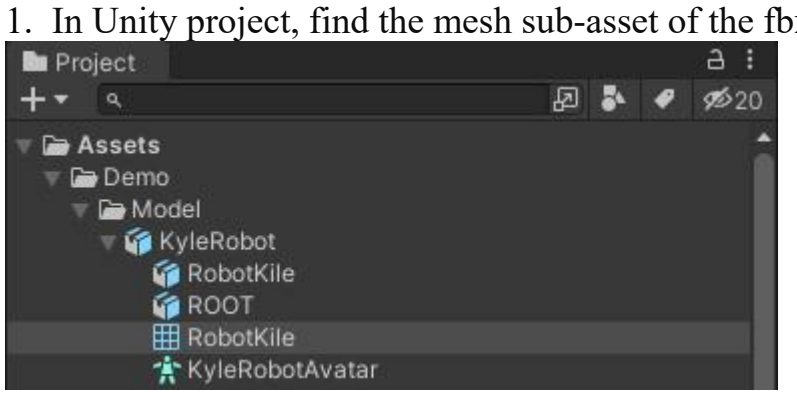
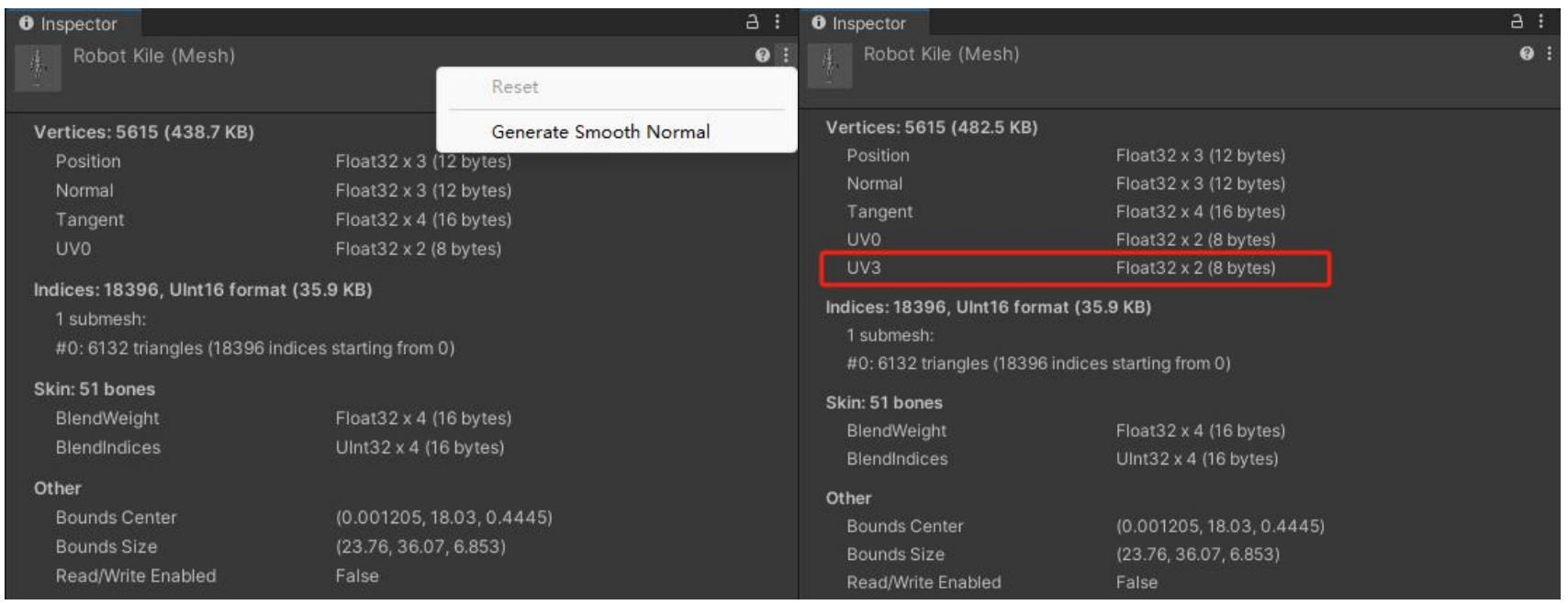
Model Outline Rendering: Smooth Normals Generator

