bool p_cache_shapes) {
CollisionShape3D *collider = memnew(CollisionShape3D);
if (!p_cache_shapes || _shape_cache == nullptr) {
if (shape_type == "box") {
Ref<BoxShape3D> box;
box.instantiate();
box->set_size(size);
_shape_cache = box;
} else if (shape_type == "capsule") {
Ref<CapsuleShape3D> capsule;
capsule.instantiate();
capsule->set_radius(radius);
capsule->set_height(height);
_shape_cache = capsule;
} else if (shape_type == "cylinder") {
Ref<CylinderShape3D> cylinder;
cylinder.instantiate();
cylinder->set_radius(radius);
cylinder->set_height(height);
_shape_cache = cylinder;
} else if (shape_type == "sphere") {
Ref<SphereShape3D> sphere;
sphere.instantiate();
sphere->set_radius(radius);
_shape_cache = sphere;
} else if (shape_type == "mesh") {
ERR_FAIL_COND_V_MSG(importer_mesh.is_null(), collider, "GLTFCollider: Error converting concave mesh collider to a node: The mesh resource is null.");
Ref<ConcavePolygonShape3D> concave = importer_mesh->create_trimesh_shape();
_shape_cache = concave;
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER COMMENTS
aaronfranke@Aaron-Franke-MBP-2021 godot %
aaronfranke@Aaron-Franke-MBP-2021 godot % code ~/Downloads/vendorprefixgltf.txt
[1130/105413.029993:FATAL:electron_main_delegate_mac.mm(71)] Unable to find helper app
[1130/105413.090378:FATAL:electron_main_delegate_mac.mm(71)] Unable to find helper app
[1130/105413.186062:WARNING:crash_report_exception_handler.cc(235)] UniversalExceptionRaise: (os/kern) failure (5)
[1130/105413.186388:ERROR:directory_reader_posix.cc(42)] opendir /Users/aaronfranke/Library/Application Support/Code/Crashpad/attachments/c71a63e4-bd34-4c92-ac9f-317597821ae6: No such
ory (2)
[1130/105413.340344:WARNING:crash_report_exception_handler.cc(235)] UniversalExceptionRaise: (os/kern) failure (5)
aaronfranke@Aaron-Franke-MBP-2021 godot %
History restored
aaronfranke@Aaron-Franke-MBP-2021 godot %

