WEBGPU Project diary

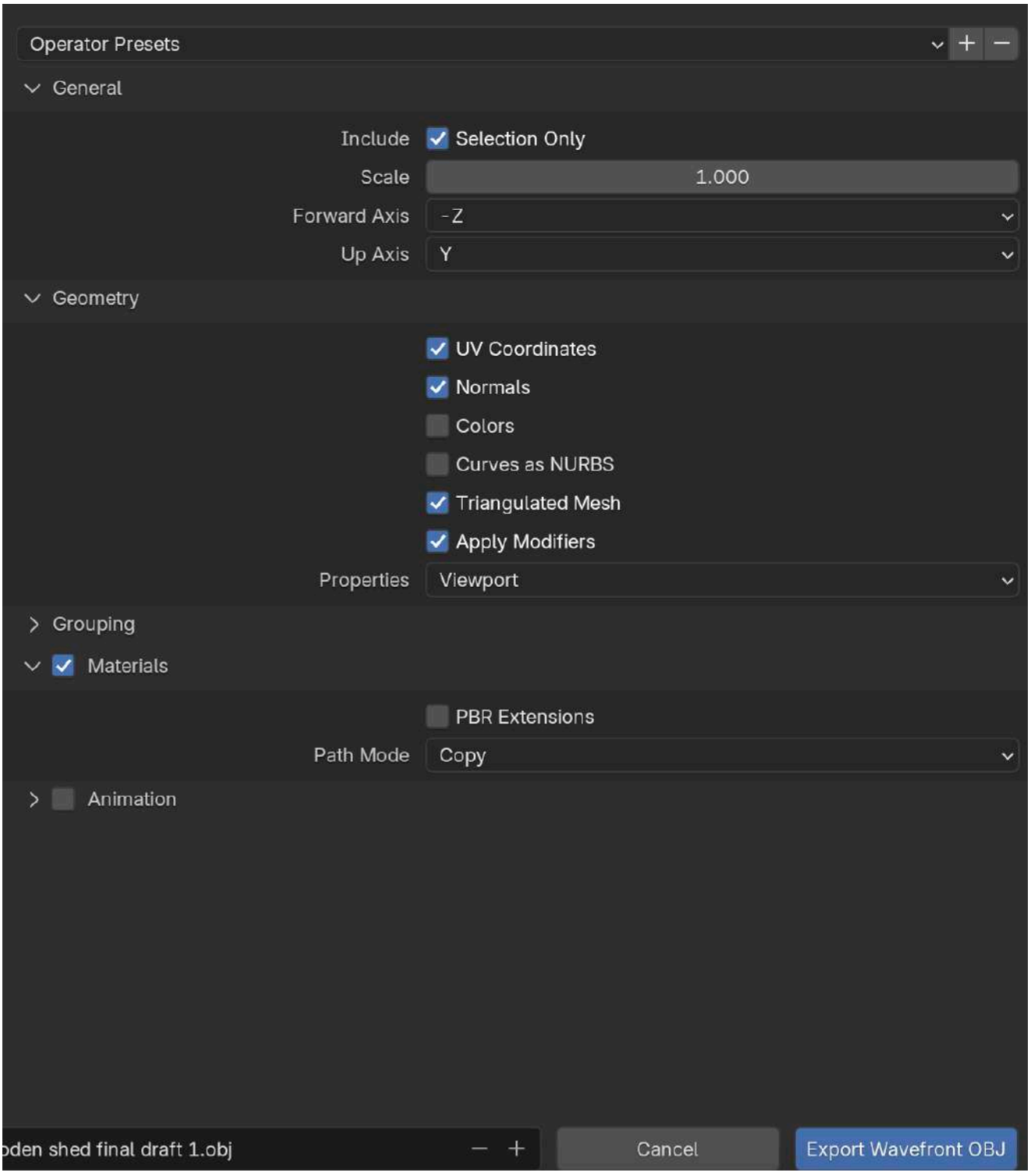
Step 1: Import an object

Task at hand :
-> Scene is massive
-> Scene isn't 2D like in labs

Resources Explored:
https://github.com/toji/webgpu-gltf-case-study/tree/main
https://github.com/Twinklebear/webgpu-gltf?tab=readme-ov-file
https://toji.dev/webgpu-gltf-case-study/

1.1 Export the blender scene :

Using the slides firstly we exported it in OBJ using these settings :


Operator Presets
General
Include
Selection Only
Scale
1.000
Forward Axis
-Z
Up Axis
Y
Geometry
UV Coordinates
Normals
Colors
Curves as NURBS
Triangulated Mesh
Apply Modifiers
Properties
Viewport
Grouping
Materials
PBR Extensions
Path Mode
Copy
Animation
oden shed final draft 1.obj
Cancel
Export Wavefront OBJ

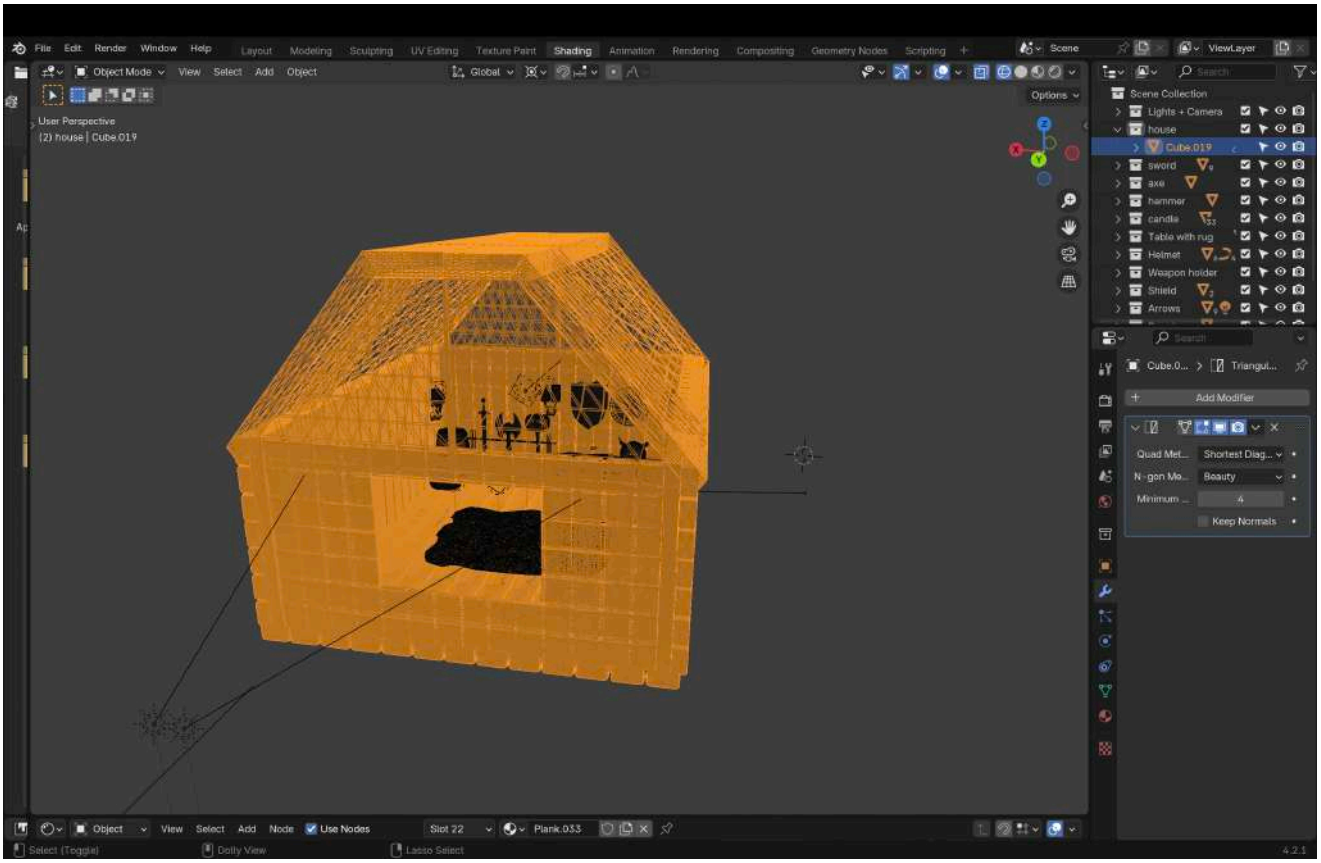
This lead us to have a file with a ton of vertices :

Wooden shed final done 7.obj

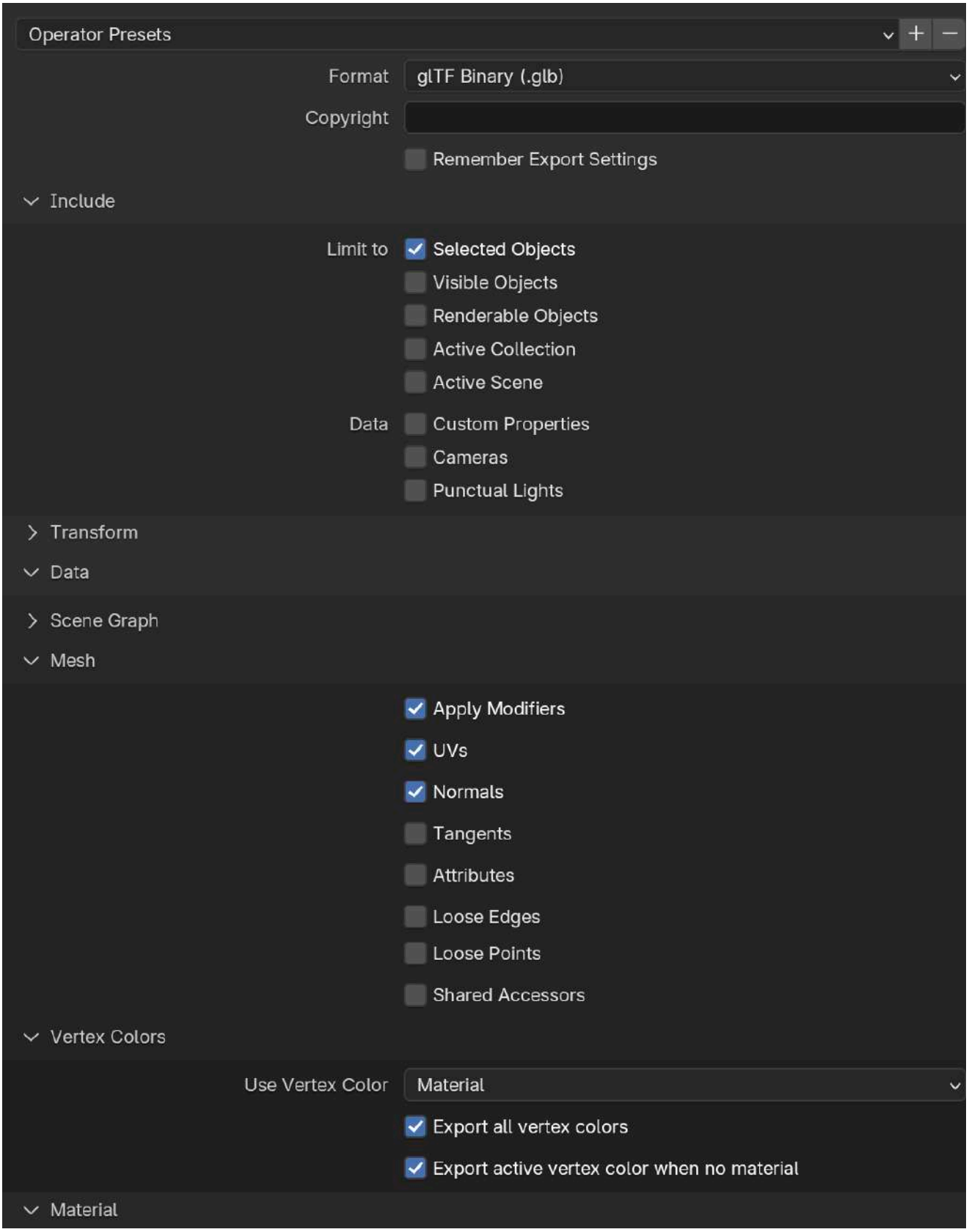
Users > ahmednawaz > Desktop > Wooden shed final done 7.obj

```
10414220 f 2729275/344253/1883932 2730815/344546/1883932 2729393/344254/1883932
10414221 f 2729275/344253/1882432 2726203/344245/1882432 2729391/344541/1882432
10414222 f 2730815/344546/1882433 2729391/344541/1882433 2726235/344245/1882433
10414223 f 2730815/344546/1883931 2729339/344379/1883931 2726236/344547/1883931
10414224 f 2729289/344258/1883930 2730816/344548/1883930 2729362/344414/1883930
10414225 f 2729289/344258/1882428 2726217/344250/1882428 2729392/344250/1882428
10414226 f 2730816/344548/1882429 2729392/344250/1882429 2726249/344545/1882429
10414227 f 2730816/344548/1883929 2729353/344381/1883929 2726250/344414/1883929
10414228 f 2729276/344262/1883928 2730817/344549/1883928 2729364/344481/1883928
10414229 f 2729276/344262/1882424 2726204/344254/1882424 2729393/344254/1882424
10414230 f 2730817/344549/1882425 2729393/344254/1882425 2726236/344547/1882425
10414231 f 2730817/344549/1883927 2729340/344383/1883927 2726237/344263/1883927
```

