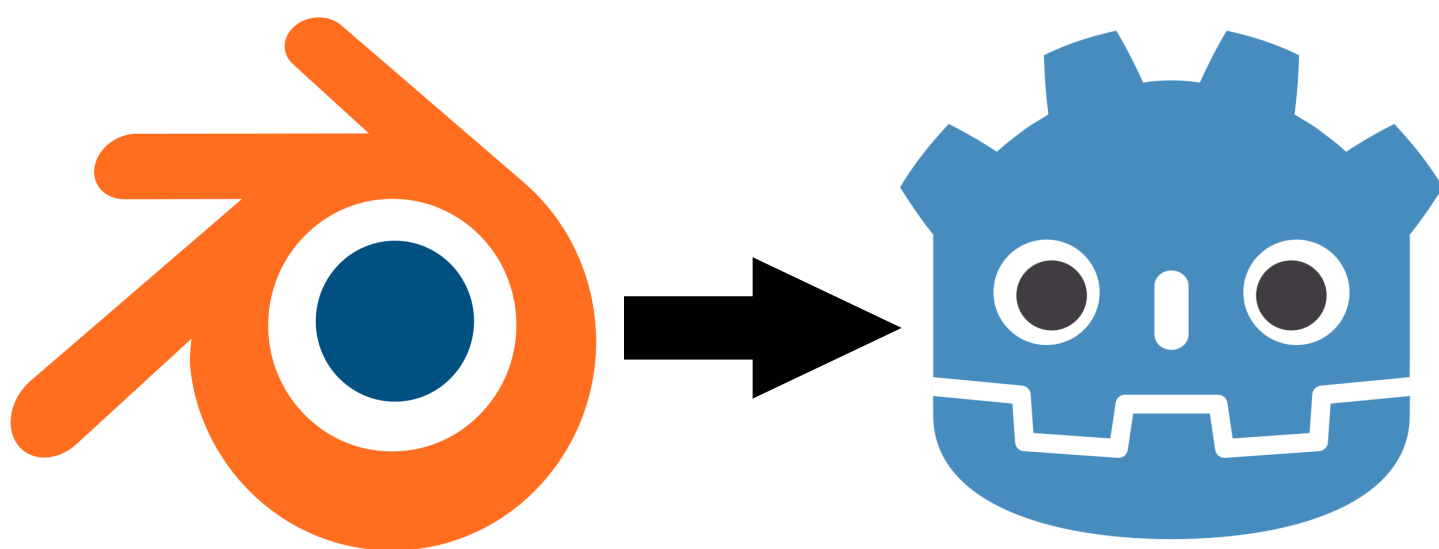


# Guide d'Importation d'Objets Blender à Godot



<b>1. Notes Préliminaires.....</b>	<b>3</b>
<b>2. Applications Nécessaires.....</b>	<b>3</b>
<b>3. De Blender à Godot.....</b>	<b>4</b>
3.1 Blender et .glb.....	4
3.2 Godot et .glb.....	5
3.3 Godot, .tscn et .scn.....	7

# 1. Notes Préliminaires

Les tests effectués pour ce guide ont été réalisés avec *Godot 4.4.1* et *Blender 4.4.3* ainsi que 4.0 sur *Windows 10*, en général toute version de *Godot* supérieure à 4.0 et toute version future de *Windows* (*Windows 11*) et de *Blender* devrait fonctionner pour ce guide. Il y a aura possiblement un système de compatibilité qui sera réalisé entre les 2 logiciels qui permettra d'obtenir directement les objets modélisés sur *Blender* en un clic dans *Godot*, mais ce n'est pour l'instant pas le cas. Nous faisons donc ce guide au cas où.

Il y a plusieurs méthodes afin d'importer des objets créés sur *Blender* directement dans *Godot*. Dans ce guide nous allons décrire la méthode que nous avons utilisée tout en expliquant les avantages et les inconvénients de cette méthode par rapport aux autres.

## 2. Applications Nécessaires

- **Blender** (<https://www.blender.org/download/>)
- **Godot 4.0+** (<https://godotengine.org/download/windows/>)

## 3. De Blender à Godot

### 3.1 Blender et .glb

Une fois notre objet modélisé sur Blender avec les bonnes textures et les bonnes UVs map, nous allons dans la fenêtre “File” puis “Export” afin de trouver le paramètre “glTF 2.0”, qui permet d’exporter notre objet en **.glb**. GLB est la forme binaire de glTF (GL Transmission Format), un format de modèles 3D utilisant le format JSON. Sa particularité est qu’il inclut notamment les textures plutôt que de les référencer en tant que fichiers externes. Ce qui permet d’avoir toutes les informations d’un objet dans un seul fichier plutôt que d’avoir à manipuler un dossier contenant 10 fichiers différents.