Quick Start

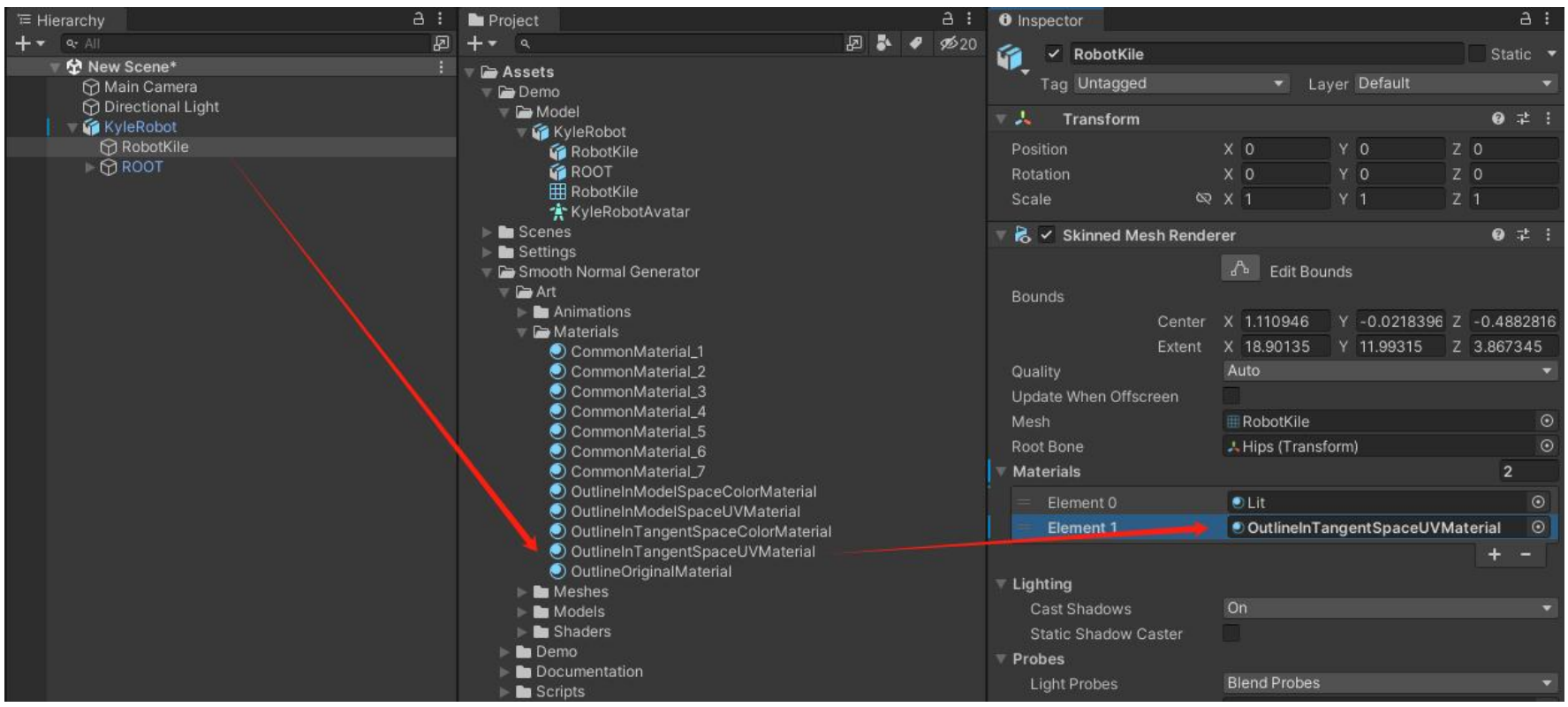
1. In Unity project, find the mesh sub-asset of the fbx asset file you want to outline.



