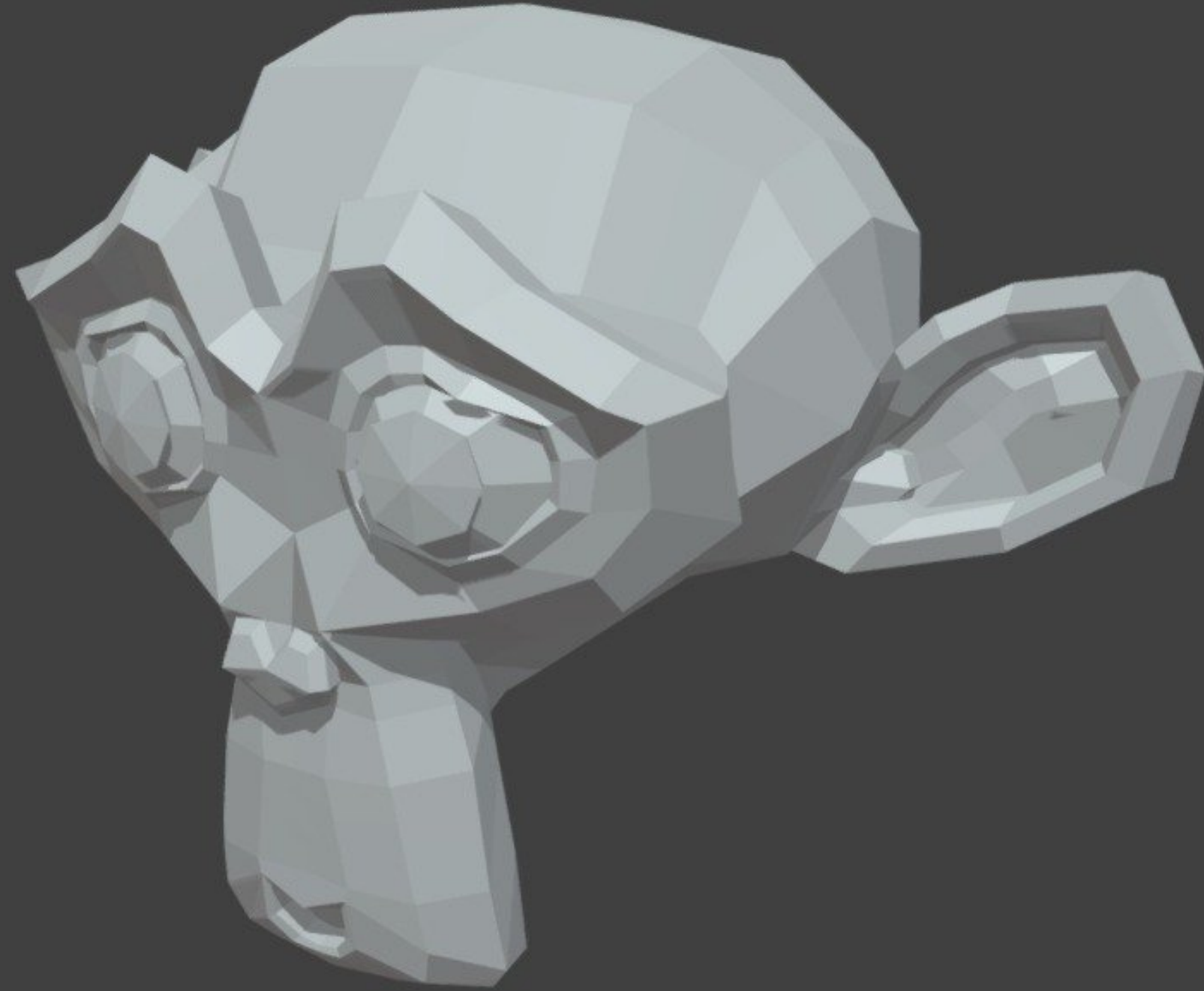


SNCF



SCRIPT BLENDER

Cahier des charges

Vérifier la syntaxes des textures

.....

Vérifier la présences de N-gones

.....

Afficher un message d'avertissement

.....

