



ARASHTAD

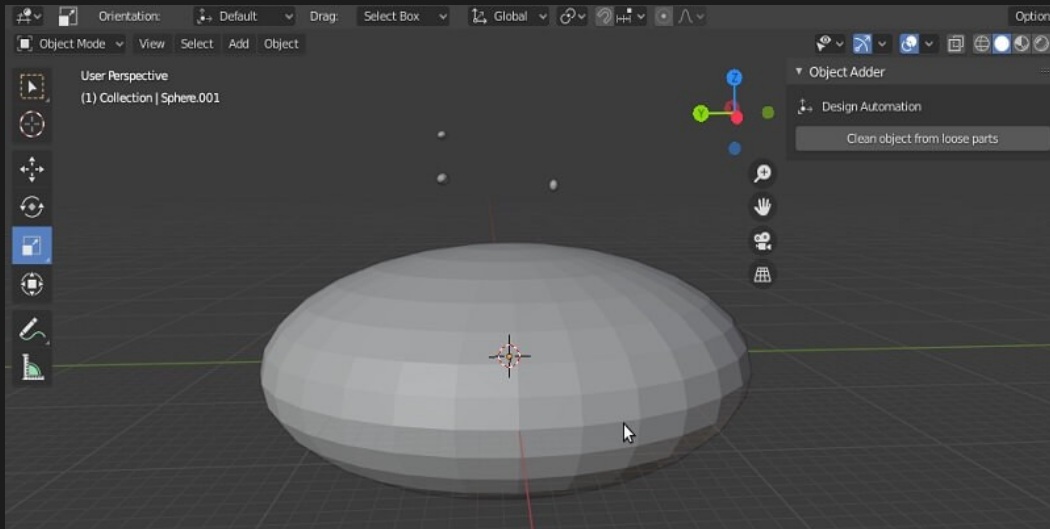
Title	A QUICK WAY TO AUTOMATICALLY REMOVE LOOSE PARTS
Description	Intro
Date	June 26, 2022
Author	Arashtad
Author URI	https://Arashtad.com



There are times that you face some unwanted extra objects all over your mesh such as when you get a raw 3d scan object that has been converted from point cloud to mesh. And, you want to quickly get rid of them all. In this tutorial, we are going to see how we can remove these loose parts in Blender and how we can create a button to quickly do that for us.

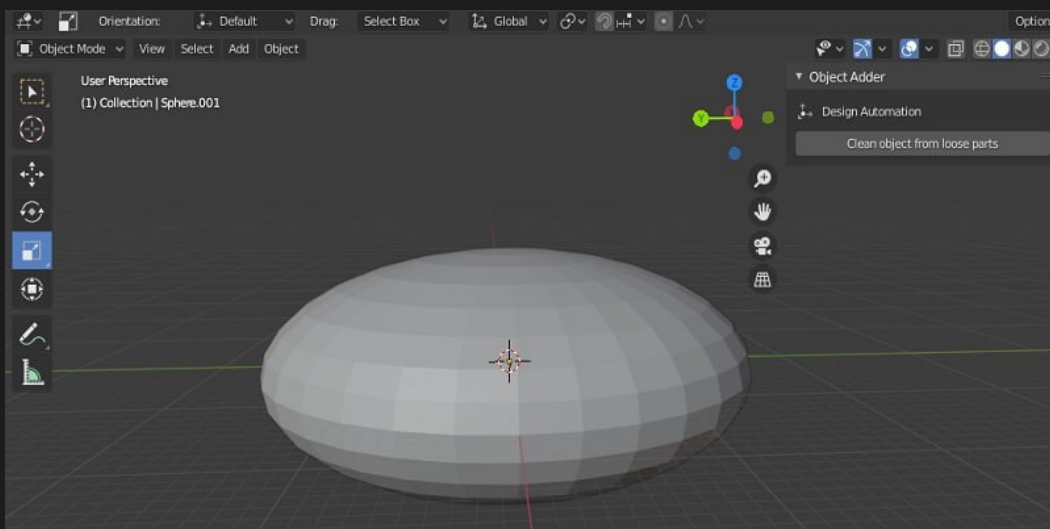
HOW TO REMOVE LOOSE PARTS IN BLENDER

There are times that you face some unwanted extra objects all over your mesh such as when you get a raw 3d scan object that has been converted from point cloud to mesh and you want to quickly get rid of them all. In this tutorial, we are going to see how we can get away with these small useless specks of dust using Blender python and how we can create a button to quickly do that for us.



IMPORTANT NOTE:

Remember that the scripts we are using here are related to Blender version 2.83 and if you are working with any other versions, it is probable that the scripts might differ a little bit, but we will show you ways to find the proper functions if there are any differences at all.