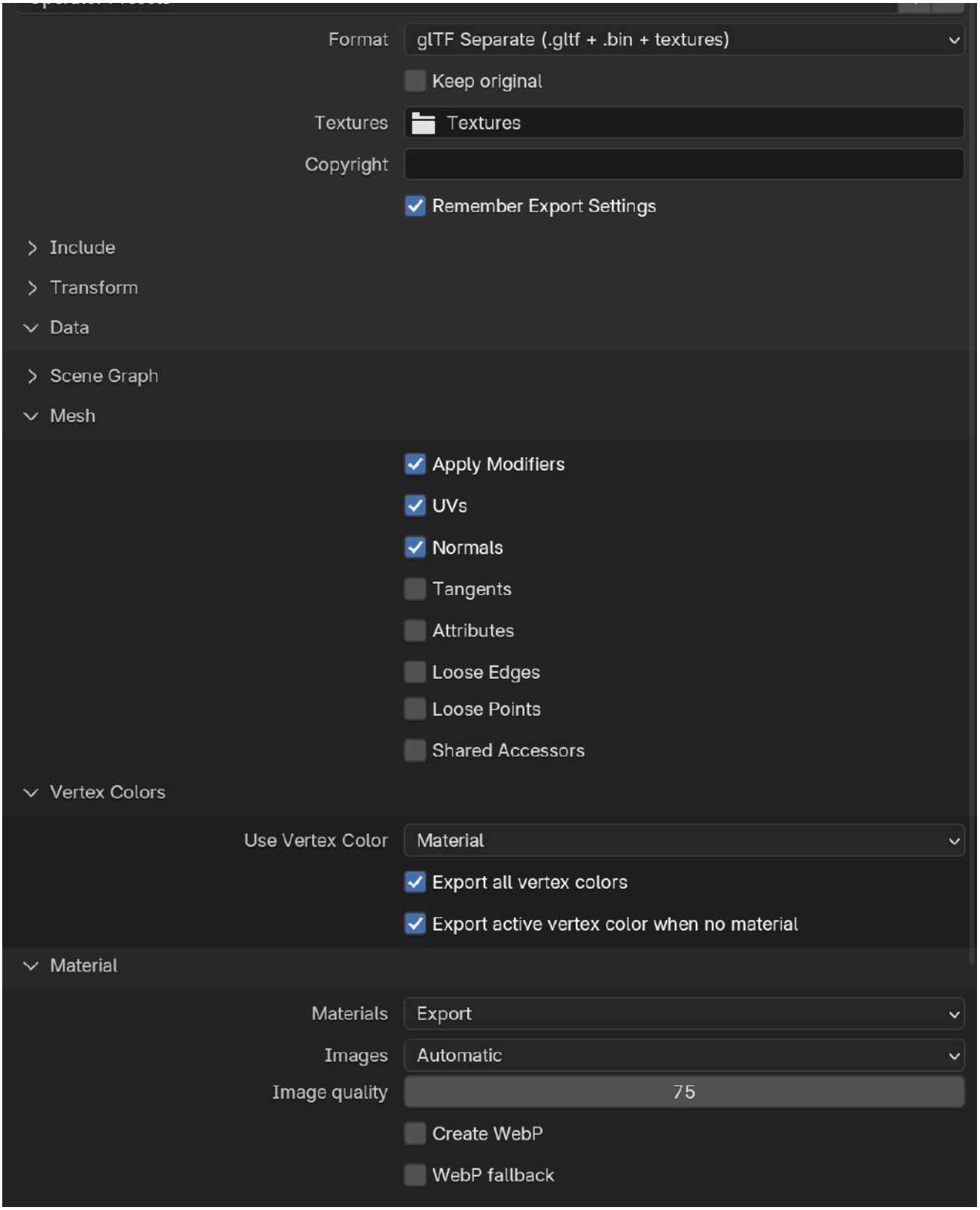
We suspect this is because in edit mode this is how our model looks like (after we trianglate it):



We exported it in GLB using these settings :



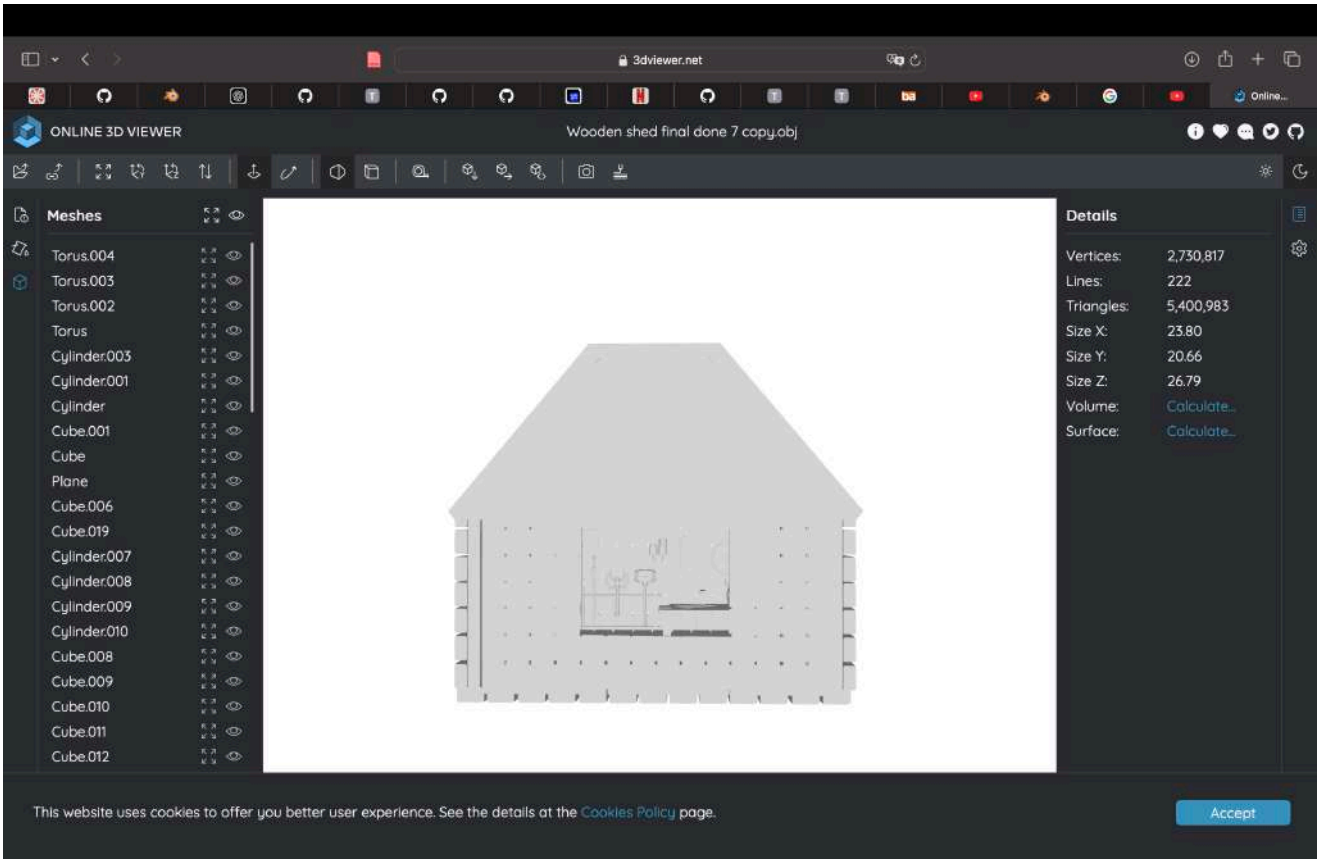
We exported it in GLTF using these settings :



1.2: Testing the import :

We used this 3D model viewer to verify if our objects and textures properly exported from blender:
https://3dviewer.net/#
https://githubdragonfly.github.io

Using OBJ we got :



Cons:

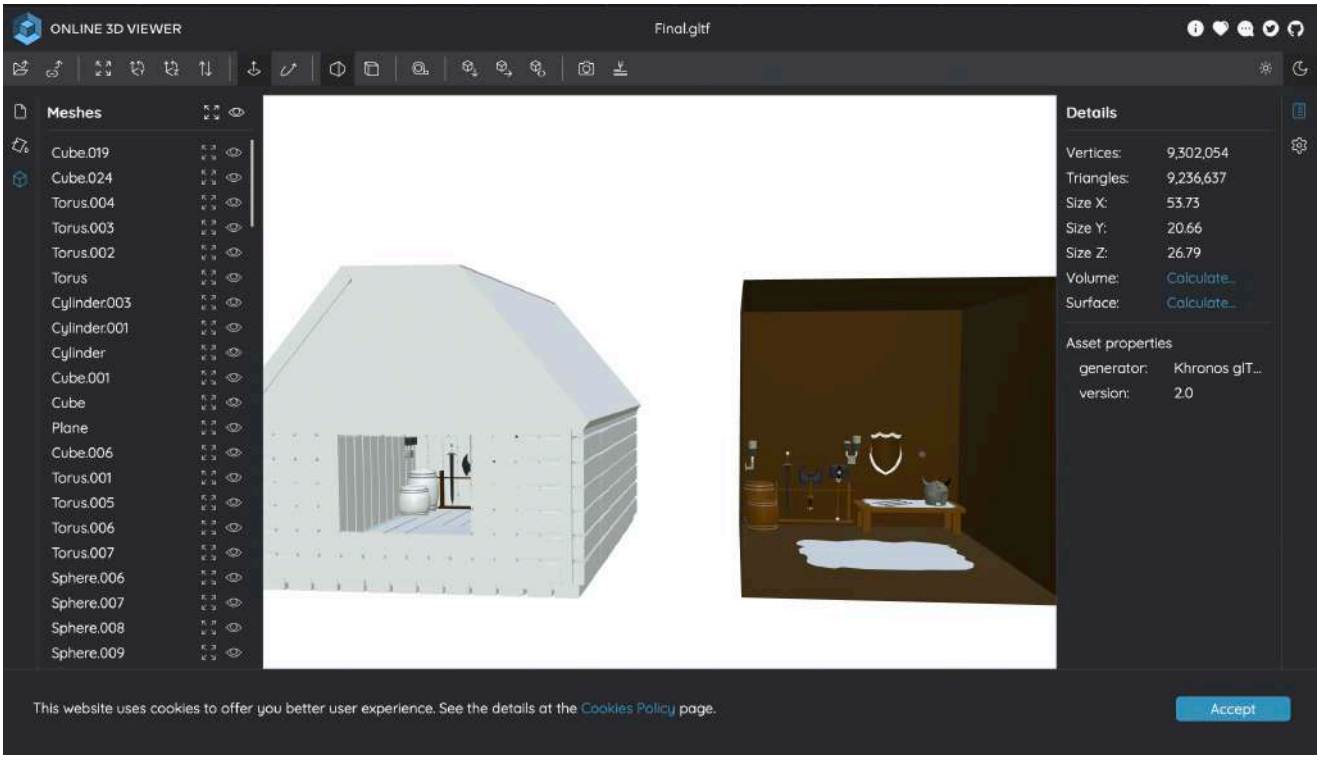
->Given labs dont import this at all

->Web browser times out because loading vertices takes too long

-> No textures have been imported

Using GLTF we got :

(file has been edited slightly)