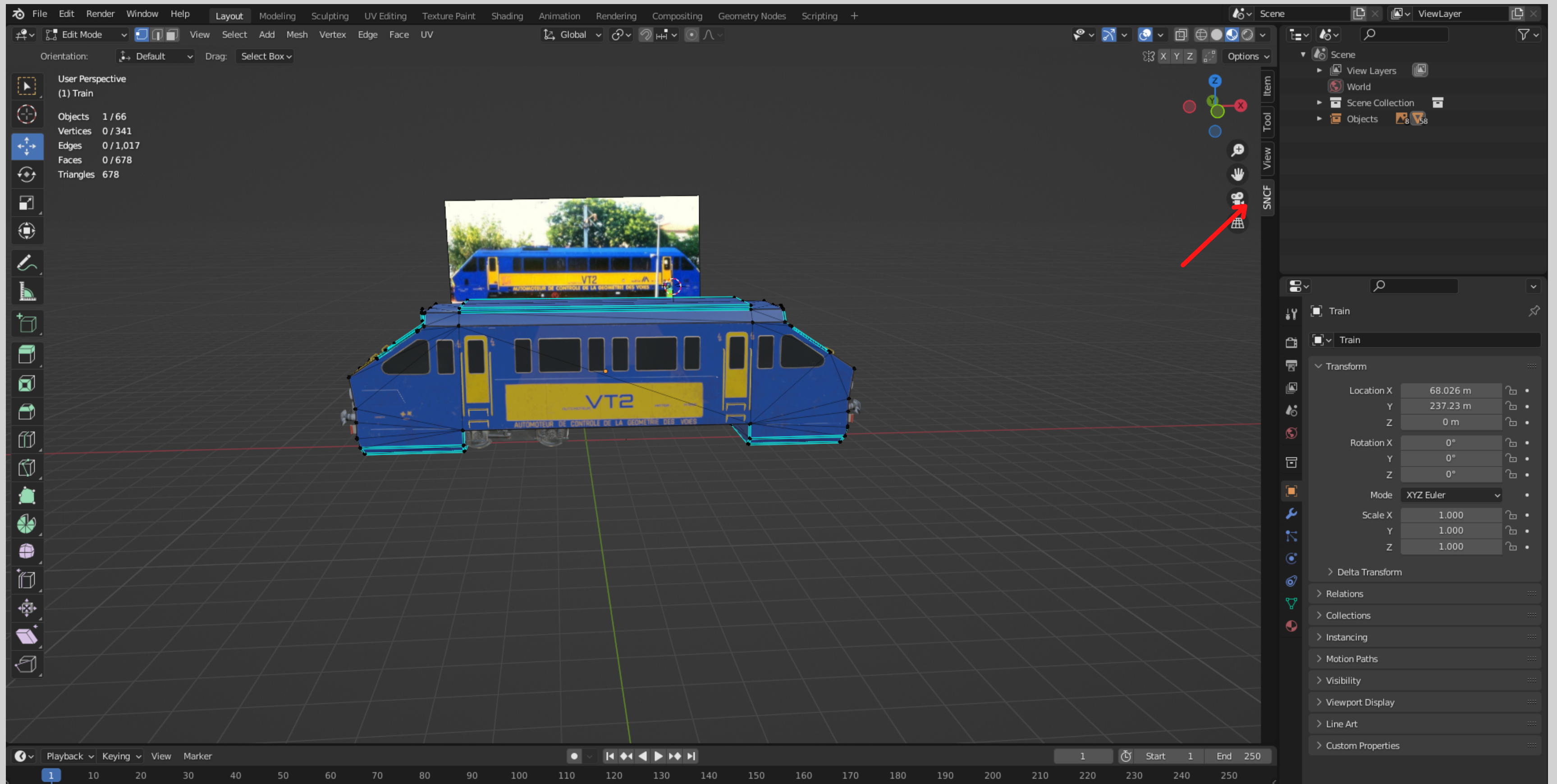
Ne pas être une action