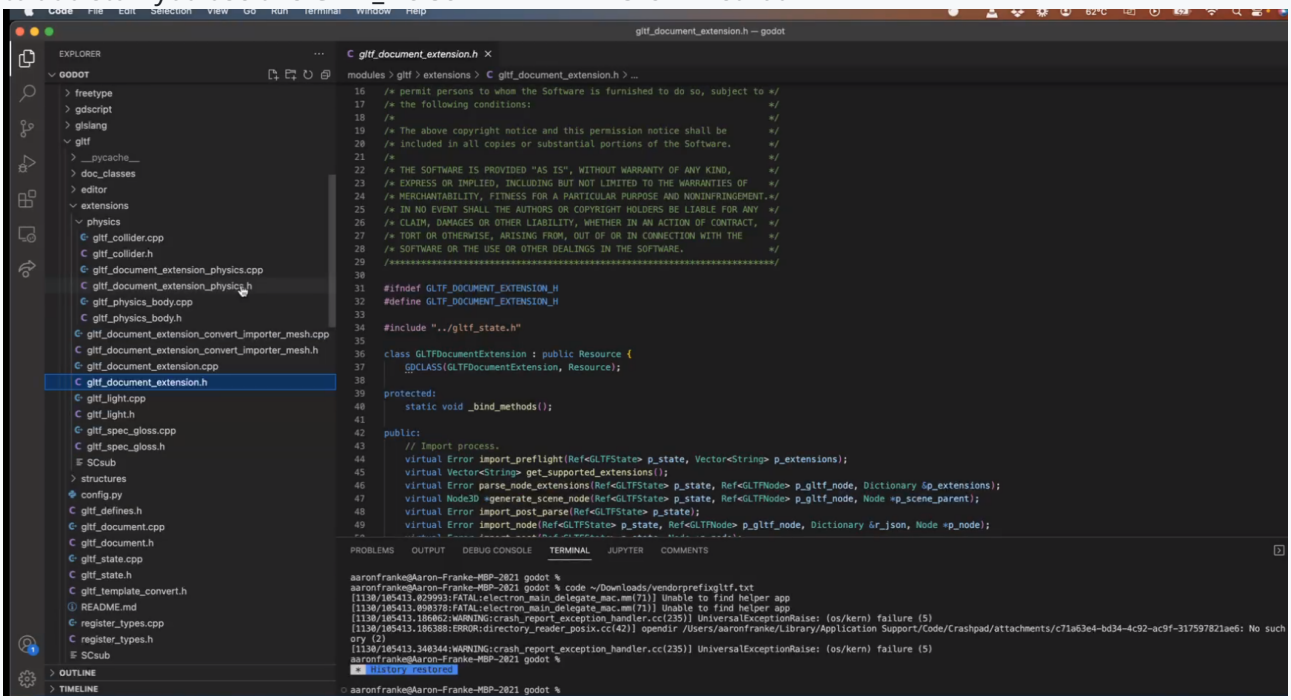


How is a gLTF document extension different from an importer?

When godot imports it goes through this thing called glTF document



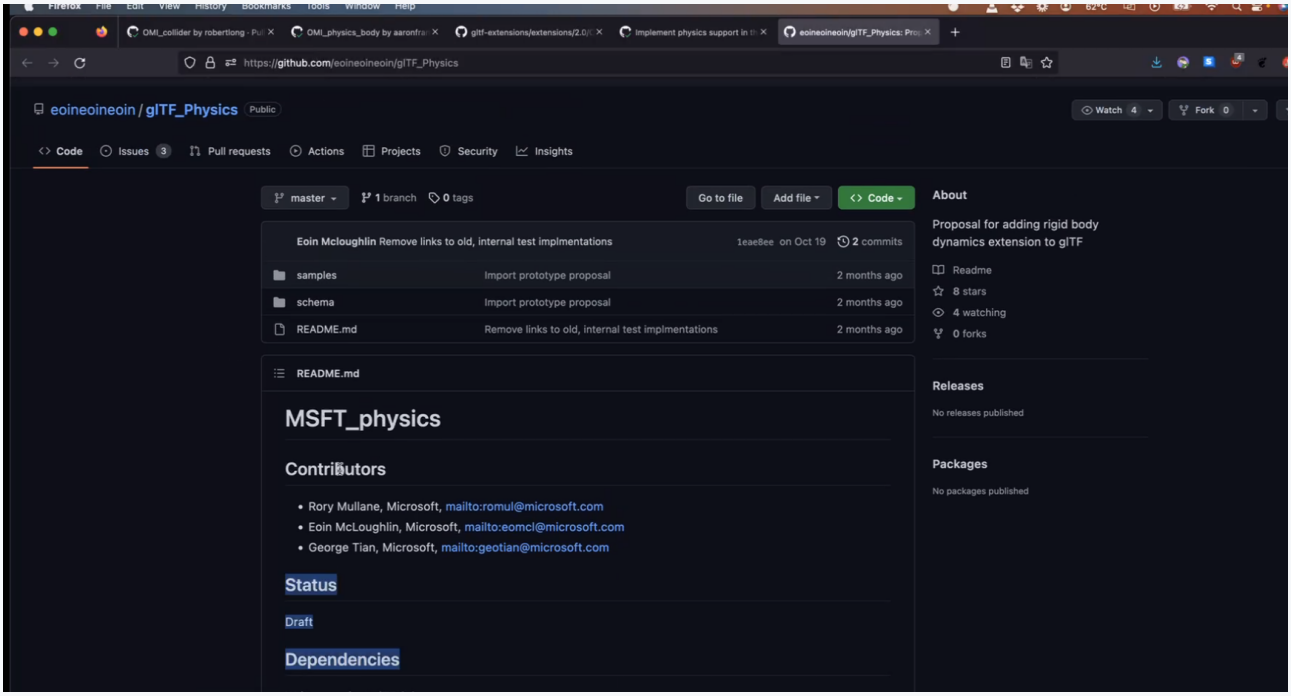
to add stuff you use this GLTF_DOCUMENT EXTENSION method