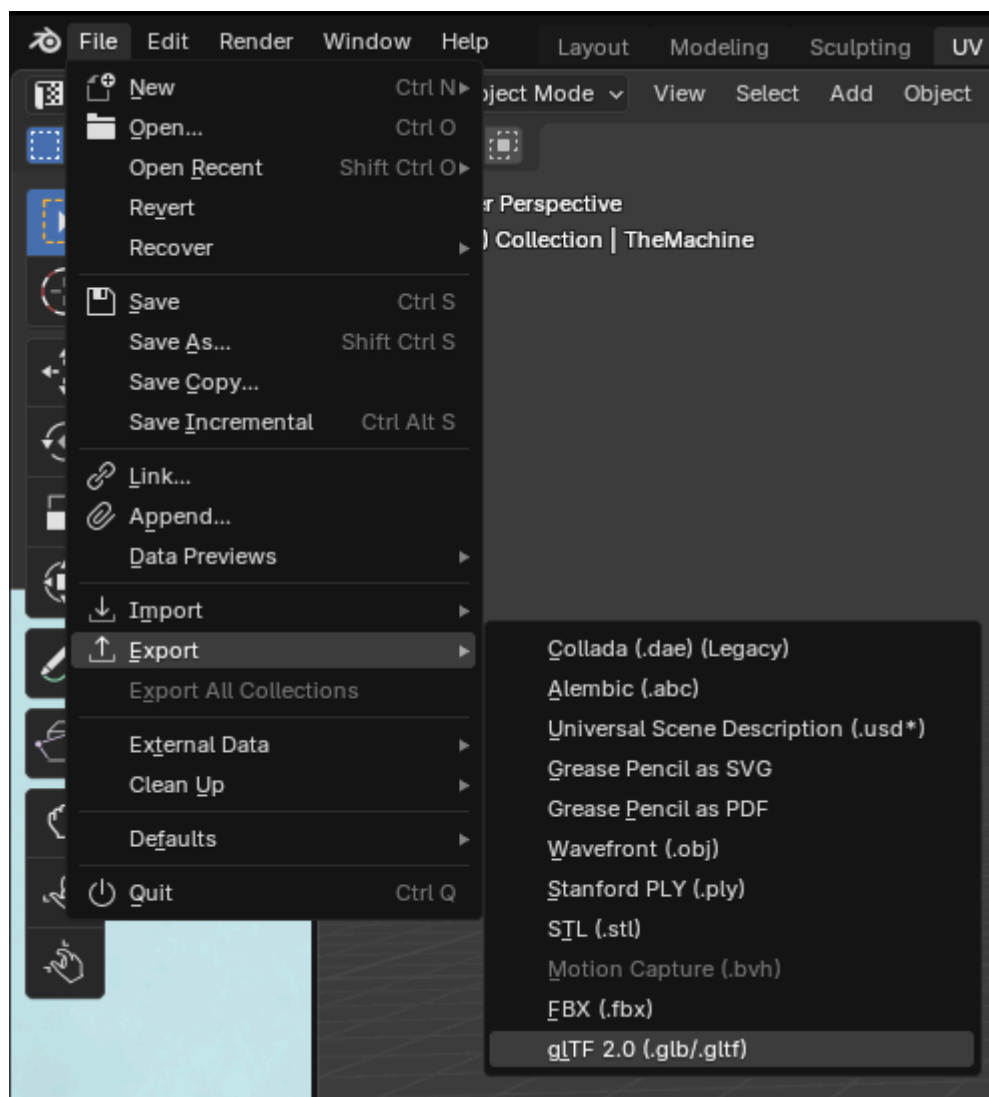


Figure 1 : Export as glTF 2.0

### 3.2 Godot et .glb

Une fois le fichier .glb obtenu, il suffit de l'importer dans Godot. Soit en l'enregistrant directement dans les fichiers de projet Godot, soit en le droppant dans les fichiers de Godot une fois celui-ci lancé. Cela crée automatiquement un .glb contenant les objets en tant que tel et l'ensemble des autres fichiers définissant l'objet : les textures PNG/WEBP sont séparées, de même pour les animations et matériaux qui sont convertis en ressources.