bloquante

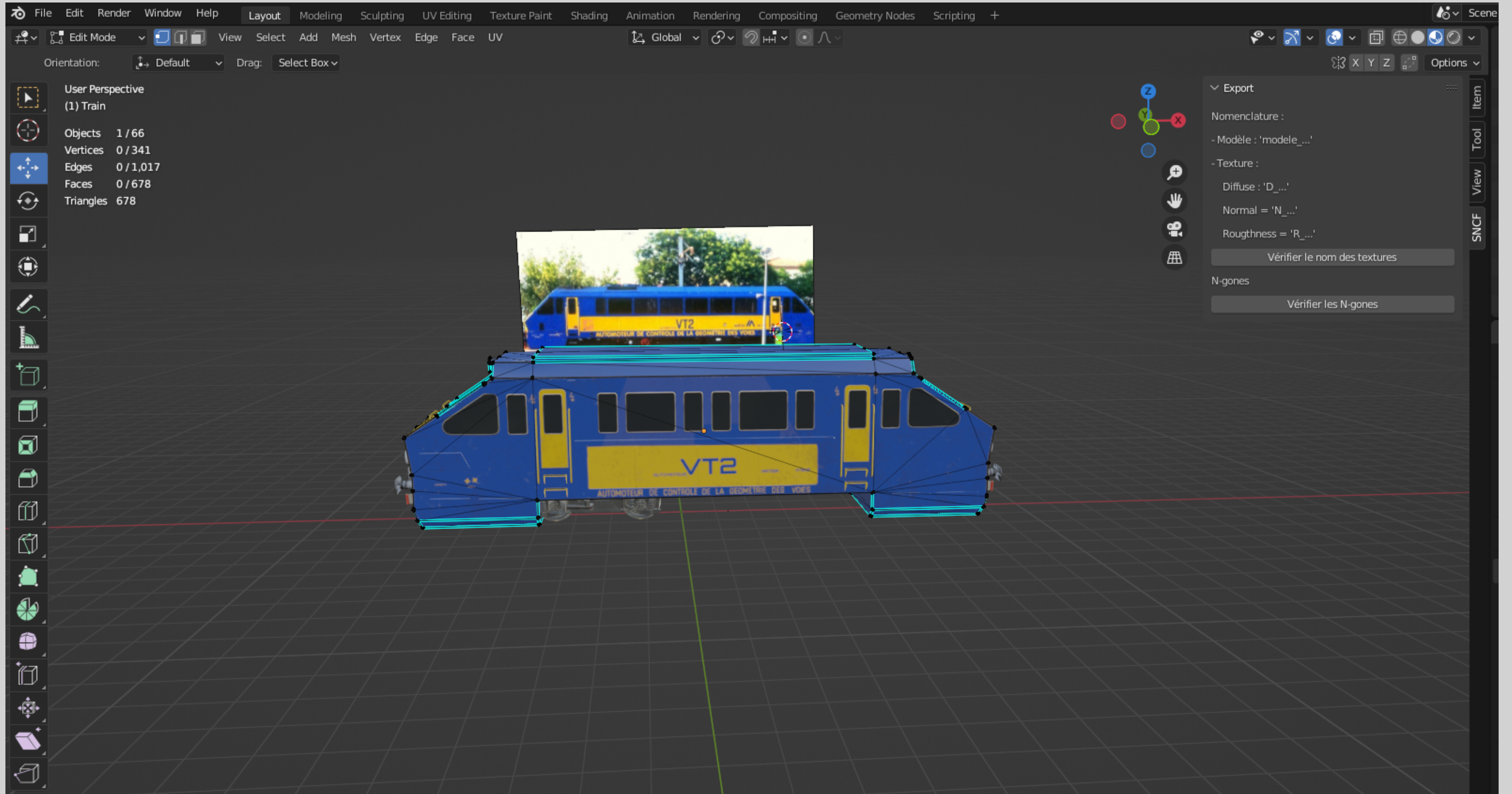
.....