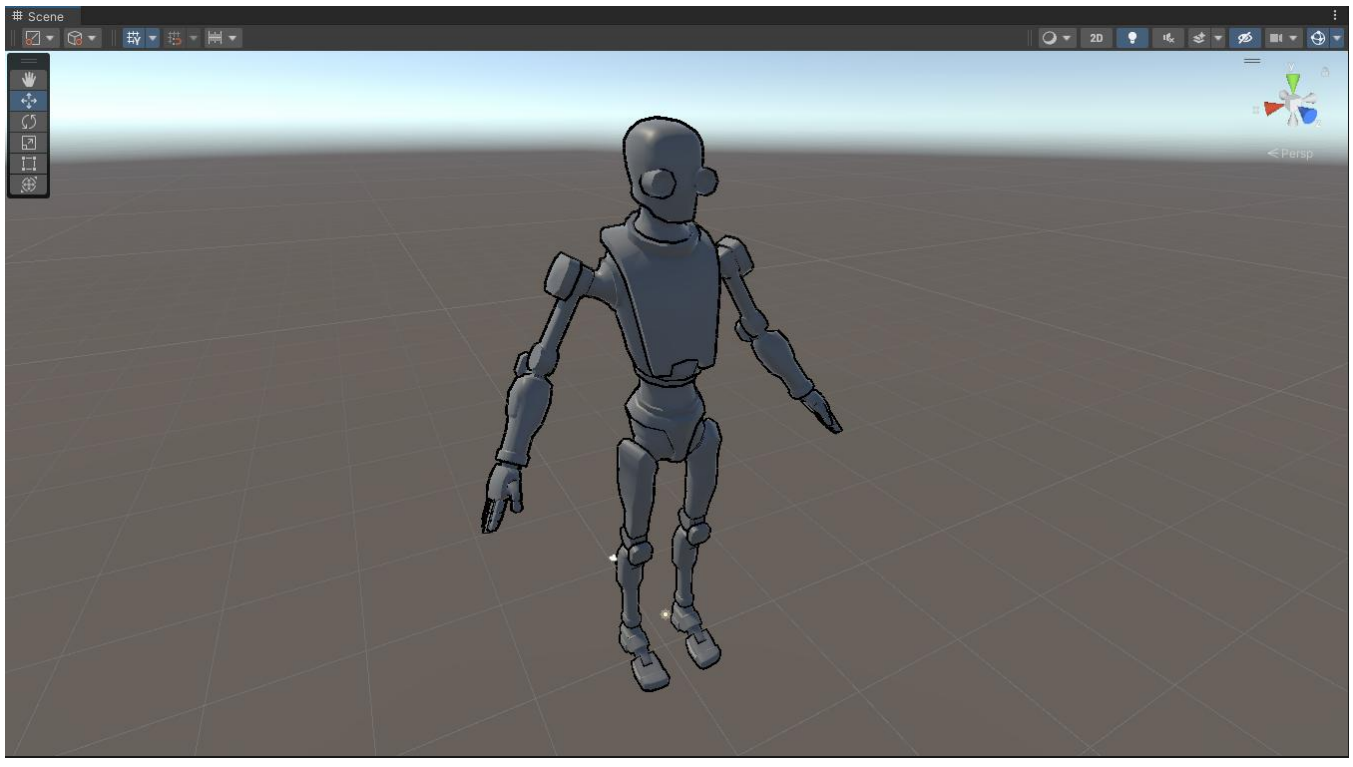
2. Click on the MenuItem (the button with three dots in the upper right corner of the Inspector), then click on "Generate Smooth Normals" in the popup menu (you will see smooth normals generated in UV3).



3. Drag the model into the scene, and then drag "OutlineInTangentSpaceUVMaterial" from "Smooth Normal Generator\Art\Materials" into the Materials of the Skinned Mesh Renderer.



4. Well Done



Advantages

1. Update FBX files with a single click, without the need to separately duplicate mesh files for individual processing, and without worrying about data loss when closing the project.
2. Supports generating smooth normals in both model space and tangent space. Smooth normals in tangent space are suitable for animated models. You can choose the UV channel to store the smooth normals, or they can be stored in the vertex color.
3. Supports least squares method for generating smooth normals, resulting in smooth normals that better adapt to extreme situations.
4. The plugin includes the MikkT algorithm, which automatically calculates tangent information when generating smooth normals, eliminating the need for tangent information to be included in the FBX file.