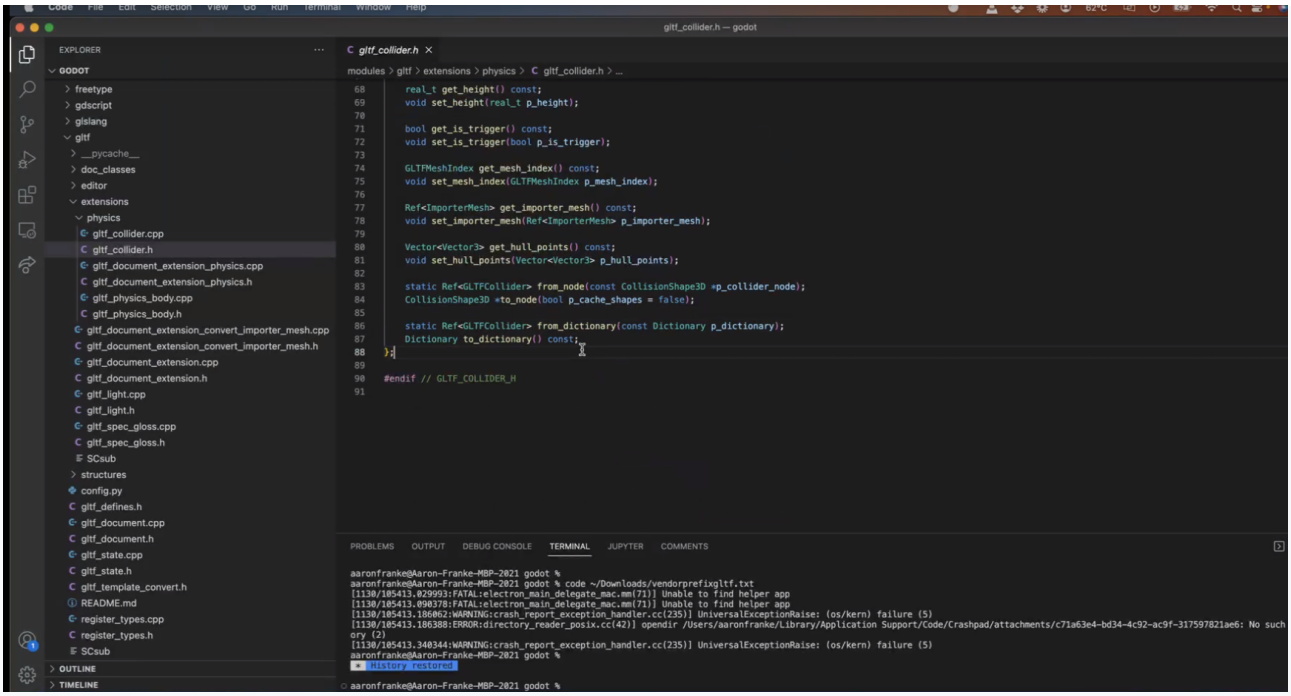
If you want to implement it into godot, you can literally just write it and drop it into godot without recompiling the engine every time.