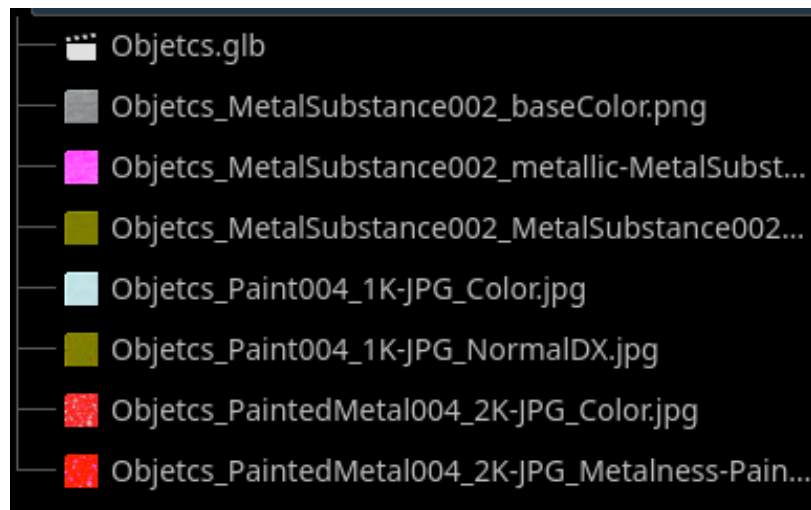


Figure 2 : Exemple de résultat après avoir drop un .glb dans Godot

Une fois ceci de fait, vous pouvez prendre le .glb et le lâcher dans la fenêtre de visualisation 3D de Godot, ce qui vous montrera directement vos objets en 3D.

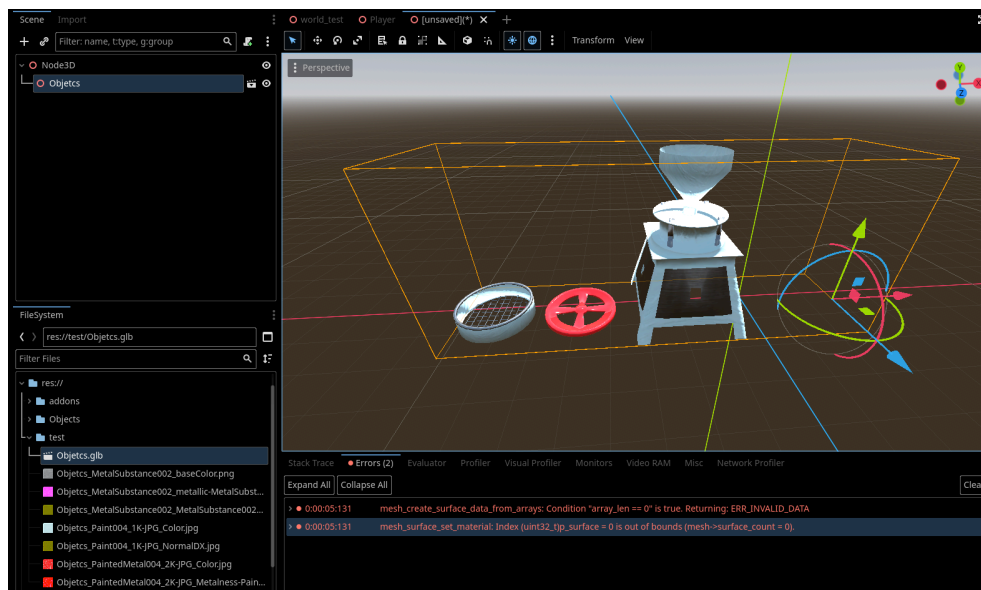


Figure 3 : Exemple de résultat après avoir lâcher un .glb dans le 3D Viewport

Maintenant vous devez cliquer droit sur le nouveau nœud qui a été créé (ici appelé "Objects") et sélectionner "Make Local". Cela créera autant de MeshInstance3D qu'il y a d'objets différents dans votre .glb. Chacune de ces MeshInstance3D contiennent les mêmes informations que les objets originaux du fichier de départ .glb.