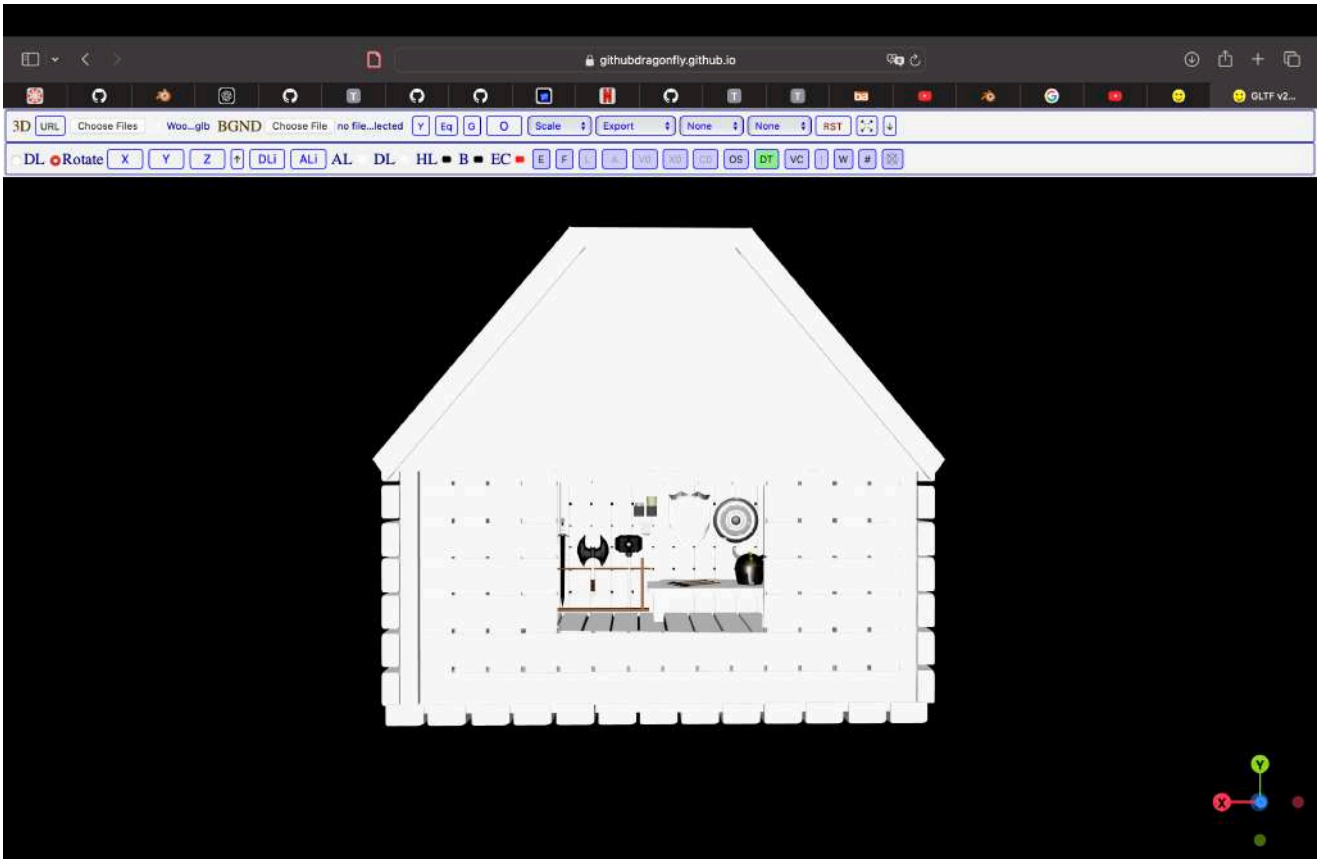


Pros:

->Most textures

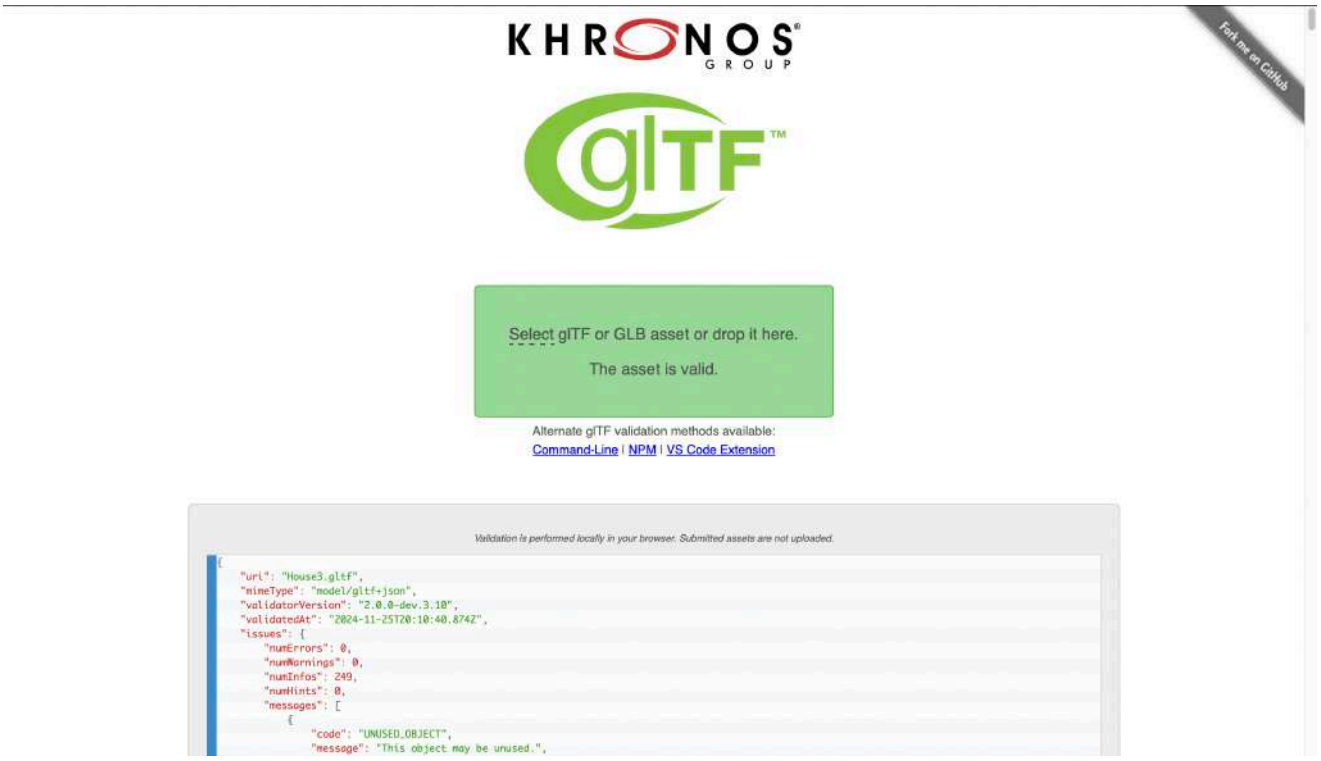
->Model looks non-broken
->Loads up efficiently and quick
Cons:
->Materials look flat
->Utilizes 2 files instead of 1 to load the object

Using GLB we got :



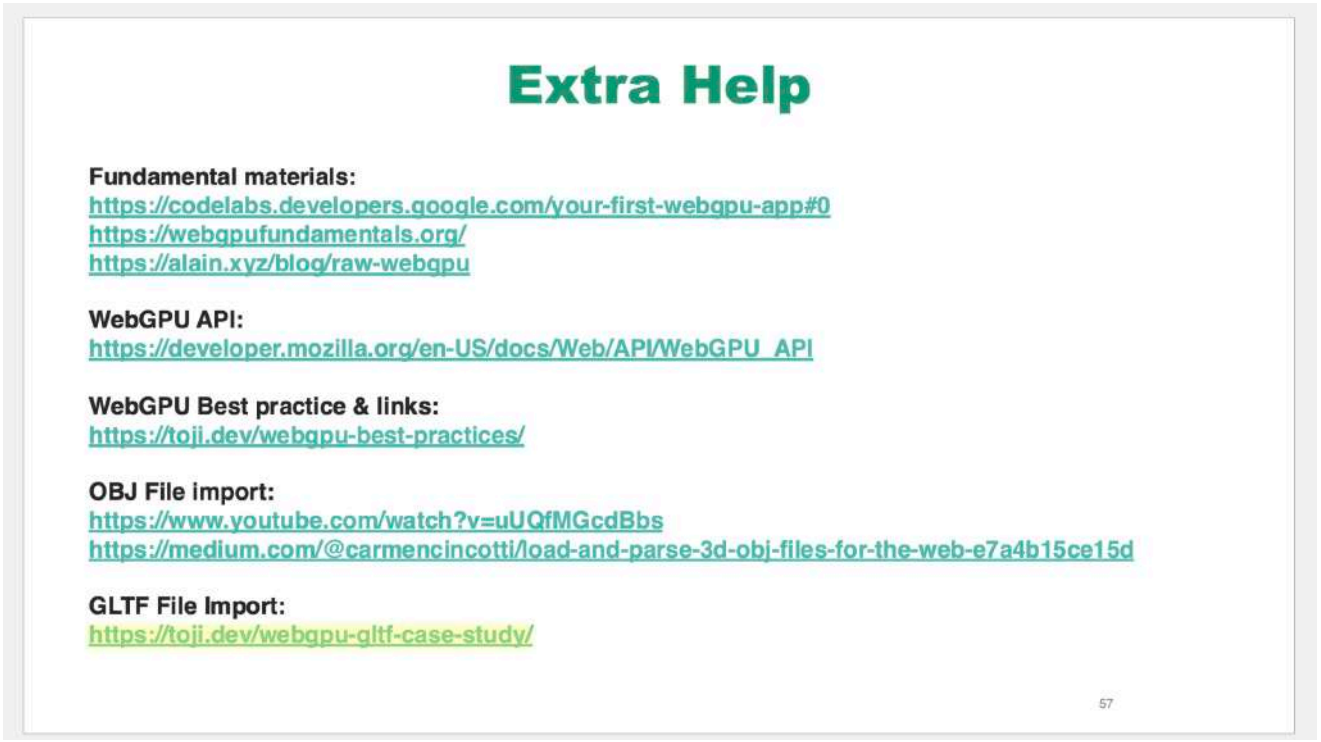
Pros:
->Most textures
->Model looks non-broken
->Uses 1 file
->Loads up efficiently and quick

We uploaded the files into GLTF checker to be sure its exported properly :
https://github.khronos.org/glTF-Validator/



-> Decided to move forward with glb / gltf
Glb-> same as gltf but packaged into 1 file
glTF->in json format , exports textures separately

1.3: Load into Chrome Browser :
Referring to the slides , there is a GitHub page resource provided to us :



The provided link has a gLTF to webGPU workflow as well as a github repository as seen below that we downloaded and ran :

toji / webgpu-gltf-case-study

Issues 2 | Pull requests | Actions | Projects | Security | Insights

webgpu-gltf-case-study Public

Watch 5

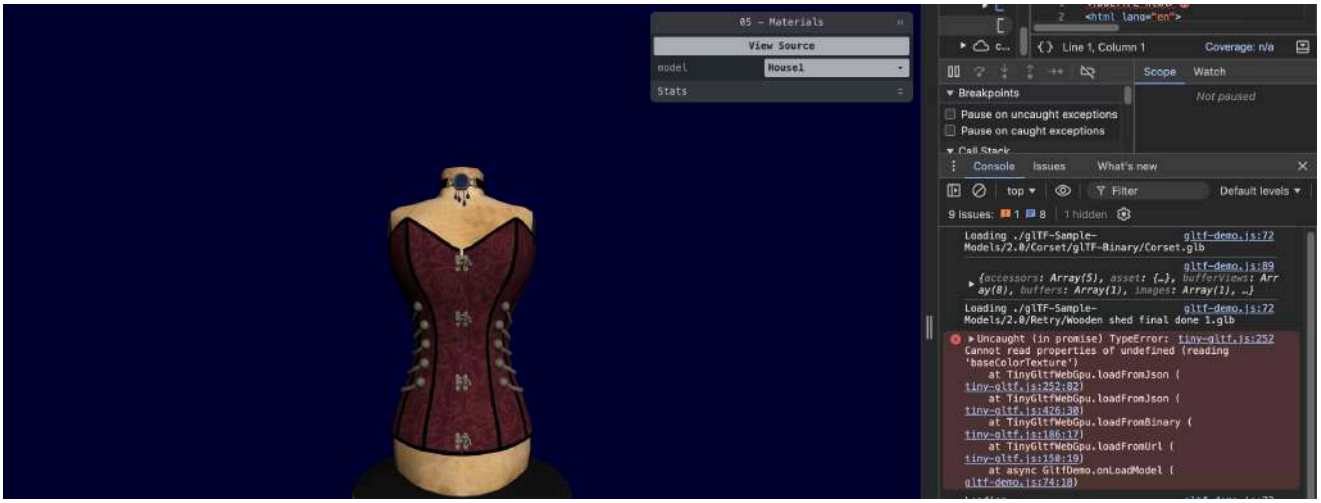
main | 1 Branch | 0 Tags | Go to file t | Add file | Code

toji Convert to using precomputed normal matrices ✓ bd99585 · 9 months ago 42 Commits