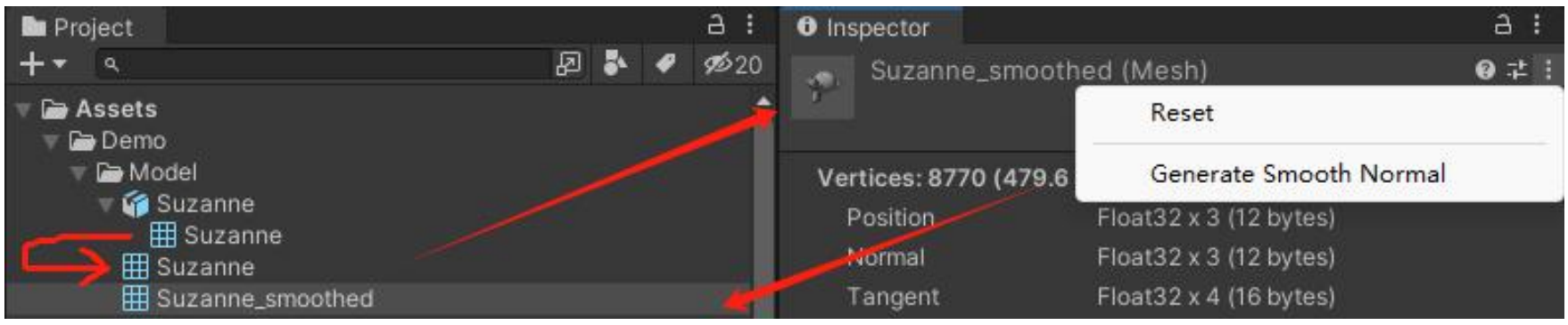
Introduction

1. Generate a duplicate of the FBX file with smoothed normals without directly overwriting it.

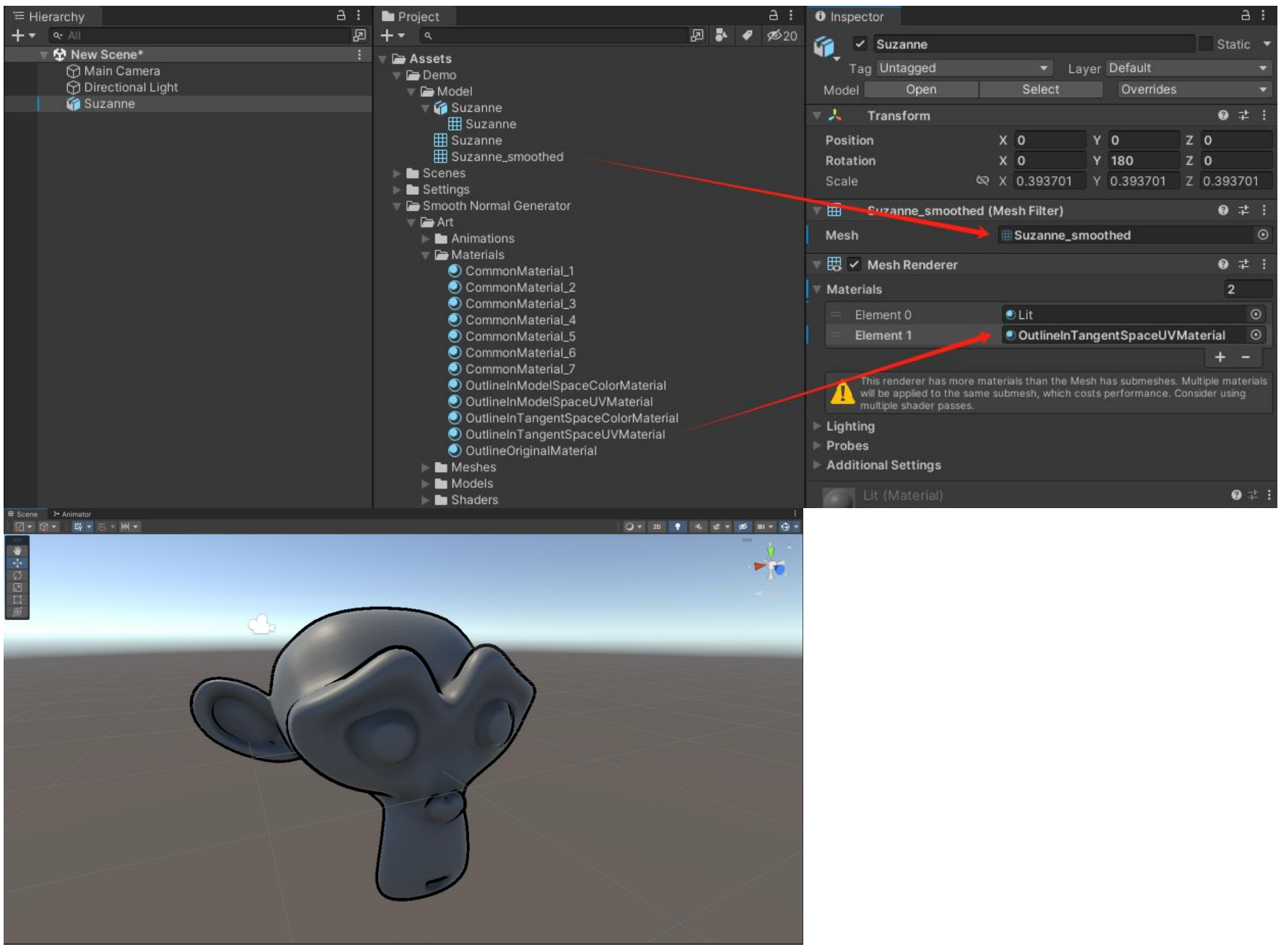
Open the Preferences panel, navigate to the Smooth Normal Generator tab, and set 'Is Overwrite Asset' to False. Repeat the steps from the quick start guide to obtain a duplicate of the FBX file with smoothed normals.