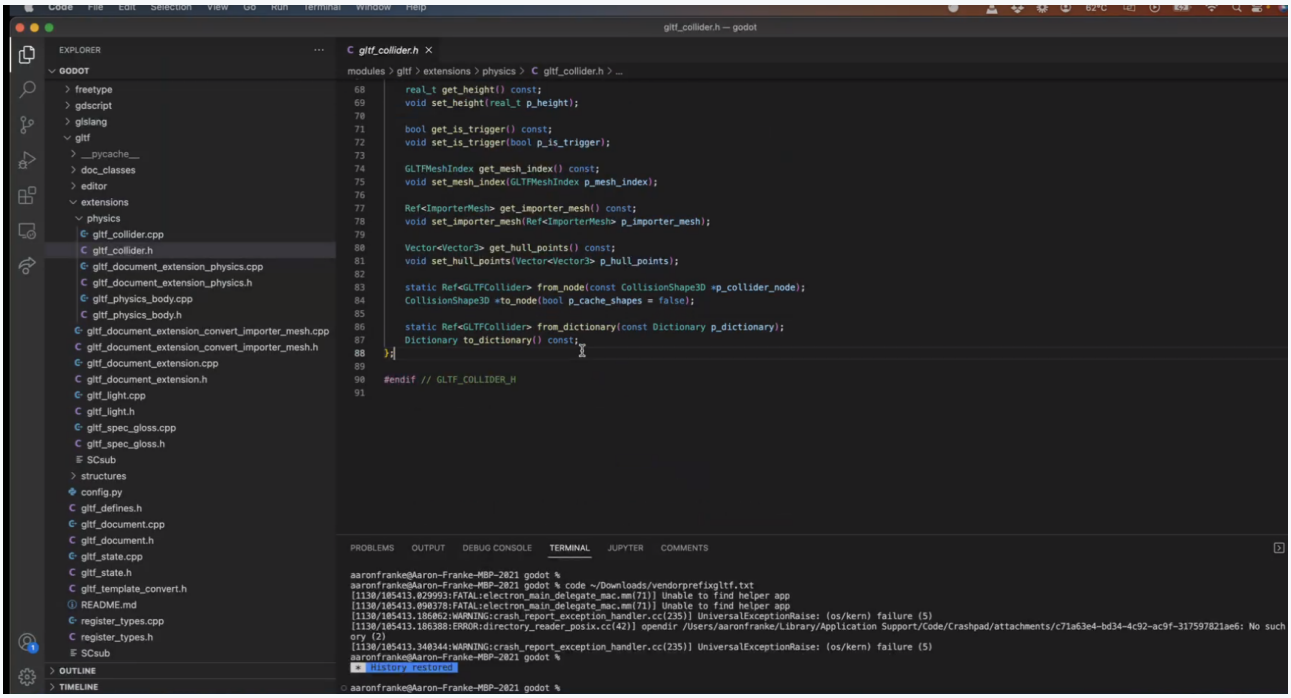
mrmetaverse on Dec 7, 2022 Maintainer Author edited

Also want to point out that this was designed in such a way to be clear about what is OMI, what is godot, and what is gLTF