Créer des export type au format FBX

.....

L'interface Blender complet



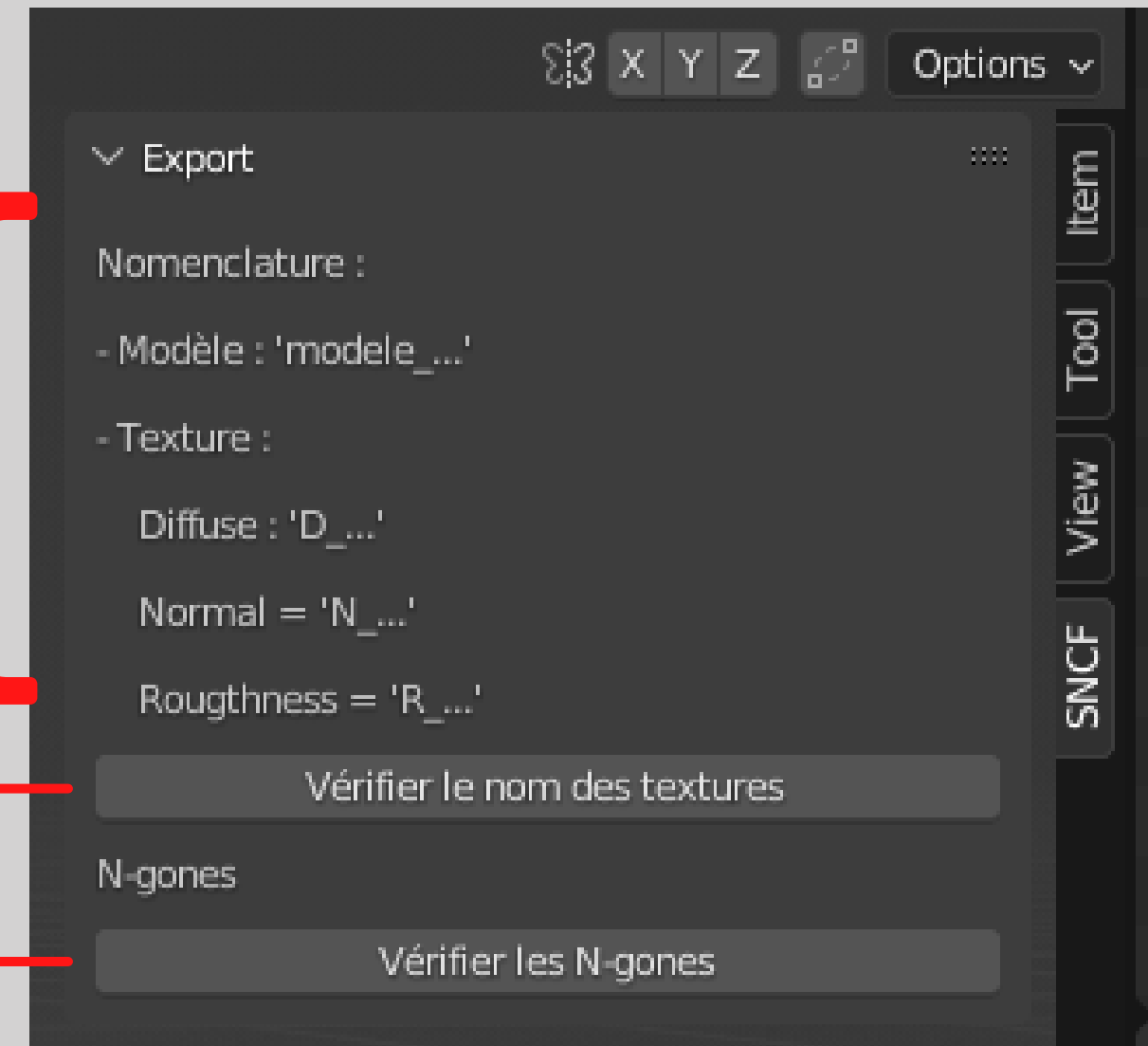


Onglet SNCF

Rappel de la Nomenclature attendu

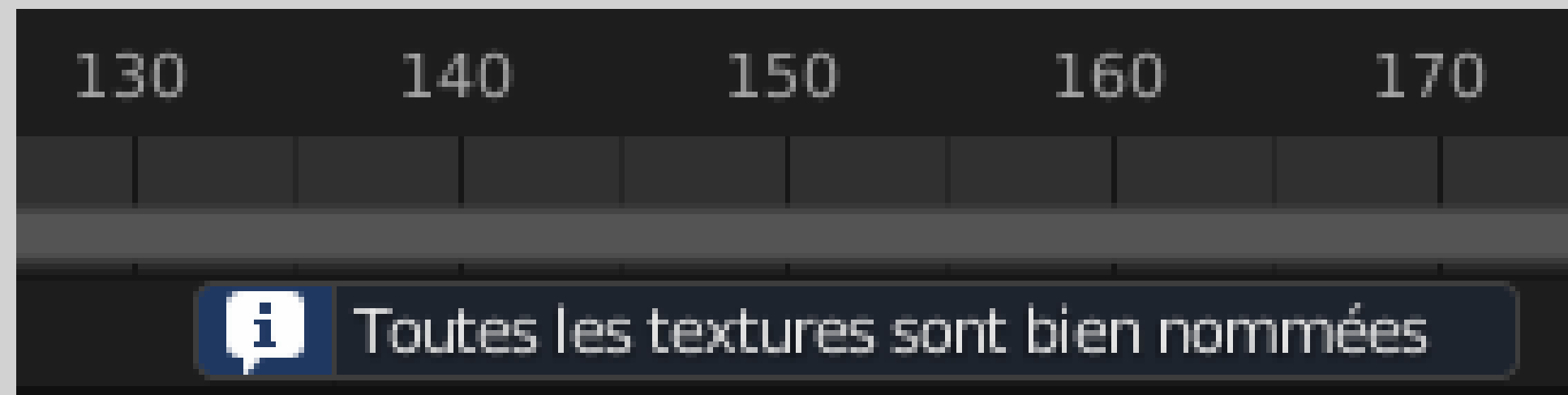
Vérifier que les textures sont bien nommées