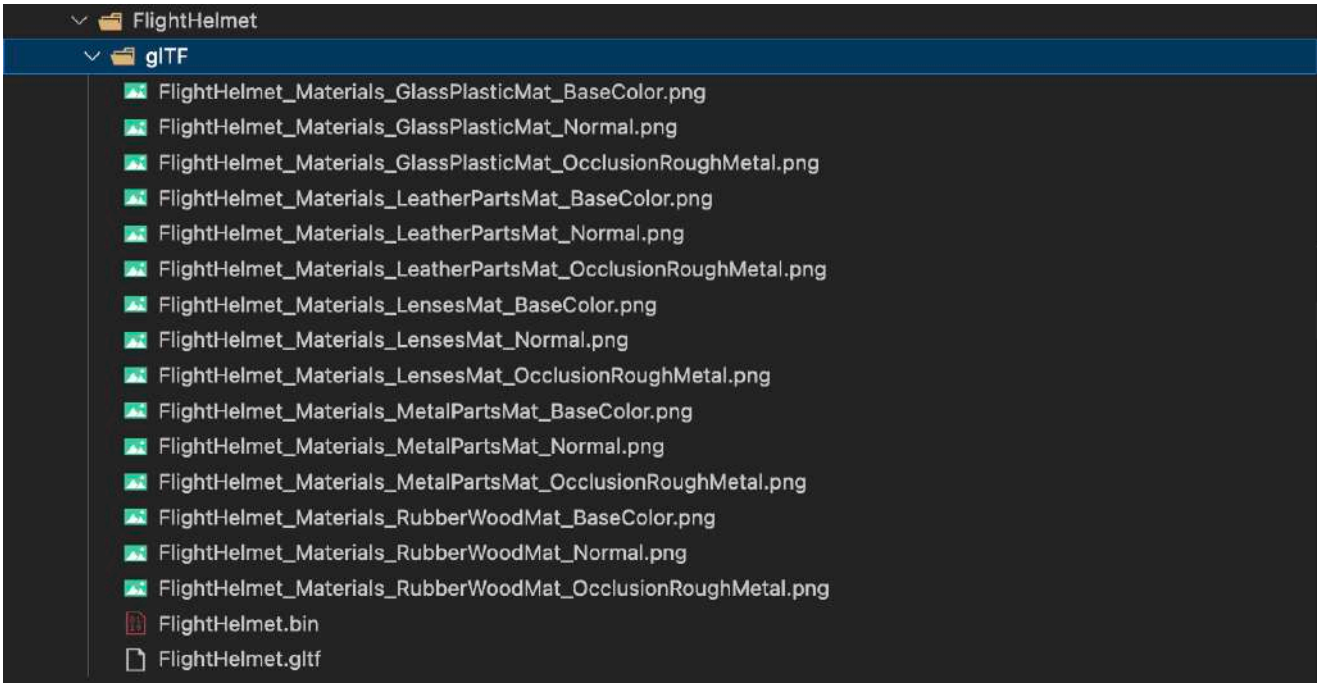
_data	Extracted samples back out into standalone files for eas...	2 years ago
_includes	Trying this out as a Jekyll page?	2 years ago
_layouts	Redirect to new article site	last year
assets	Added a header image	2 years ago
samples	Convert to using precomputed normal matrices	9 months ago
.gitignore	Trying this out as a Jekyll page?	2 years ago
01-naive-renderer.md	Extracted samples back out into standalone files for eas...	2 years ago
02-buffer-layouts.md	Extracted samples back out into standalone files for eas...	2 years ago
03-pipeline-caching.md	Extracted samples back out into standalone files for eas...	2 years ago
04-instancing.md	Extracted samples back out into standalone files for eas...	2 years ago
05-materials.md	Extracted samples back out into standalone files for eas...	2 years ago
404.html	Trying this out as a Jekyll page?	2 years ago
Gemfile	Trying this out as a Jekyll page?	2 years ago

Click to launch **Sample 01 - Naive Rendering**

We changed the code up to try to import our model and got several import errors like:



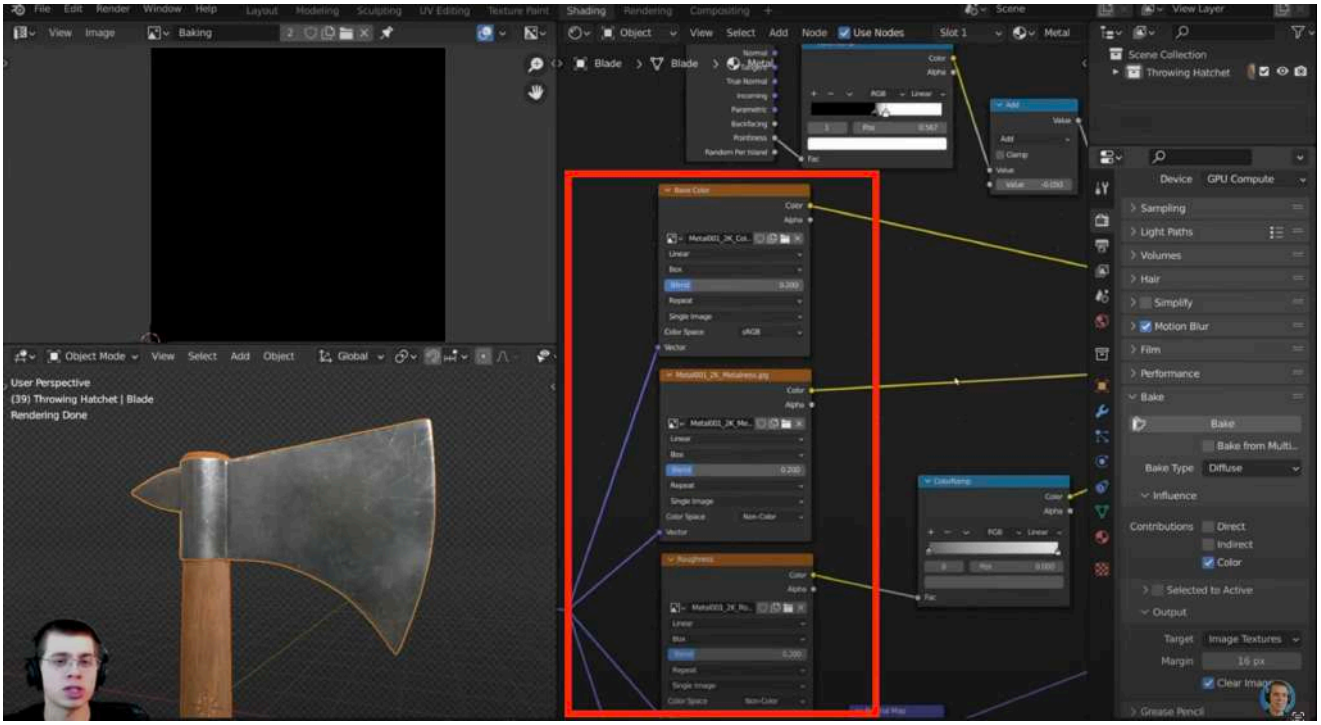
Looking at their model structure we learn that lots of image maps textures are used to display material in this project :



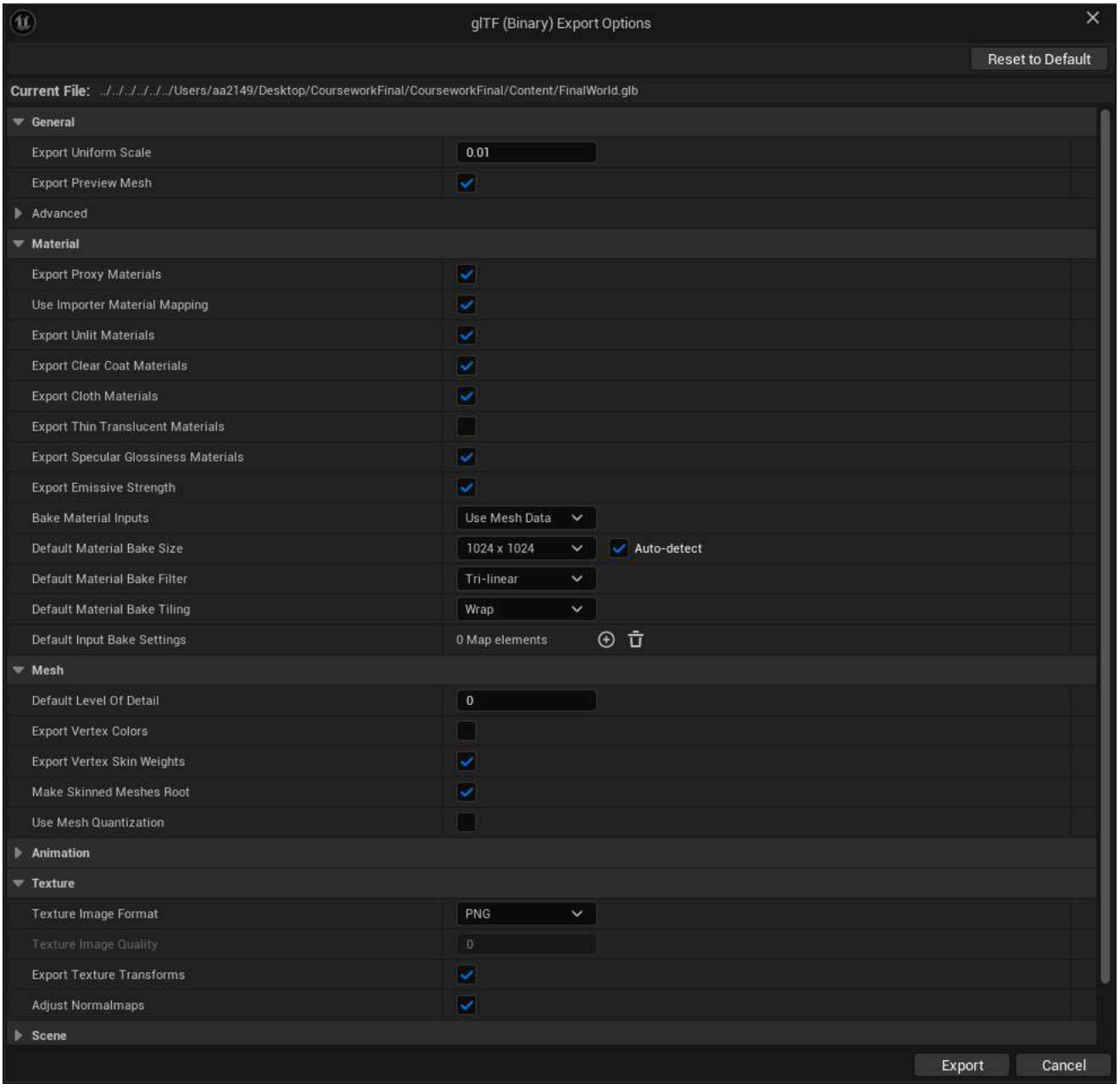
Which probably means we also need to export our textures accordingly . To do so we head back to blender :
Looked into how this is done, researched into baking materials by using :
https://www.youtube.com/watch?v=h3OsSzLhIUo
https://www.youtube.com/watch?v=TNE3z424YpM

https://youtu.be/wG6ON8wZYLc?si=cxrCmbVUuDHHNa31
Looking at the video I realized glTF works best with image textures. While doing UnReal Engine part of this course we fixed all the missing textures by importing image texture maps