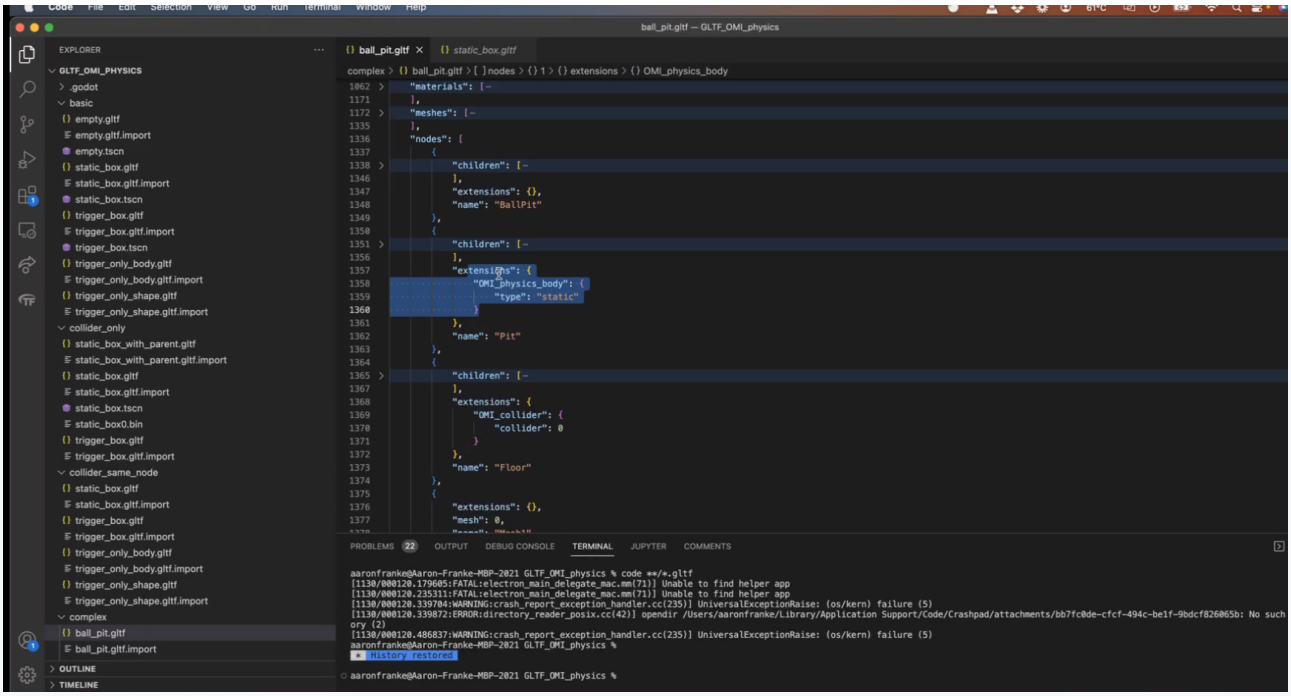


A "competing" standard recently emerged from inside of Microsoft. We could change a few things and use mostly the same codebase for OMI physics *and* Microsoft physics.