PYTHON SCRIPTS

First of all, we write our utility functions and then apply them in the main panel.

```
import bpy

#####
#####          Utility Functions
#####

def clean_from_loose_parts(obj):
    deselect_objects()

    # separate the mesh
    if obj.type == 'MESH':
        curr_obj_name = obj.name

        obj.select_set(True)
        bpy.ops.mesh.separate(type='LOOSE')

        # separated_objects = tuple(obj.name for obj in
bpy.context.selected_objects)
        separated_objects_count = len(bpy.context.selected_objects)
        print("Mesh object {} separated in to {}
object(s)".format(curr_obj_name,

        separated_objects_count))

    if separated_objects_count > 1:
        # remove mesh objects which have less than max vertices
        obj_max_vertices =
get_max_vertices_object_from_selection()
        obj_max_vertices.select_set(False)
        bpy.ops.object.delete()
        # restore input name of the object
        obj_max_vertices.name = curr_obj_name

    else:
        obj.select_set(False)
```

The above function will do the main job of removing small loose parts for us.

```
def get_max_vertices_object_from_selection():
    max_vert_count = 0
    obj_max_vertices = None
    for obj in bpy.context.selected_objects:
        vert_count = len(obj.data.vertices)
        if max_vert_count < vert_count:
            max_vert_count = vert_count
            obj_max_vertices = obj

    return obj_max_vertices
```

The above function will count the vertices of the selected objects and return the one that has the maximum amount of vertices.

```
def get_object_by_name(obj_name):
    assert obj_name in bpy.data.objects, "Error getting object by
name:{}".format(obj_name)
    obj = bpy.data.objects[obj_name]

    return obj
```

The above function will get the object by name and store it in a variable.

```
def deselect_objects():
    bpy.ops.object.select_all(action='DESELECT')
```

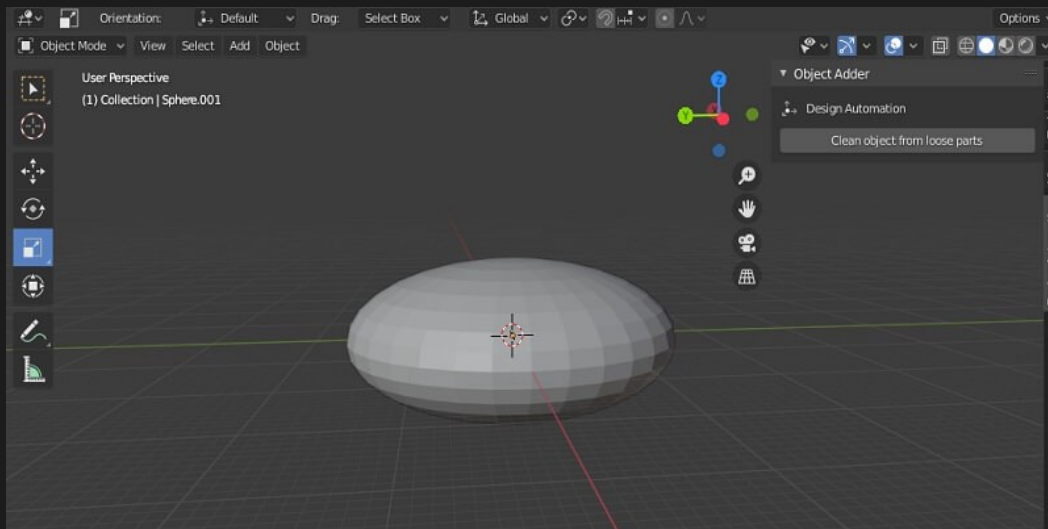



```
#####  
##### Register and Unregister  
#####  
  
def register():  
    bpy.utils.register_class(MainPanel)  
    bpy.utils.register_class(WM_Function_myOp)  
  
def unregister():  
    bpy.utils.unregister_class(MainPanel)  
    bpy.utils.unregister_class(WM_Function_myOp)  
  
if __name__ == "__main__":  
    register()
```

In the above main panel script, we use the utility function `clean_from_loose_parts(obj)`. The rest is related to the GUI of the panel.

TESTING THE SCRIPTS

Now, it is finally time to run our code using CTRL + P or the run button at the top of the scripting window and we will see that a panel appears with a button using which we can remove extra loose parts.



And as you can see, the small loose parts are removed. Consequently, we can have a clean object from noises. You can apply this tool to any raw 3D scan and make the edition of it much faster.



GETTING RID OF LOOSE PARTS IN BLENDER: WR

In this tutorial, we have managed to create a panel that can remove the noises and loose parts from all the objects that have them in Blender. Most of the time we face the issue of having loose parts when we receive a raw 3D scan from a scanner device. Having such a tool is very useful and saves a lot of time for you.