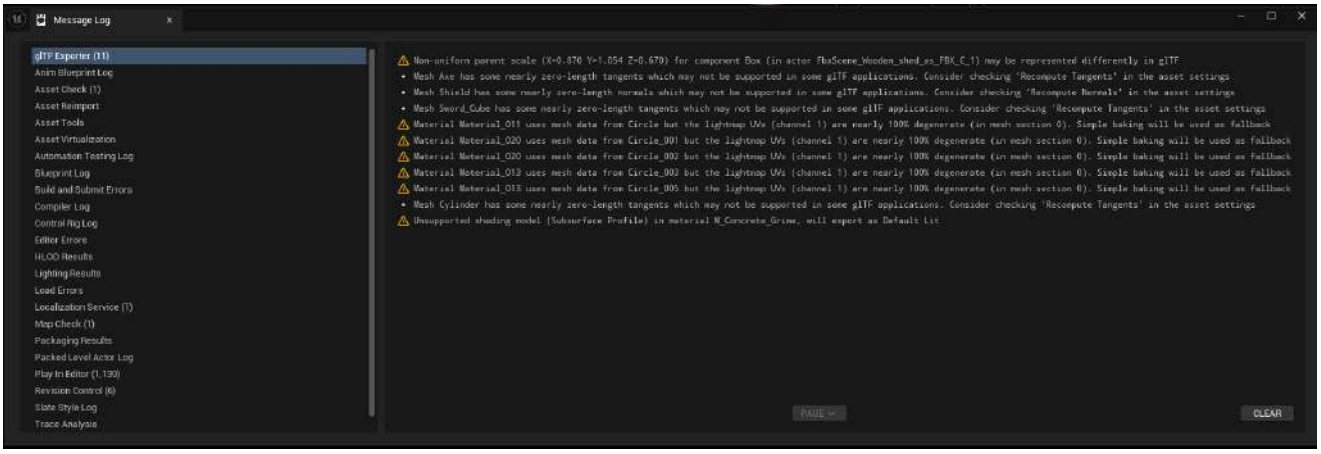
provided.


Baking
Object Mode
User Perspective
(39) Throwing Hatchet | Blade
Rendering Done
Shading
Use Nodes
Scene Collection
Throwing Hatchet
Device
GPU Compute
Sampling
Light Paths
Volumes
Hair
Simplify
Motion Blur
Film
Performance
Bake
Bake Type
Diffuse
Influence
Contributions
Direct
Indirect
Color
Selected to Active
Output
Target
Image Textures
Margin
16 px
Clear Image
Grease Pencil

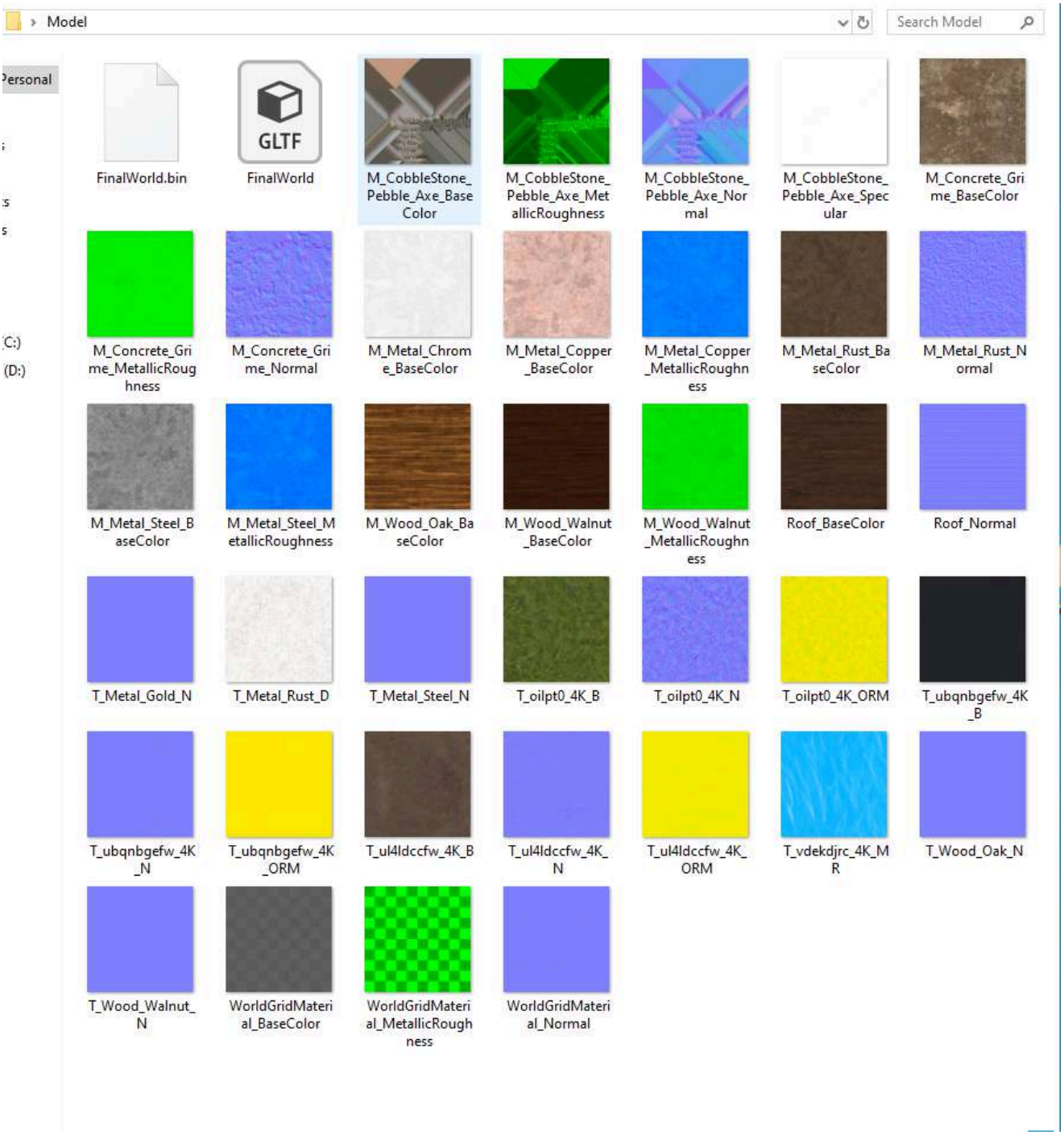
Now ill attempt to export my model from UnReal as glb.



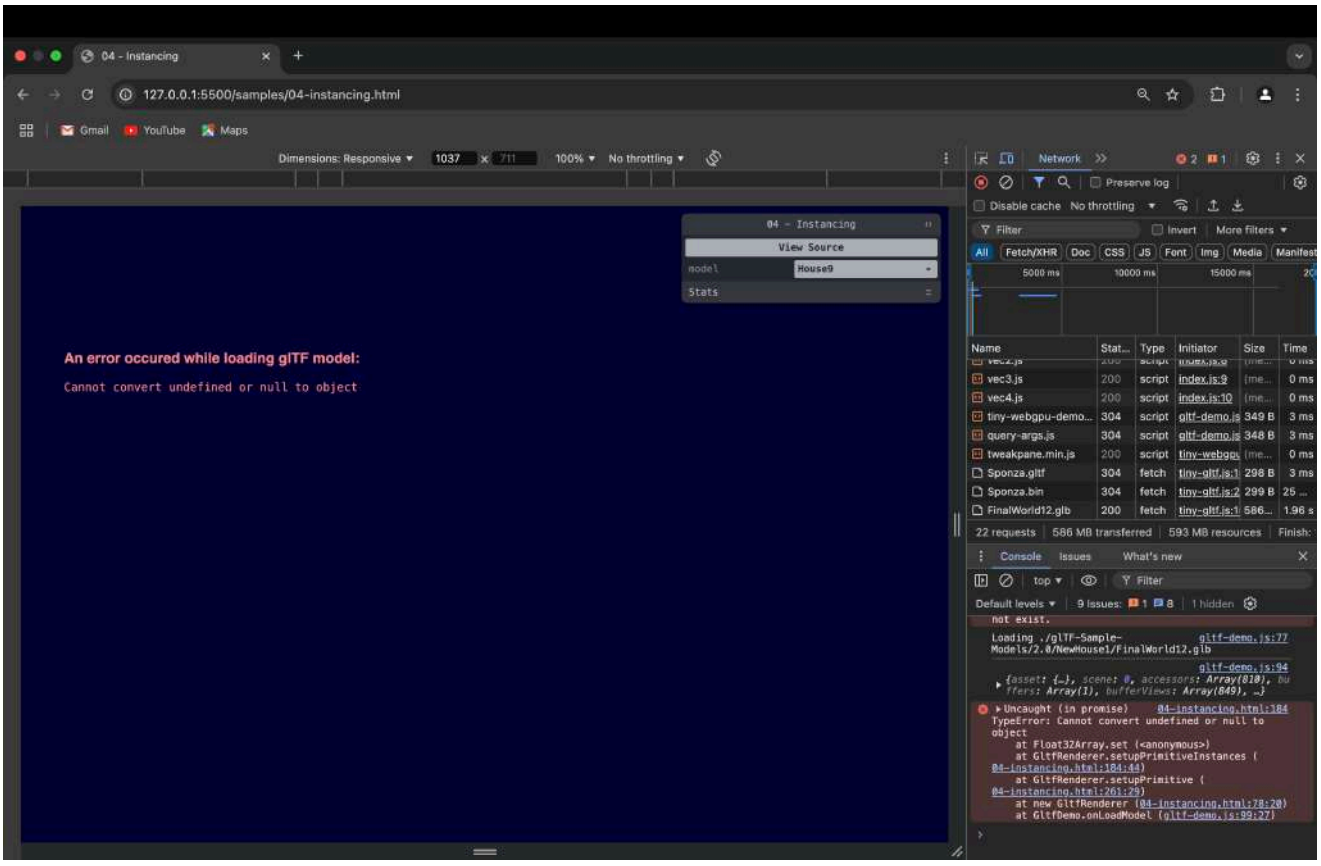
On export I again got errors:



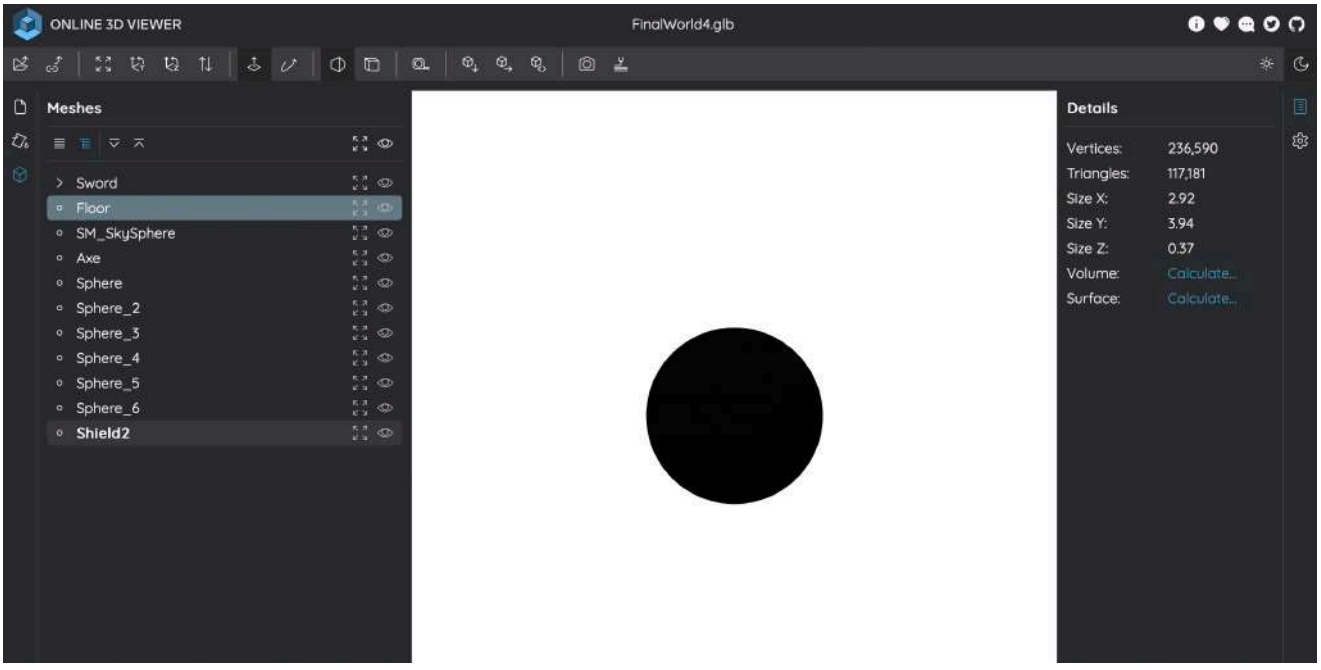
I also tried out the GLTF version and export I was able to see a lot of image map textures which meant this was going on the right track:



After exporting I attempted to load up the model again and received a different error:

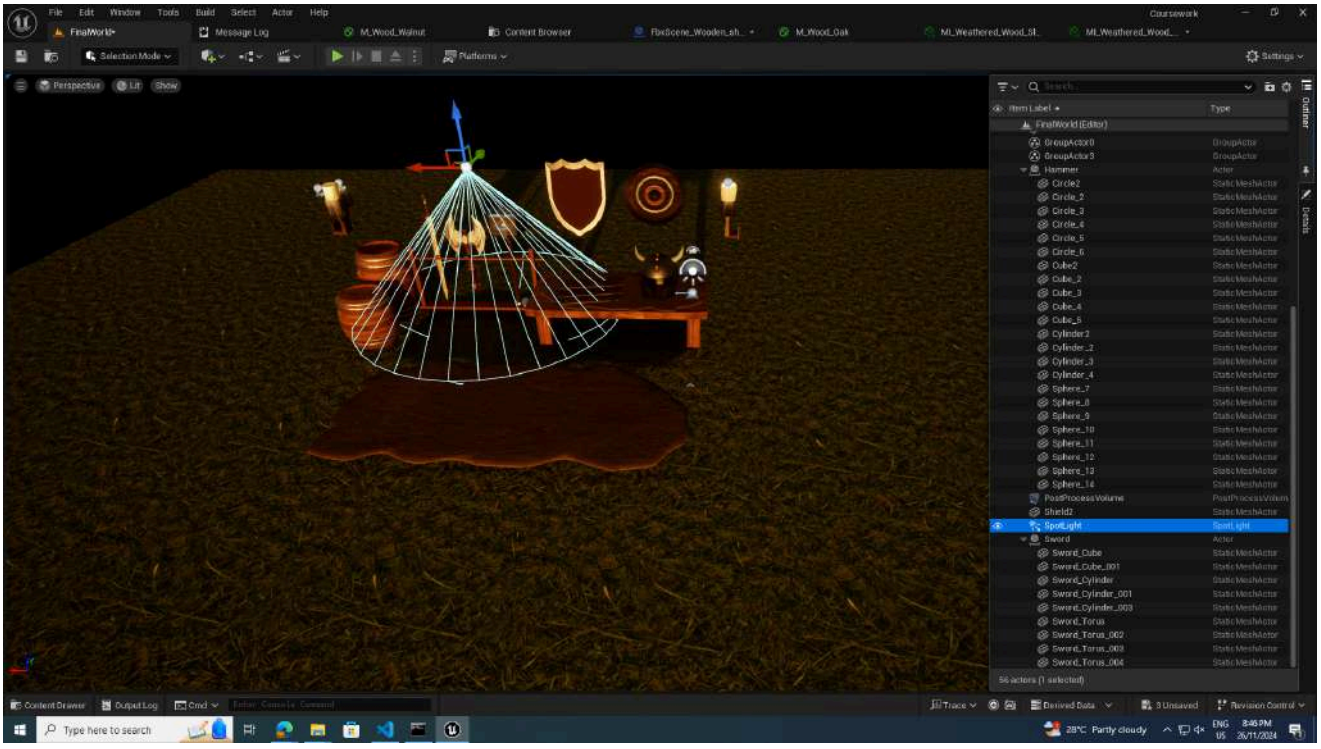


In the model viewer our export looked a little bit like:



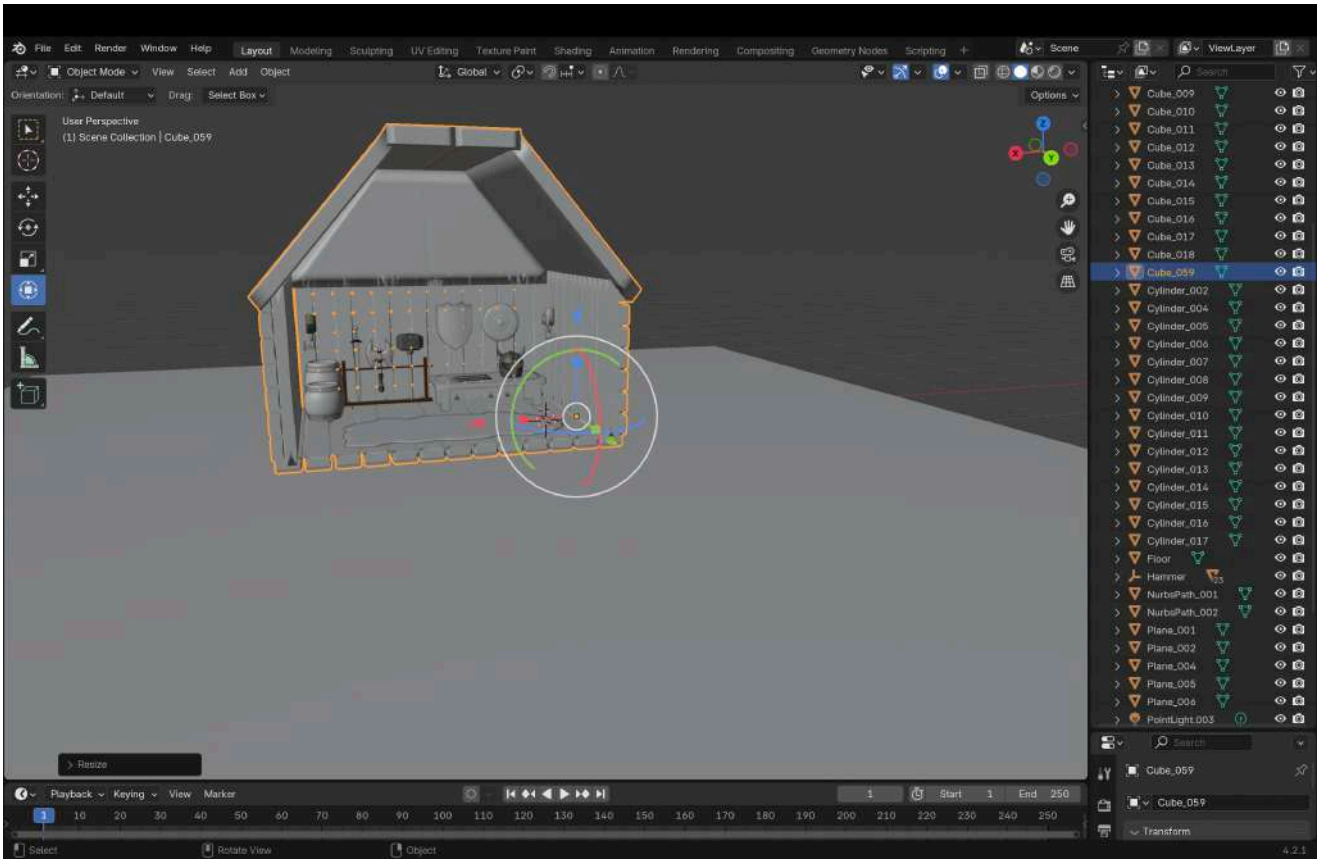
In the GLTF validator it showed a lot of unused objects were the reason why this was happening:

I headed back into unreal engine to clean up extra things like Post process volumes , any lightings , atmosphere or anything other then the mesh itself.

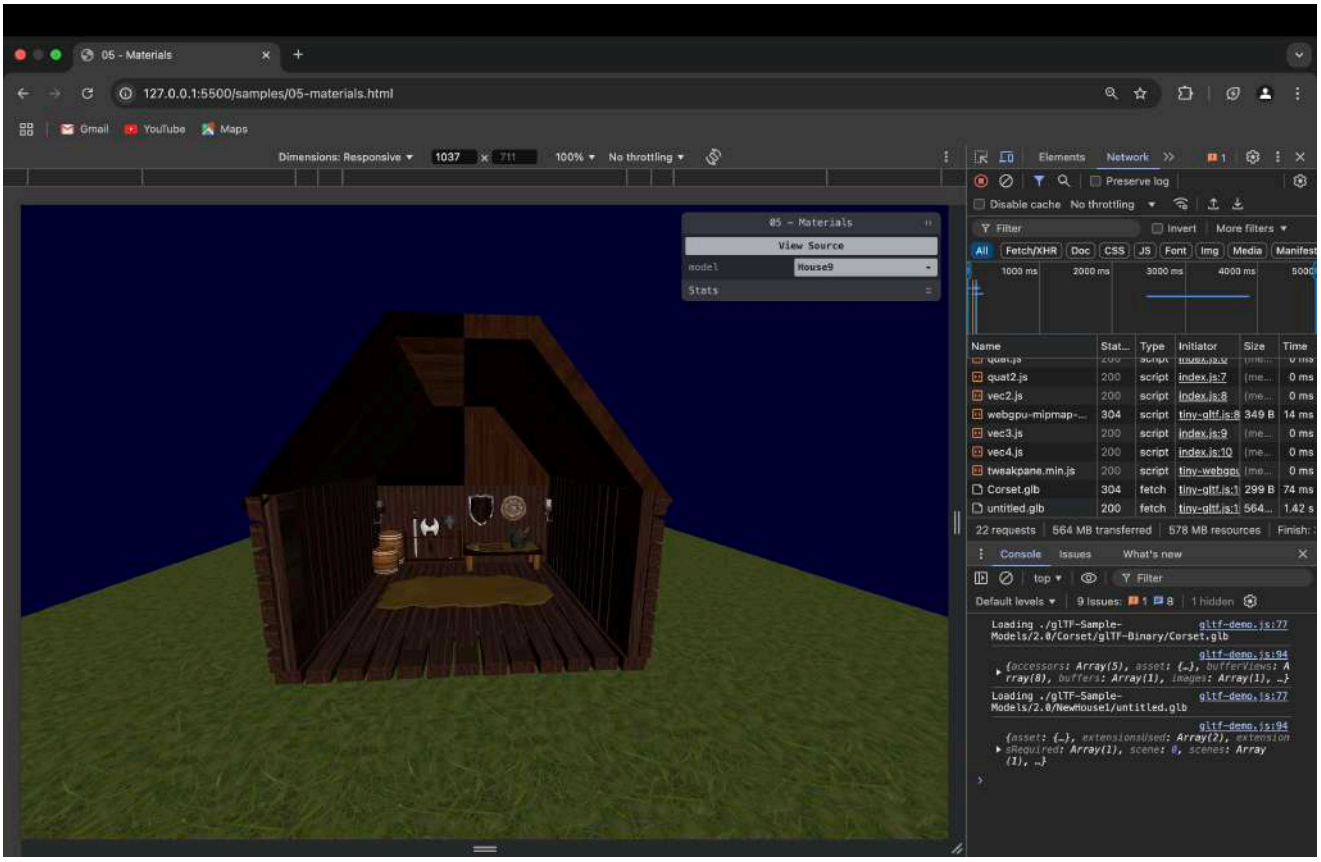


After that i loaded the model up in blender to make the final changes like resizing and verifying that my model had the textures loaded up properly. I ended up tweaking some

properties like roughness, metallic properties in blender

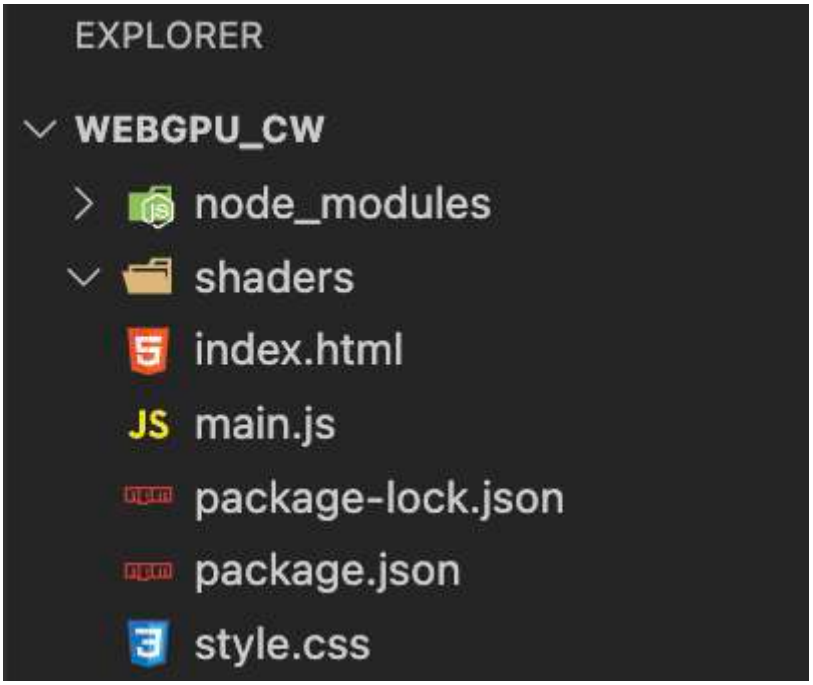


On the next reload we were successful to load our model up.