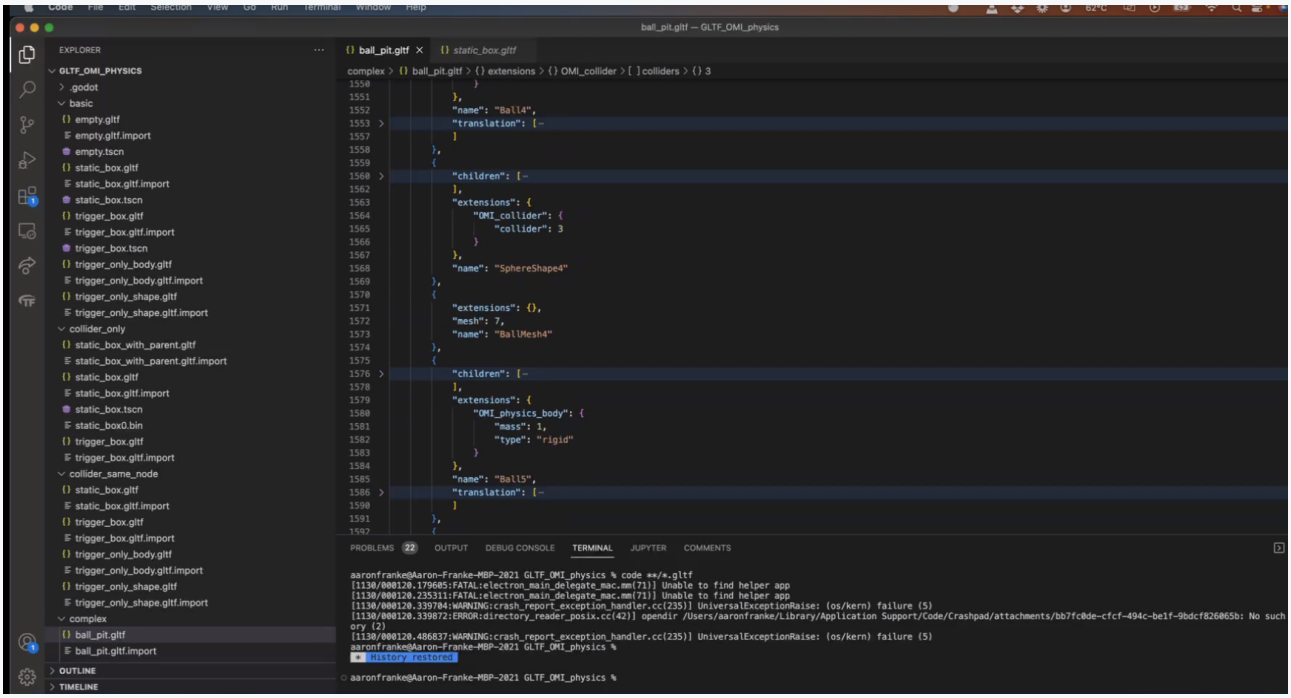


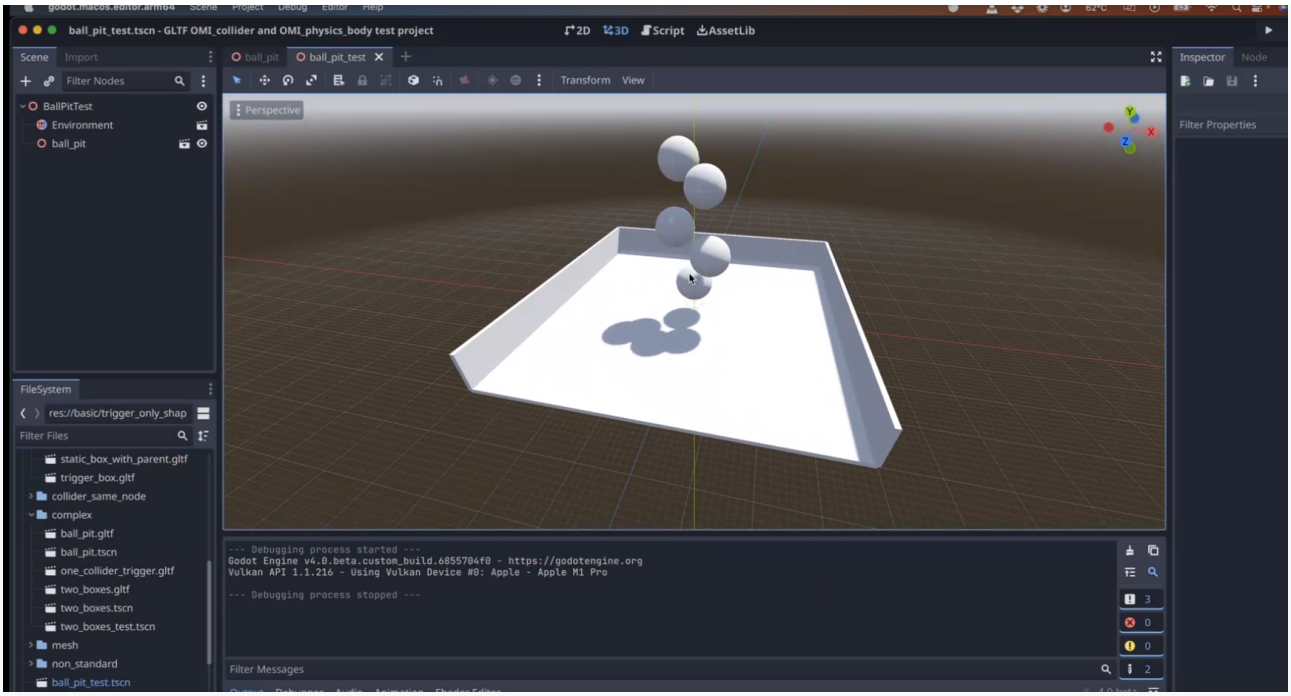


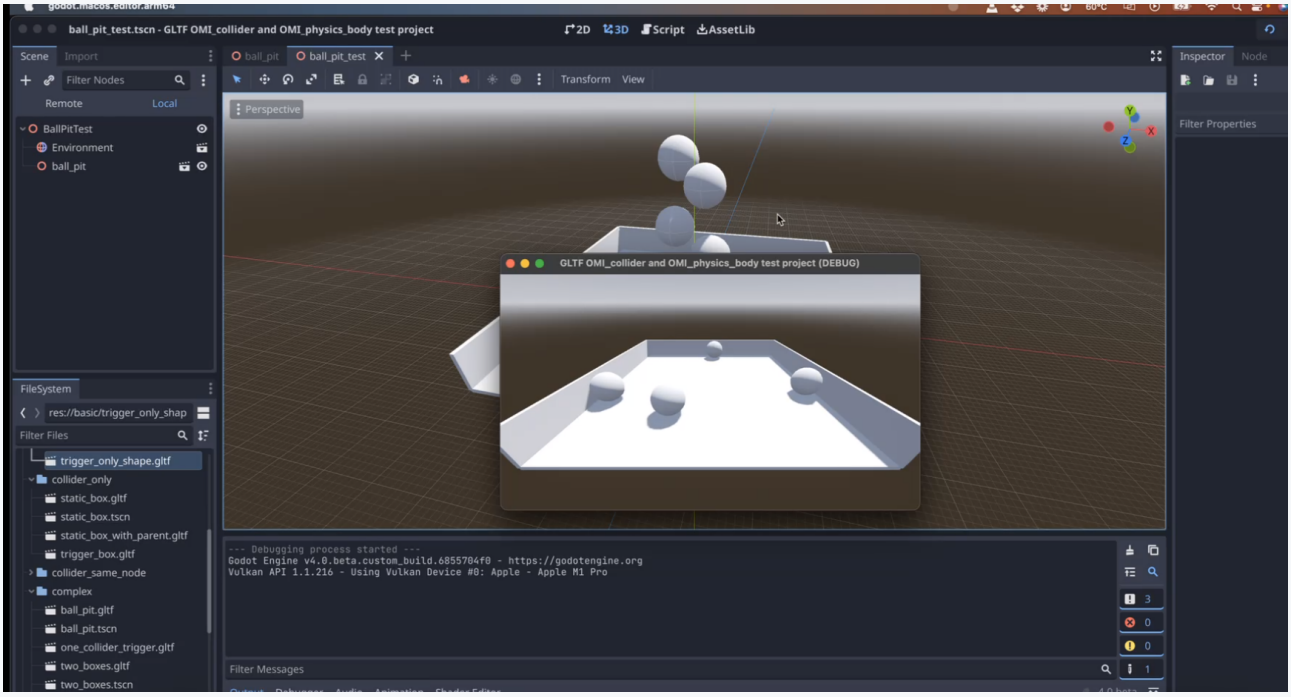
We have a root node, and extensions, etc. (I tried to keep up 😉)



All 5 of these spheres only need the 1 shape resource:





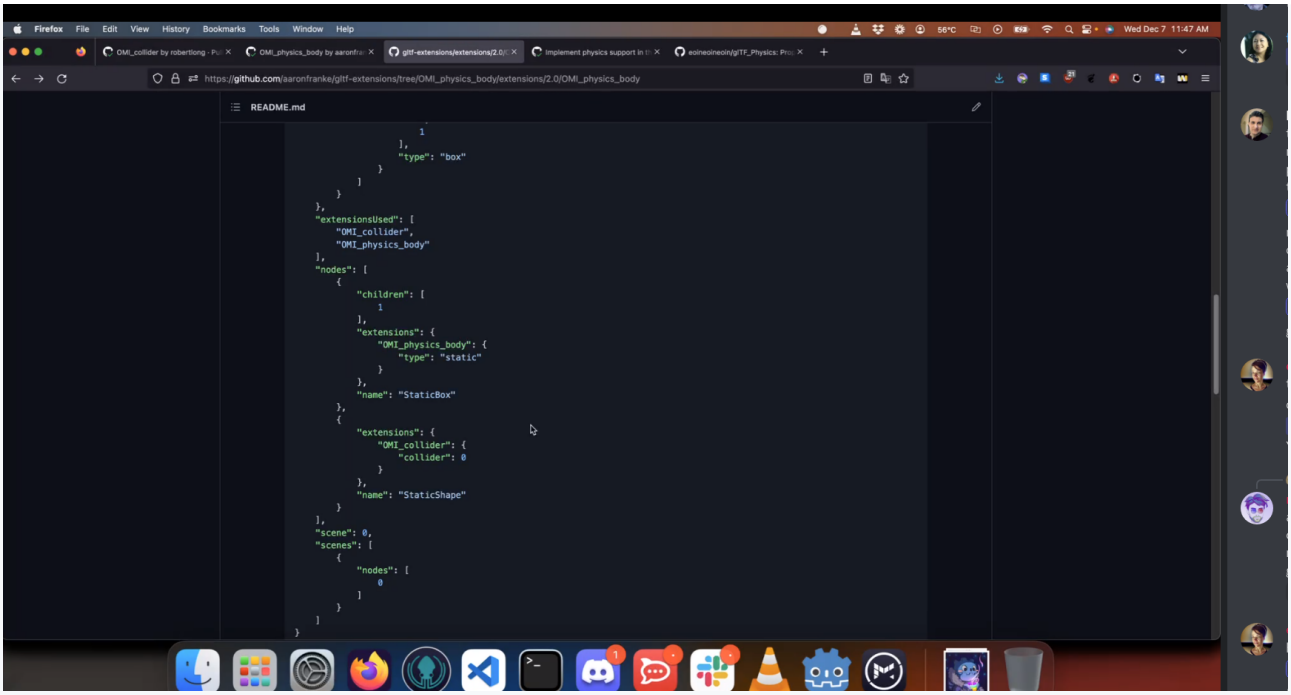


mrmetaverse on Dec 7, 2022 Maintainer Author edited

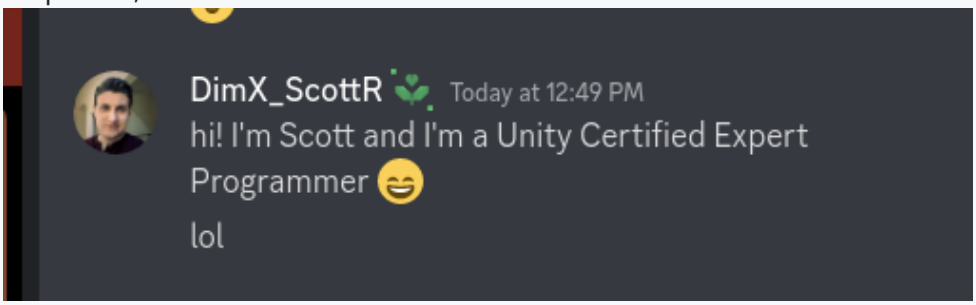
CTA - HOW YOU CAN HELP

We are always looking for groups to test, experiment with, and implement our proposed extensions. (It's also part of our standard process for considering an extension or protocol as "ready").

We would love to work together with other experts, like an Unreal expert, and Unity expert to see what we could change or edit.



Ha perfect, we're 2/3 there then! :



Next week... invite a friend! Who do we know that is an Unreal engine expert who could test/implement these extensions?

Category

General

Labels

None yet

1 participant