Vérifier la présence de N-gones



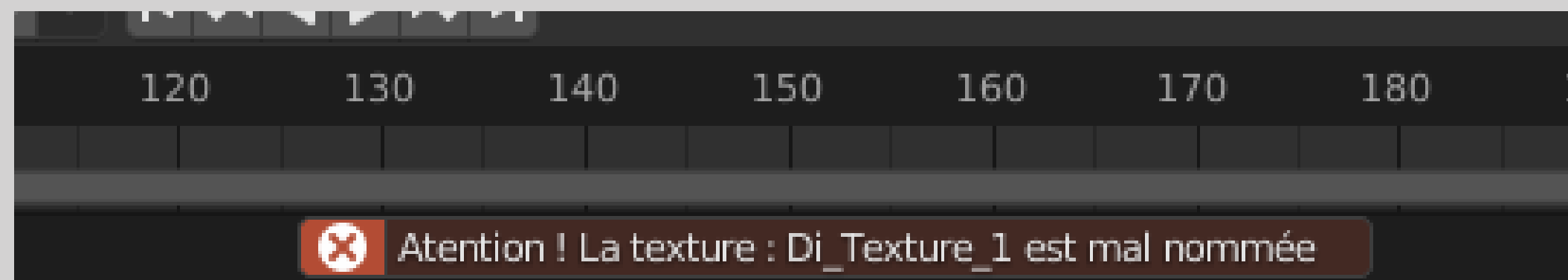
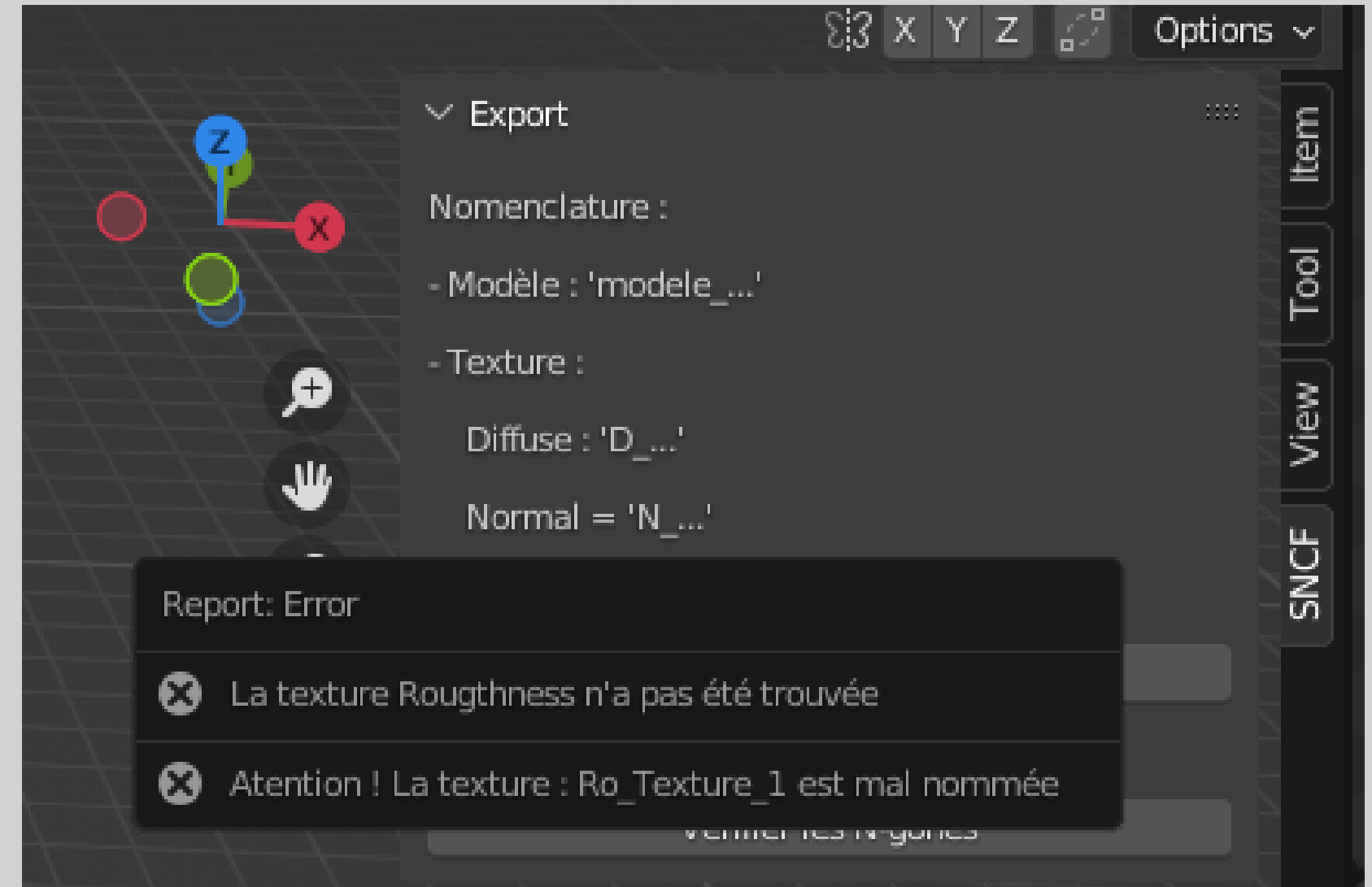
Lorsque toutes les textures sont bien nommées