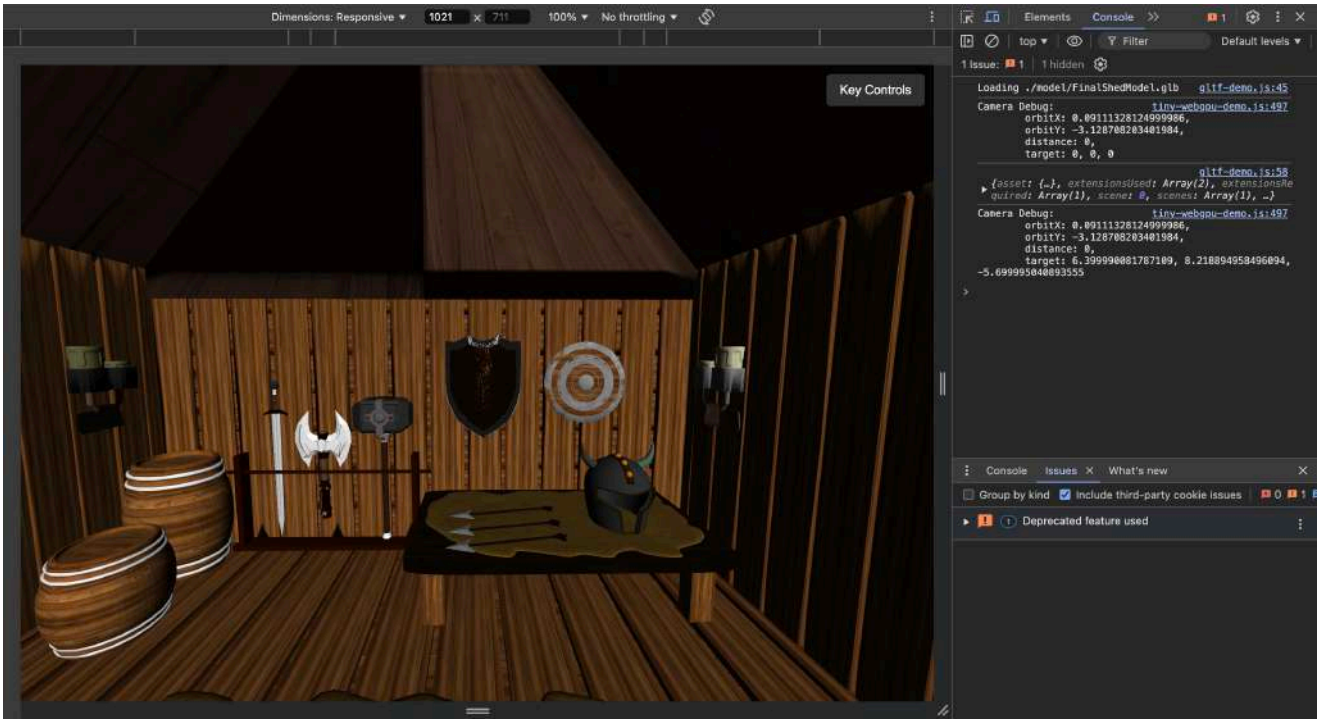


Step 2: Implement Camera and Lights

I set up a new node project with this structure :



We fixed the starting scene to look inside our shed by using console to get the precise values of the view.



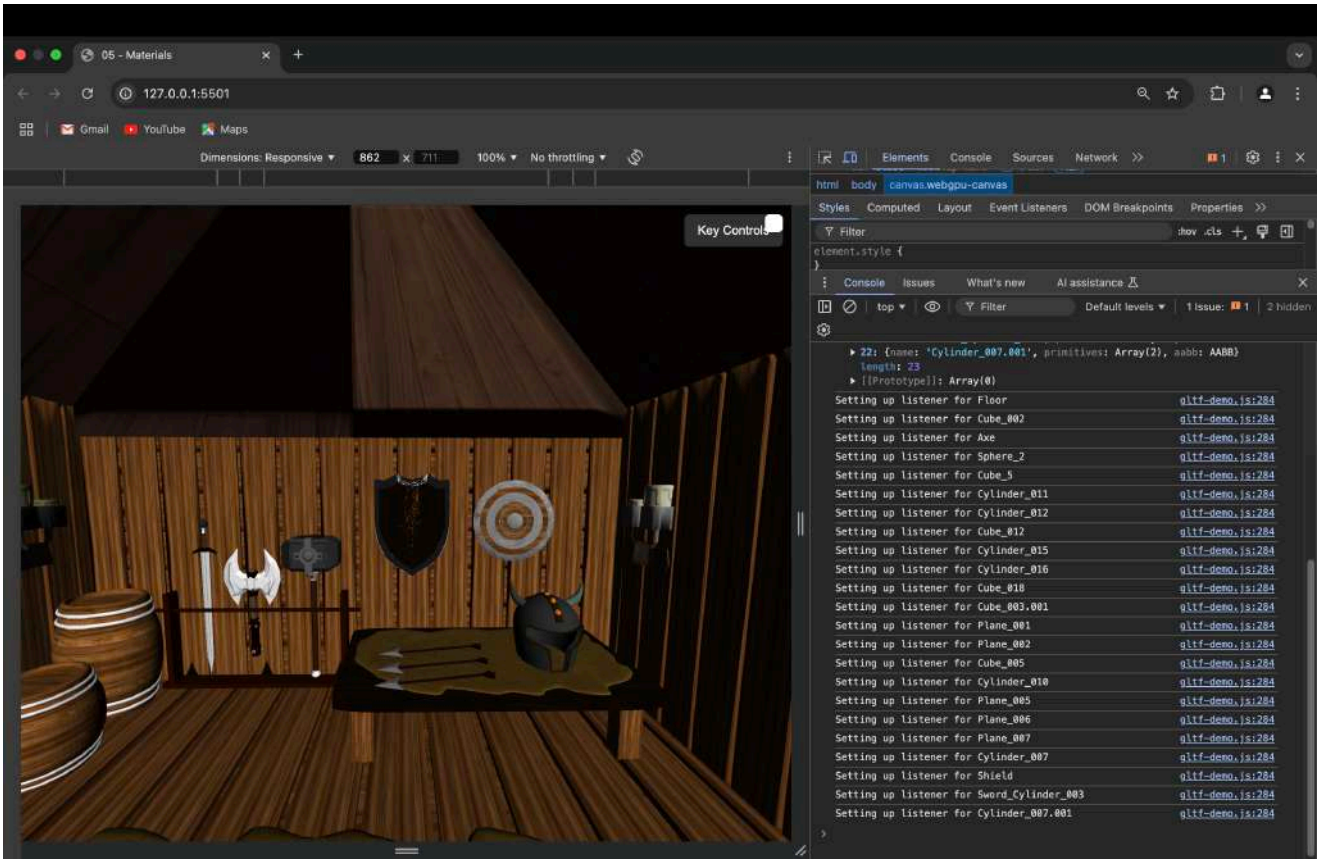
To add interaction my idea is to :

1. make a on click function
2. when the mesh is being loaded, add a on-click event listner on each one of them
3. add checks like allow 1 mesh to be animated at a time

The next step was to add interact to my meshes however I needed a way to initialize an on-click function upon them so I attempted to set up event click listener for the meshes and

utilize a RayCasting mechanism to detect my clicks on the meshes. I kept getting errors like this :

" The error Uncaught TypeError: Cannot read properties of undefined (reading '0') occurs in the mat4.invert function because it is trying to invert a matrix that is undefined or improperly initialized. This happens when the getRay method in your Camera class calls mat4.invert on an undefined or invalid projection or view matrix."



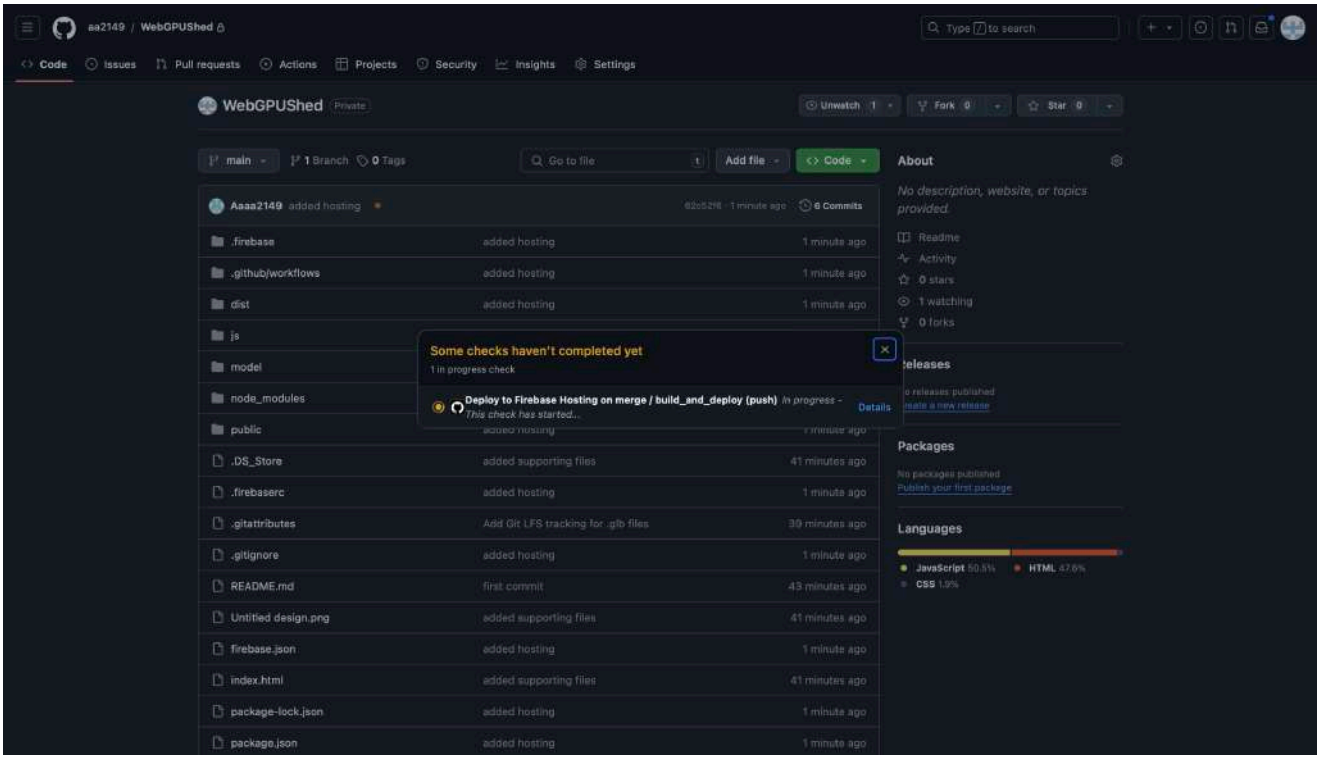
Another way we implement this was we made a method in the index.html that listens for key clicks . Once a number is clicked we load the model and allow interaction to take place , this happens in the index.html. I mapped the keys to the models we have to fetch.

```
const Model = {
  1: './model/barell.glb',
  2: './model/HammerFix.glb',
  3: './model/Axe.glb',
  4: './model/CandleHolder1.glb',
  5: './model/CircleShield.glb',
  6: './model/SwordFix.glb',
  7: './model/helmet.glb',
  H: './model/FinalShedModel.glb',
};
```