2. Generate smoothed normals for a single mesh file.

Simply copy the mesh from the FBX file, then generate smooth normals as in the quick start guide.

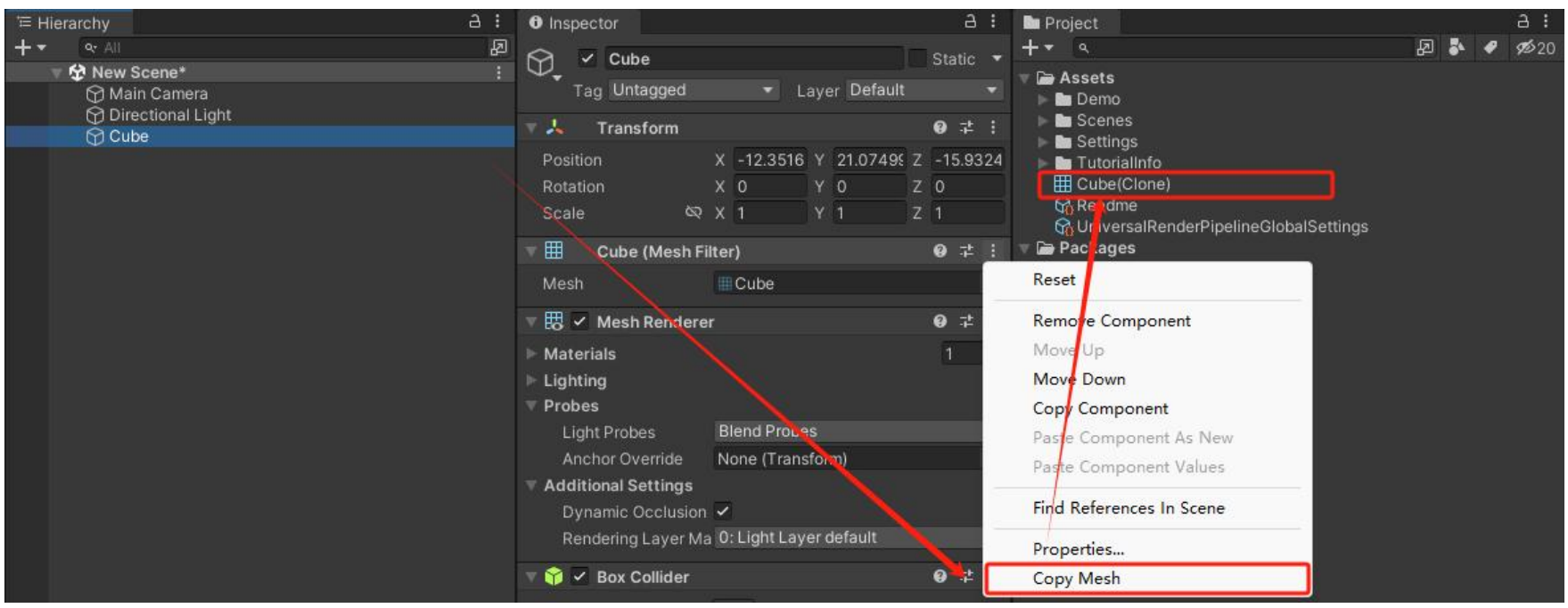


Drag modified mesh into MeshFilter, then drag the corresponding material into Mesh Renderer.