- Un message apparaît pour le signaler



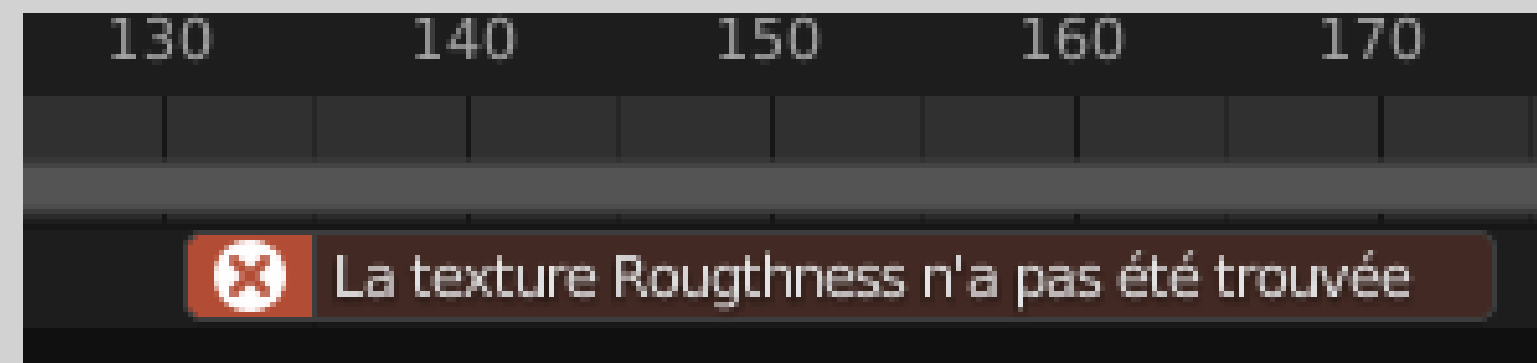
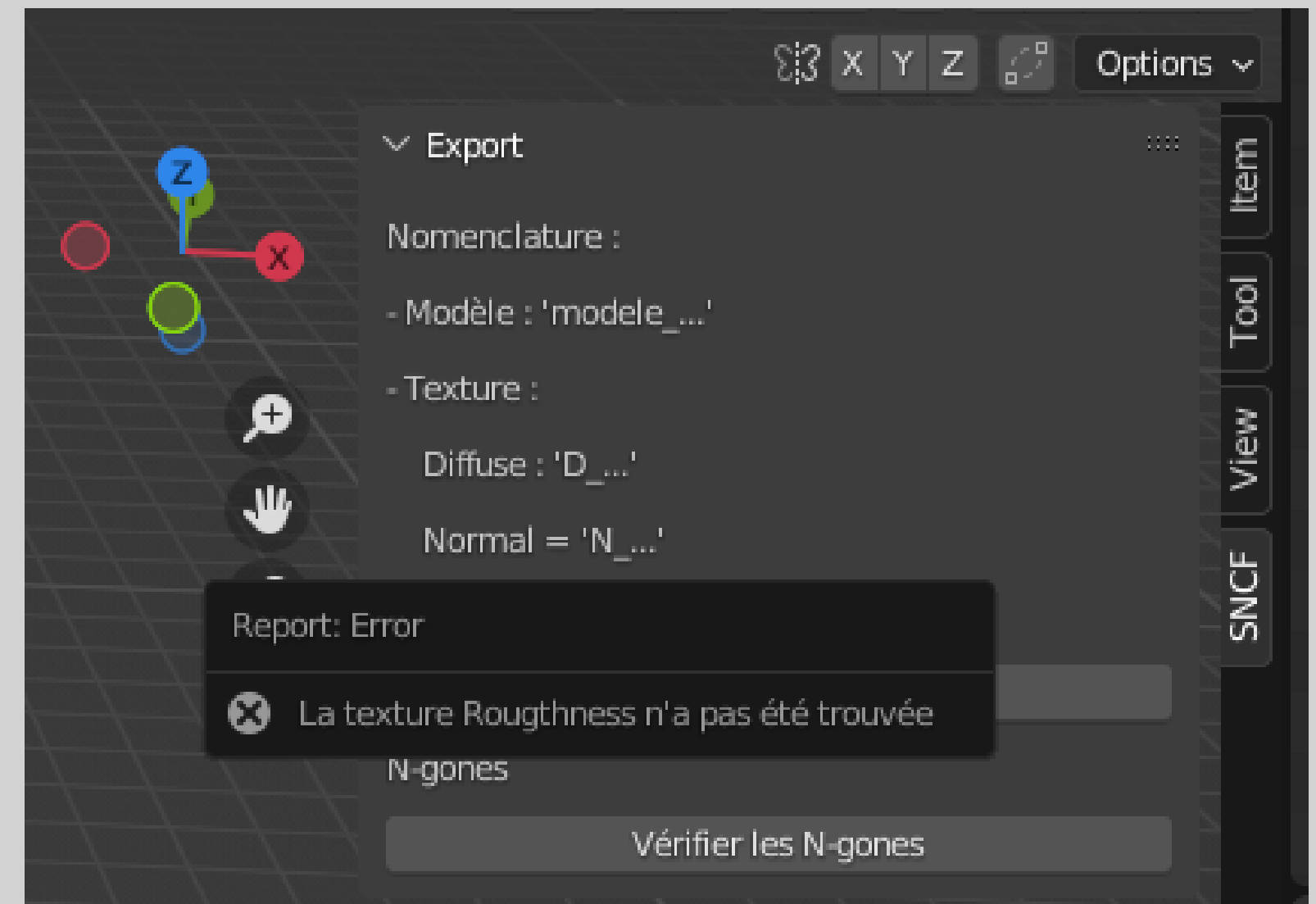
Lorsqu'une texture est mal nommée

- 2 messages apparaissent pour le signaler à l'utilisateur
- il n'est pas bloqué



Lorsqu'une texture n'est pas trouvée

- 2 messages apparaissent pour le signaler à l'utilisateur
- l'utilisateur n'est pas bloqué