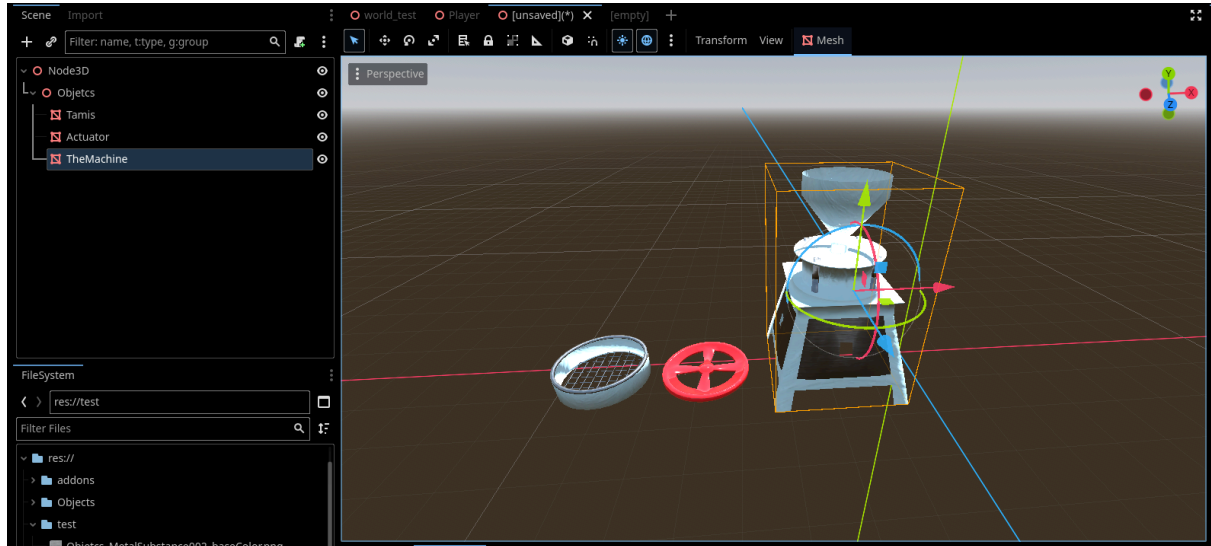


Figure 4 : Continuation de l'exemple

### 3.3 Godot, .tscn et .scn

Il ne vous reste maintenant plus qu'à enregistrer chacun des objets comme une scène différente afin de pouvoir les placer comme bon vous semble dans votre jeu. Pour cela vous devez cliquer droit sur un objet en particulier et cliquer sur "Save Branch as Scene". Vous pouvez aussi simplement créer une nouvelle scène et "CTRL + C" votre objet puis le "CTRL + V" dans la nouvelle scène.

Vous pouvez enregistrer les objets sous format .scn et .tscn. Le .scn est un format binaire qui est plus rapide à charger pour Godot qu'un format .tscn mais étant donné qu'il est binaire il est très compliqué de le lire d'un point de vue humain. Dans notre cas nous sommes restés sur du .tscn, mais vous pouvez les enregistrer au format .scn si vous le souhaitez, cela n'est pas forcément très grave mais l'optimisation est toujours intéressante.

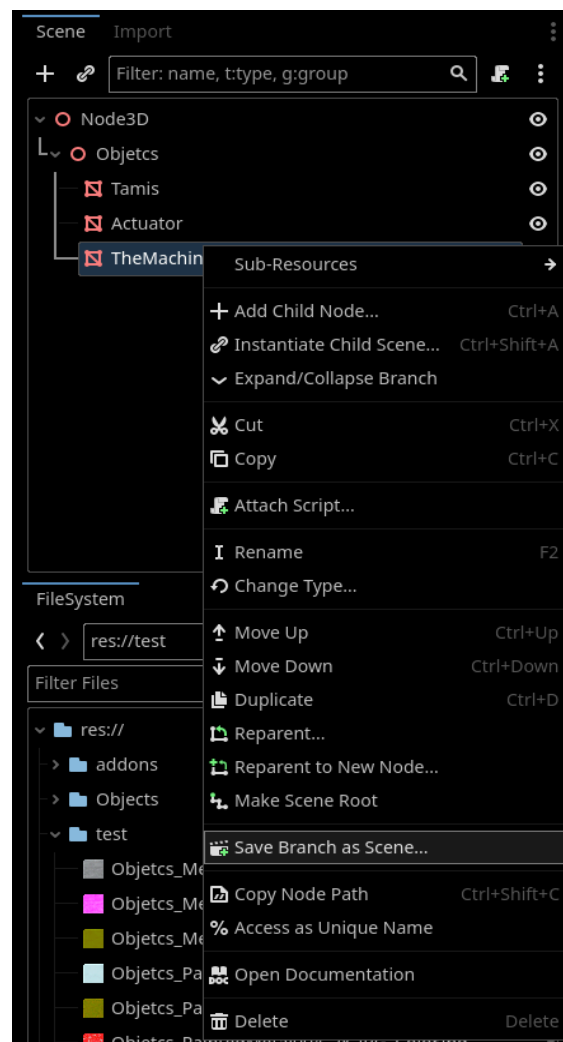


Figure 5 : Save Branch as Scene

Une fois tous les objets enregistrés, vous pouvez supprimer le .glb : il ne contient plus que des informations redondantes.

De plus, vous pouvez déplacer les fichiers comme vous le souhaitez, ceux-ci sont toujours liés entre eux. Vous pouvez par exemple mettre les PNGs dans un dossier “Textures” et mettre tous les objets dans des fichiers séparés, ceux-ci ne perdront pas leurs textures. Godot changera les liaisons automatiquement (comparé à d’autres logiciels de game design qui ne disposent pas de cette fonctionnalité très simple, comme UE5).

Voilà, vous avez désormais vos objets modélisés depuis Blender directement prêt à être utilisé en jeu sur Godot.