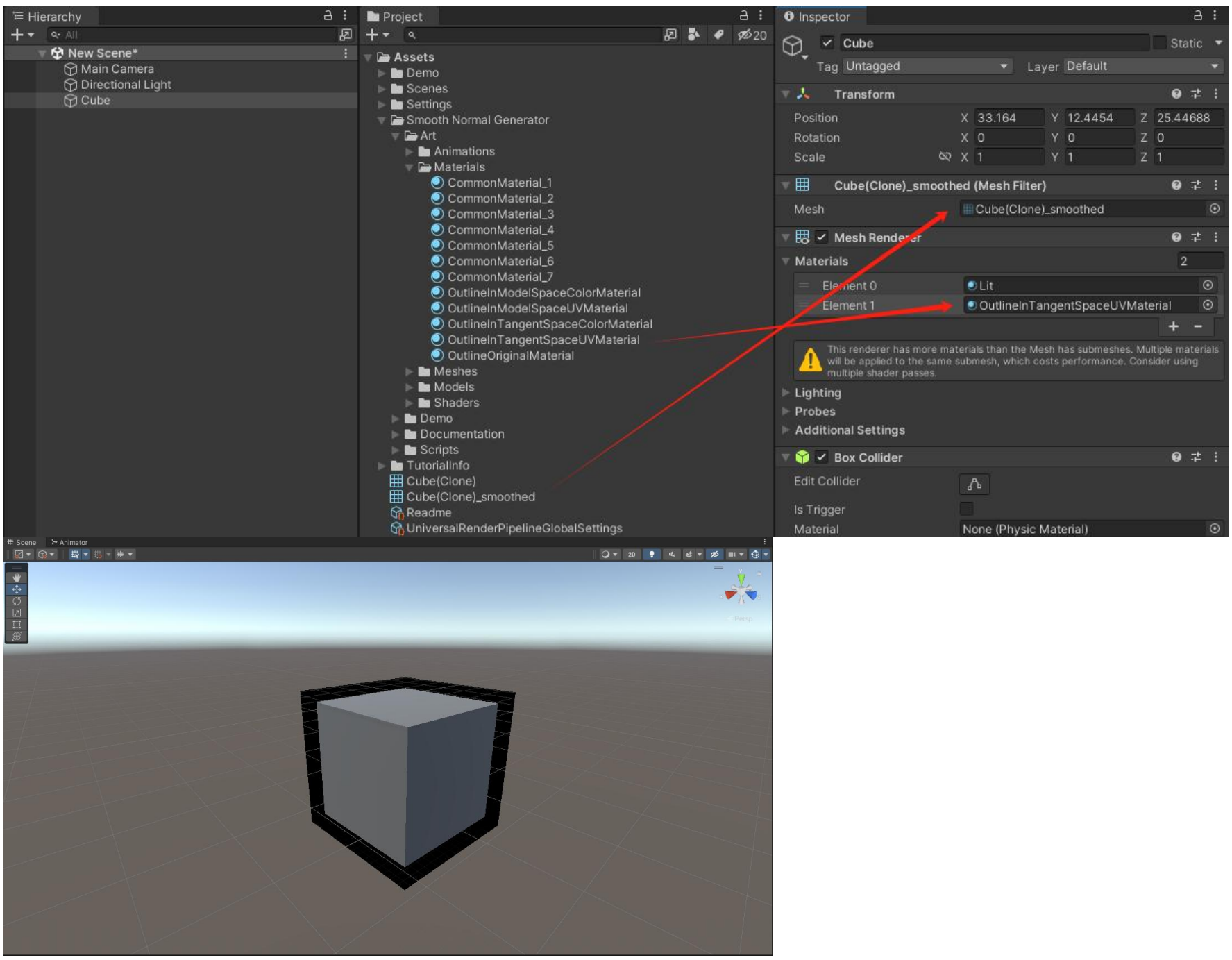


3. Generate smoothed normals for the built-in model.

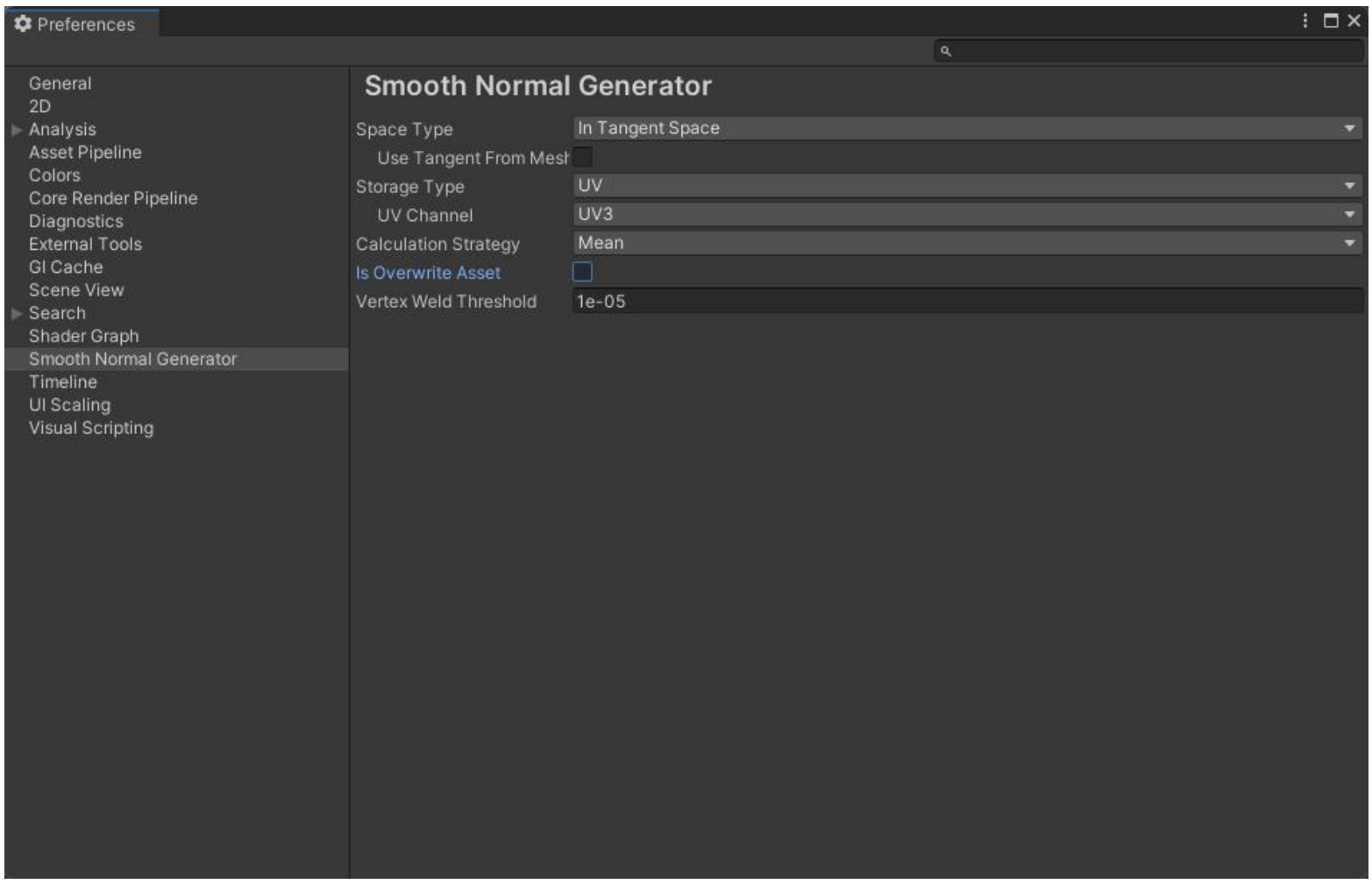
Create a Cube in the scene, click on the MenuItem of the MeshFilter component， then click on "Copy Mesh" in the popup menu (you will see mesh file in Asset directory).