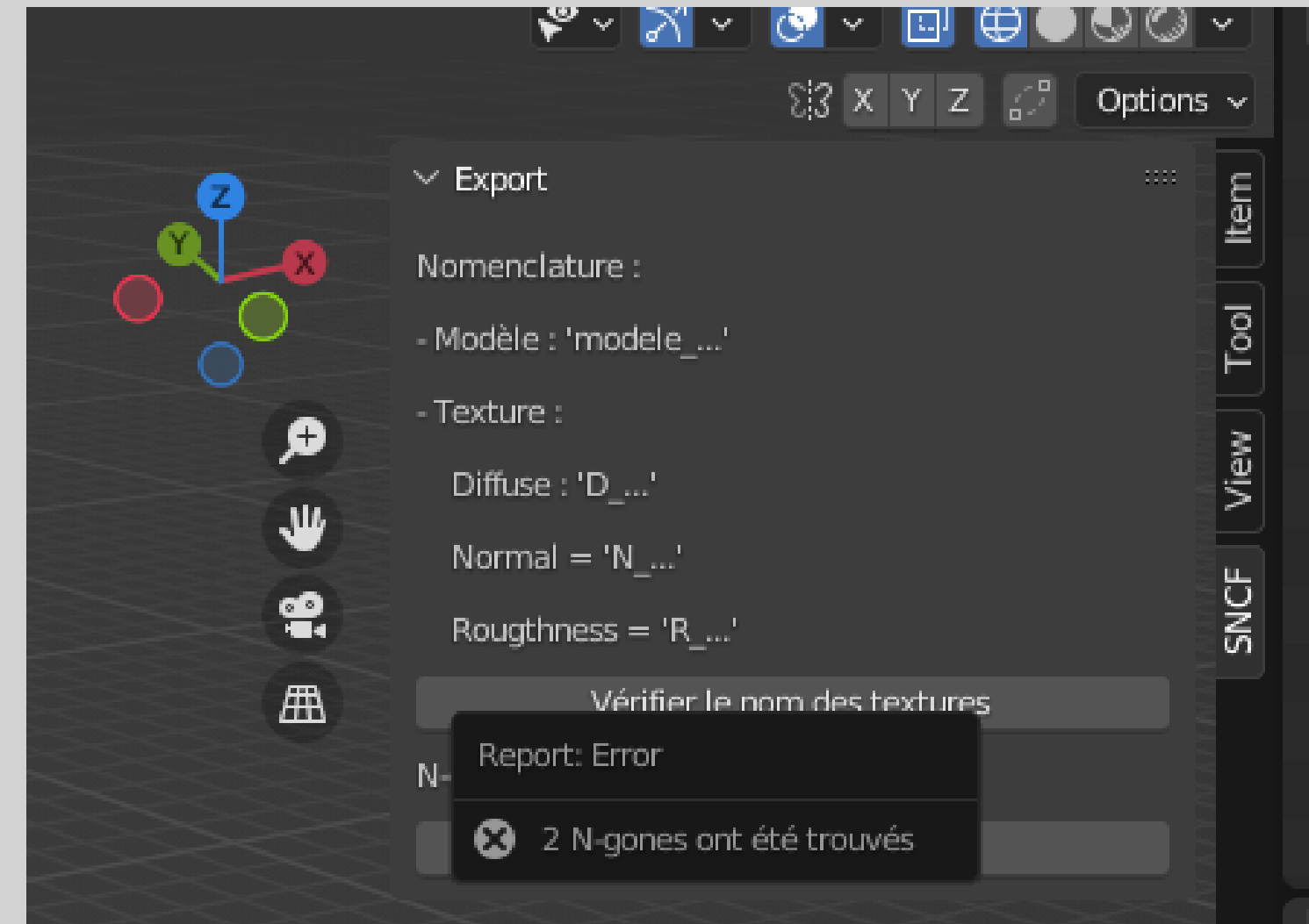
Lorsque qu'il n'y a pas de N-gones

- Un message apparaît pour le signaler



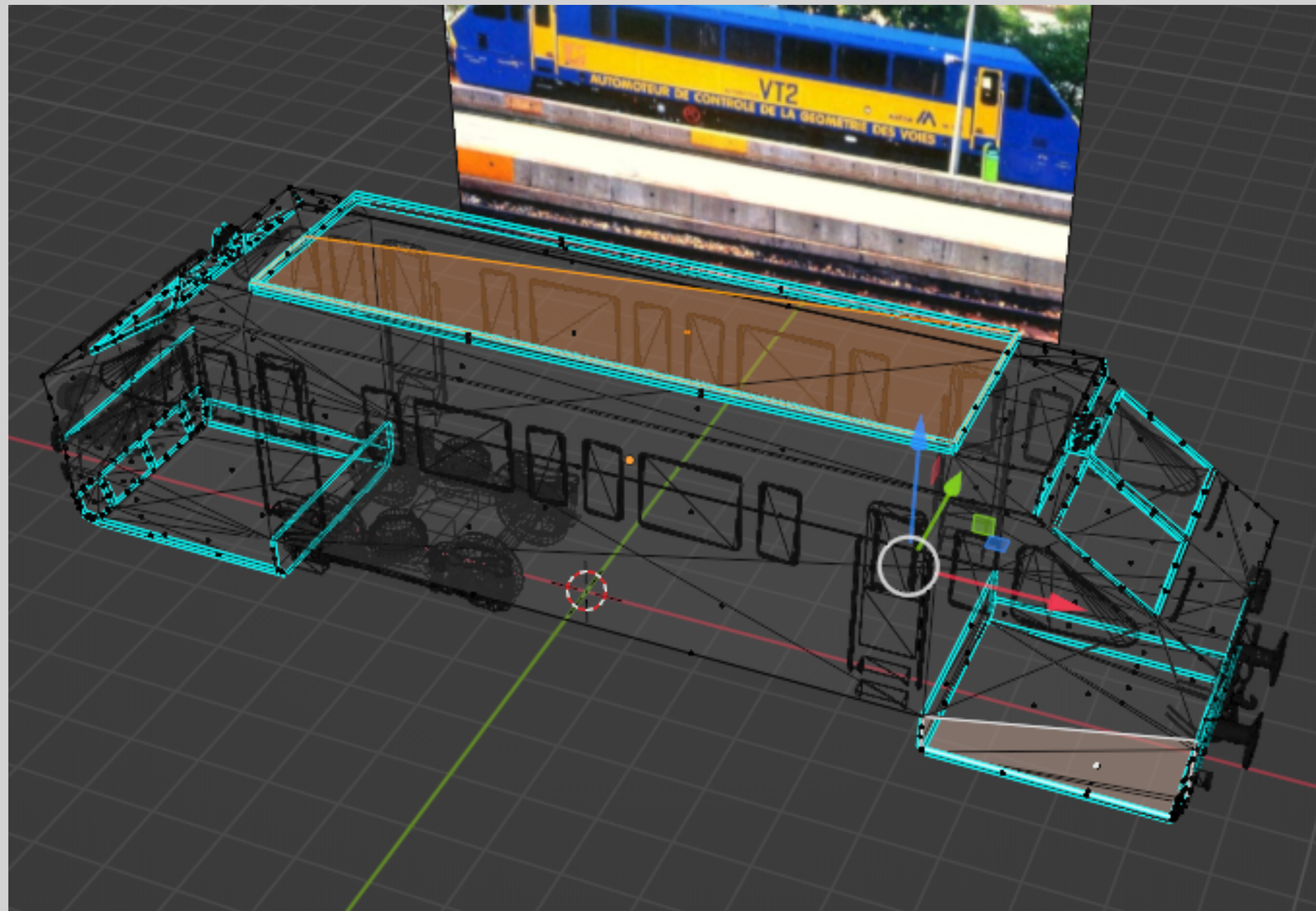
Lorsque des N-gones sont trouvés

- 2 messages apparaissent pour le signaler à l'utilisateur
- il n'est pas bloqué



Lorsque des N-gones sont trouvés

- Les N-gones s'affichent sur le mesh dans la View3D



Pour plus de lisibilité

- Mesh en Wireframe
- N-gones oranges

Export au format FBX

Finalelement non traités car perte de praticité

- Export Anim Operators
- Export Mesh Operators
- Export Scene Operators

```
bpy.ops.export_scene.fbx(filepath="", check_existing=True, filter_glob='*.fbx', use_selection=False, use_visible=False, use_active_collection=False, global_scale=1.0, apply_unit_scale=True, apply_scale_options='FBX_SCALE_NONE', use_space_transform=True, bake_space_transform=False, object_types={'ARMATURE', 'CAMERA', 'EMPTY', 'LIGHT', 'MESH', 'OTHER'}, use_mesh_modifiers=True, use_mesh_modifiers_render=True, mesh_smooth_type='OFF', use_subsurf=False, use_mesh_edges=False, use_tspace=False, use_triangles=False, use_custom_props=False, add_leaf_bones=True, primary_bone_axis='Y', secondary_bone_axis='X', use_armature_deform_only=False, armature_nodetype='NULL', bake_anim=True, bake_anim_use_all_bones=True, bake_anim_use_nla_strips=True, bake_anim_use_all_actions=True, bake_anim_force_startend_keying=True, bake_anim_step=1.0, bake_anim_simplify_factor=1.0, path_mode='AUTO', embed_textures=False, batch_mode='OFF', use_batch_own_dir=True, use_metadata=True, axis_forward='-Z', axis_up='Y')