I also Implemented a sky box using this resource :
https://opengameart.org/content/sky-box-sunny-day
https://webgpufundamentals.org/webgpu/lessons/webgpu-skybox.html

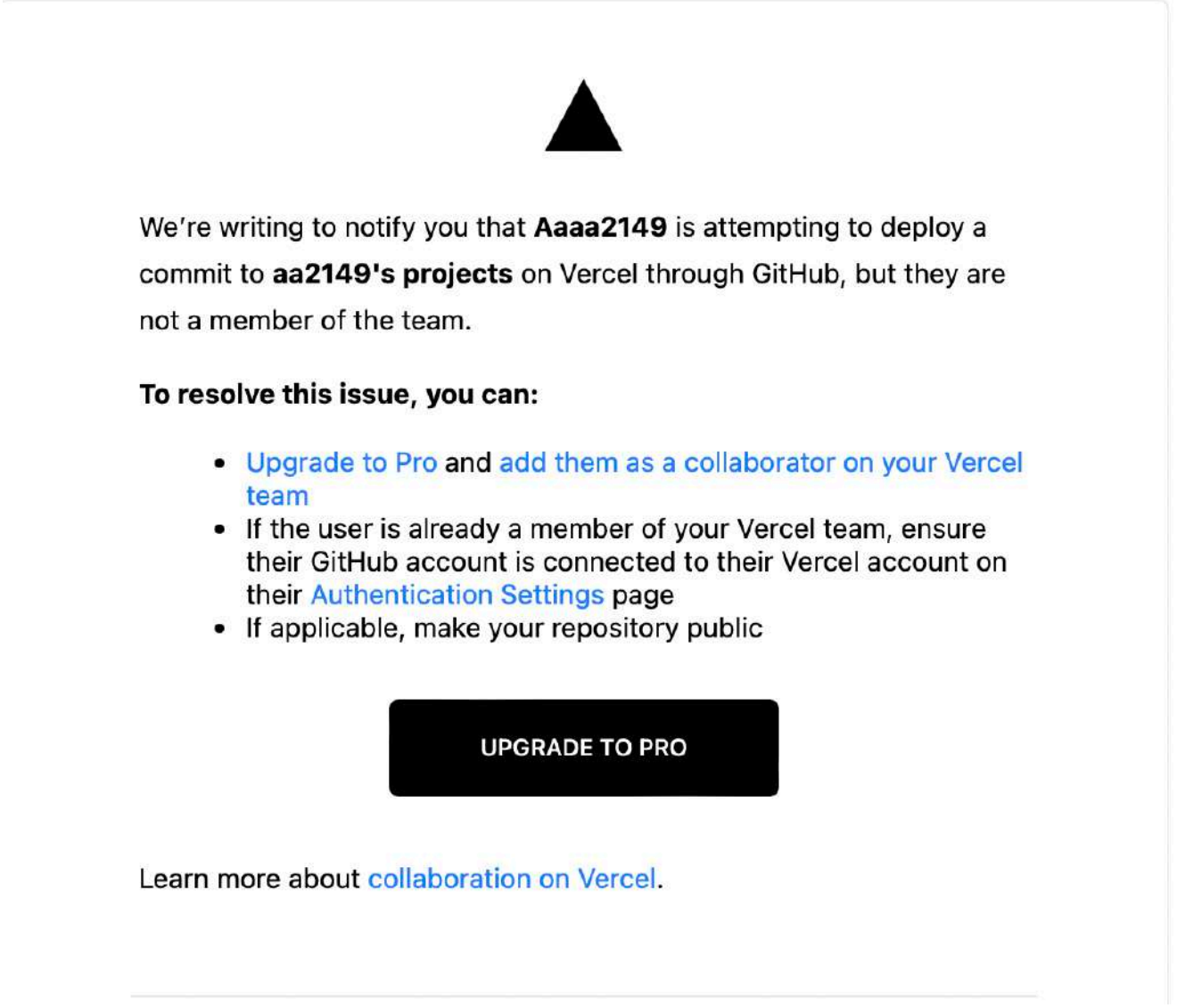
Extra Final Step : Hosting
When web applications are relevent it makes the most sense to have them hosted. For that i tried several method like firebase , vercel and netlify.

Firebase:
I tried doing it from firebase hosting since it offers a good 5GB free limit to host application however i kept have build issues and so I tried out another way.

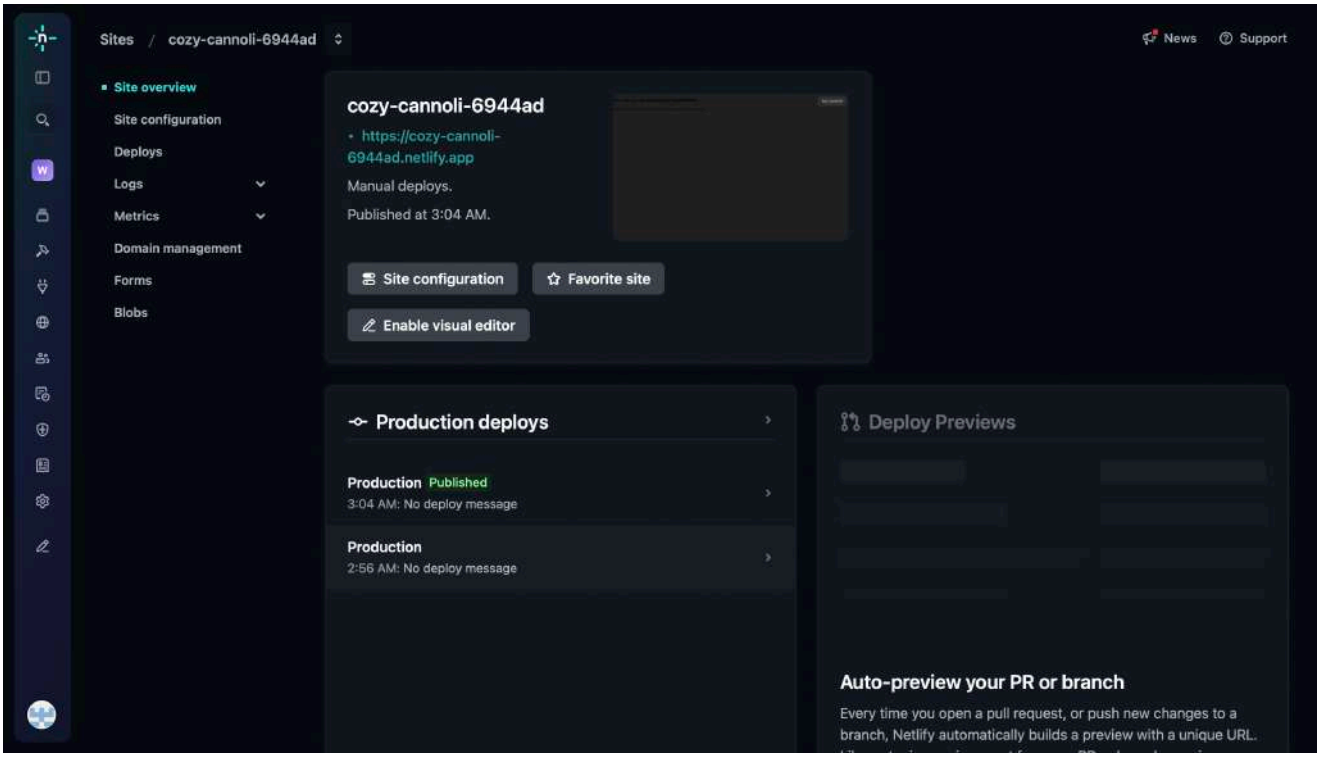


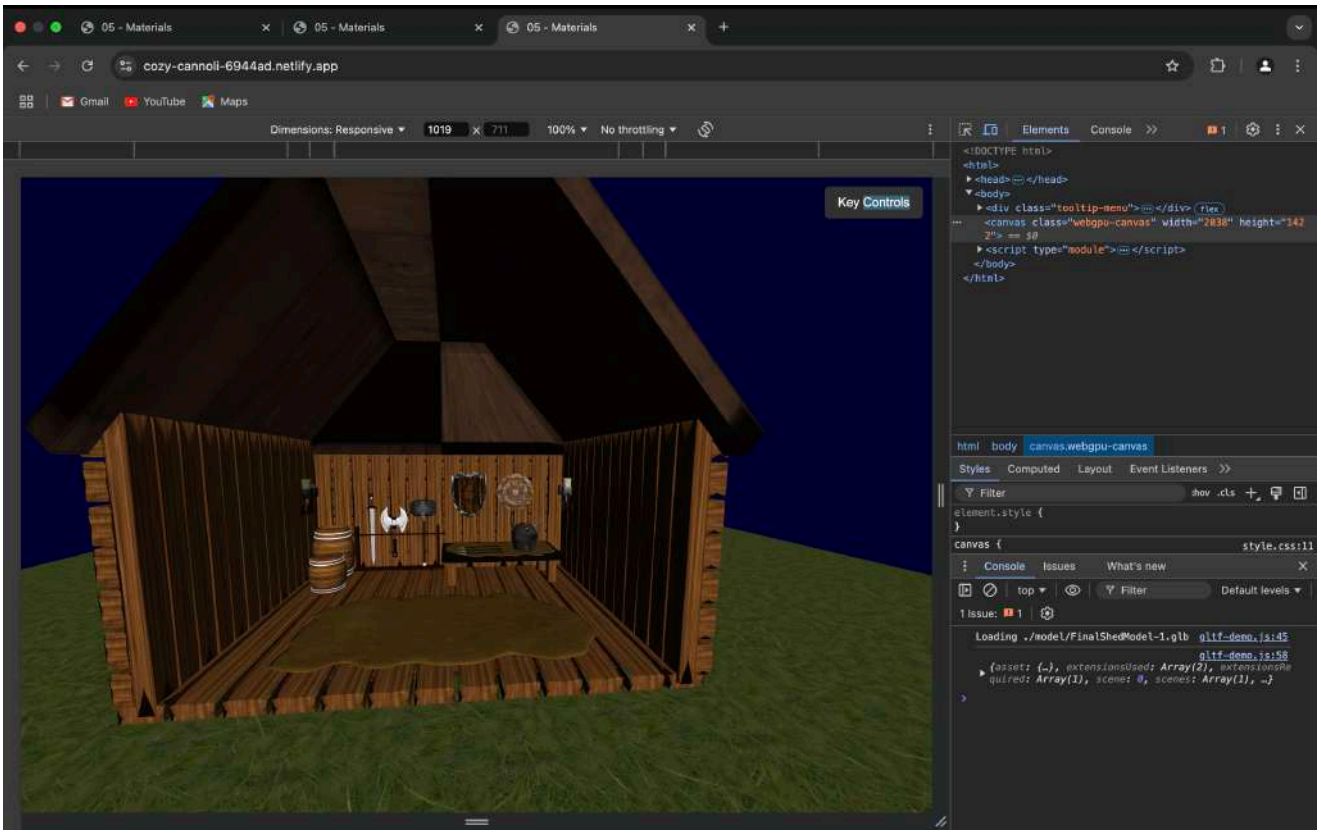
Vercel:
I set up a vercel x github workflow however on vercel there exists a limit of 100MB and so i

wasn't able to host my model there



Finally I tried it out on Netilfy and it worked out!




05 - Materials
05 - Materials
05 - Materials
cozy-cannoli-6944ad.netlify.app
Gmail
YouTube
Maps
Dimensions: Responsive
1019
711
100%
No throttling
Key Controls
Elements
Console
<!DOCTYPE html>
<html>
<head></head>
<body>

<canvas class="webgpu-canvas" width="2838" height="1422"> == $0
<script type="module"></script>
</body>
</html>
html
body
canvas.webgpu-canvas
Styles
Computed
Layout
Event Listeners
Filter
element.style {
}
canvas {
style.css:11
Console
Issues
What's new
top
Filter
Default levels
1 Issue:
Loading ./model/FinalShedModel-1.glb
gltf-demo.js:45
gltf-demo.js:58
{asset: {…}, extensionsUsed: Array(2), extensionsRequired: Array(1), scene: 0, scenes: Array(1), …}