Generate smoothed normals for this mesh file, drag modified mesh into MeshFilter, then drag the corresponding material into Mesh Renderer.



4. Preferences option



Space Type: In which space are the smooth normals generated, model space or tangent space.
Use Tangent From Mesh: When generating smooth normals in tangent space for an FBX file, the priority is to utilize the tangent information provided by the FBX itself. If such information is not found, the tangents will be automatically computed.
Storage Type: In which data format are the smooth normals stored, UV or vertex color. If stored in uv, smooth normals will be encoded as 2D vectors. For the decoding method, please refer to the relevant shader.
UV Channel: In which channel of uv are smooth normals stored.
Calculation Strategy: In which method are the smooth normals generated, mean averaging or least squares.
Is Overwrite Asset: Should the source file be overwritten directly.
Vertex Weld Threshold: When generating smooth normals for meshes not sourced from FBX files, if the distance between two vertices is less than this value, they are considered to be at the same position, and both vertices contribute to the calculation of a single smooth normal.

5. Model Space vs Tangent Space

For FBX model files with animations, it is imperative to generate smooth normals in tangent space. This is because vectors in tangent space maintain a constant direction relative to the tangent coordinate system on vertices, irrespective of vertex transformations. This ensures that the model's silhouette is accurately depicted throughout bone animation playback.

Conversely, for static objects, it is recommended to generate smooth normals in model space. This approach eliminates the need for vertex transformations from tangent space to model space during shader rendering, resulting in superior performance efficiency.

6. Least squares method for generating smooth normals

Assuming a vertex contains n normals and we want to compute a smooth normal, we aim for the smooth normal to lie as close as possible to the average direction of these n normals,

Given that N_i represents the i-th normal involved in the computation and N_{res} represents the final result, we aim for N_{res} to intersect, as closely as possible, the intersection point of the raised normal directions of n faces.

Thus, we have

$$dot((N_{res} - N_i), N_i) = 0$$

In matrix form, we have

$$\begin{bmatrix} \vec{x}(N_1) & \vec{y}(N_1) & \vec{z}(N_1) \\ \vec{x}(N_2) & \vec{y}(N_2) & \vec{z}(N_2) \\ \vdots & \vdots & \vdots \\ \vec{x}(N_n) & \vec{y}(N_n) & \vec{z}(N_n) \end{bmatrix} \begin{bmatrix} \vec{x}(N_{res}) \\ \vec{y}(N_{res}) \\ \vec{z}(N_{res}) \end{bmatrix} = \begin{bmatrix} |N_1| \\ |N_2| \\ \vdots \\ |N_n| \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$A_{n\times 3} N_{res} = E_{n\times 1}$$

If the matrix is invertible, based on the least squares method, we have

$$N_{res} = ({A_{n\times 3}}^T A_{n\times 3})^{-1} {A_{n\times 3}}^T E_{n\times 1}$$

Attention

1. Avoid polygons with a high number of edges in the model, as it may lead to failure in tangent calculation.
2. Ensure the mesh distribution of the model is reasonable to achieve more suitable orientations for smooth normals and a more natural depiction of the model's silhouette.
3. For FBX models with animation in the scene, ensure that the scale of the prefab root node and the skeleton root node is maintained at 1:1:1. If you need to change the model's scale, adjust the Scale Factor in the import settings and regenerate smooth normals.
4. The shaders mentioned in this document are only intended for use in the provided sample scenarios. If you have additional rendering requirements, such as using the same material to render textures and outlines, please refer to the shader examples in the samples and write your custom shader accordingly.

For instance, if you've generated smooth normals in tangent space and stored them in UV2, you can duplicate the example shader OutlineInTangentSpaceUV3.shader and modify the copy. Change TEXCOORD3 to TEXCOORD2 in the a2v section, and use this shader's material to render the model's outline.

```
24      struct a2v {
25          float4 vertex : POSITION;
26          float3 normal : NORMAL;
27          float4 tangent : TANGENT;
28          float2 smooth_normal : TEXCOORD2; //TEXCOORD3
29      };
30
```