```

Complexité

Lors de l'exécution de "vérifier le nom des textures"

```
def allTexture(self):  
    textureDict = createTextureDict()  
    if missingTexture():  
        for key in textureDict :  
            if textureDict[key] == '' :  
                self.report({'ERROR'}, message = "La texture " + key + " n'a pas été trouvé")  
            findWrongTexture(self)  
        else :  
            self.report({'INFO'}, message = "Toutes les textures sont bien nommées")
```

$O(3)$ ← $O(3)$

$O(3)$ ← $O(3)$

Au pire des cas : $O(4n) \sim O(n) \Rightarrow$ complexité linéaire

Dans notre cas il y a dans le pire des cas 12 opérations

Complexité

Lors de l'exécution de "vérifier les N-gones"

$O(n)$

$O(1)$

```
def ngon(self):  
    activateEditMode()  
    ngonList = createNgonList()  
    if len(ngonList) > 0 :  
        activeWireframe()  
        bpy.ops.mesh.select_face_by_sides(number=4, type='GREATER')  
        if len(ngonList) == 1 :  
            self.report({'ERROR'}, message = str(len(ngonList)) + " N-gone a été trouvé")  
        else :  
            self.report({'ERROR'}, message = str(len(ngonList)) + " N-gones ont été trouvés")  
    else:  
        self.report({'INFO'}, message = "Aucun N-gone n'a été trouvé")
```

$O(1)$

$O(1)$

Au pire des cas : $O(n) \Rightarrow$ complexité linéaire

Dans notre cas il y a au plus n opérations, avec n le nombre de polygone du modèle 3D

Liens du git et documentation

<https://github.com/leandre95/SNCF>

Documentation techniques sous formes d'un read me