Contact Author:

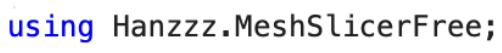
hanzemeng2001518@gmail.com

Purpose:

Mesh Slicer Free slices non deformable (so can't be used with Skinned Mesh Renderer) meshes into exactly two meshes. Mesh Slicer Free can handle some pretty complex meshes, such as those with holes. In general, as long as the intersection of the slice is a non self-intersecting polygon (more precisely, a planar subdivision), Mesh Slicer Free will slice correctly and, relatively, efficiently.

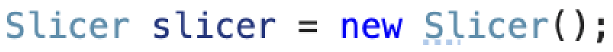
Usage:

You want to use this namespace:



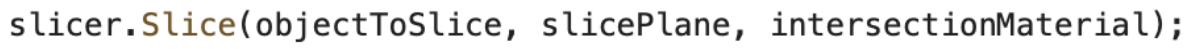
```
using Hanzzz.MeshSlicerFree;
```

You want to create a Slicer object by:



```
Slicer slicer = new Slicer();
```

You want to slice the object by:



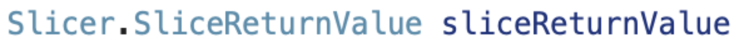
```
slicer.Slice(objectToSlice, slicePlane, intersectionMaterial);
```

objectToSlice is of type GameObject and has a Mesh Filter and Mesh Renderer component attached. **Make sure the mesh field in the Mesh Filter component has the read/write option enabled.**

slicePlane is of type Plane. **Make sure this plane is defined using world space coordinates.**

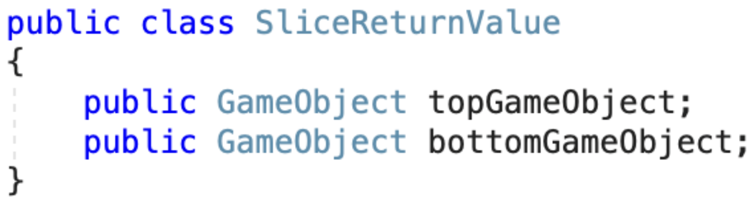
intersectionMaterial is of type Material. The slicer will fill the intersections with this material.

If the slicer operated correctly, it will return:



```
Slicer.SliceReturnValue sliceReturnValue
```

The definition of SlicerReturnValue is:



```
public class SliceReturnValue
{
    public GameObject topGameObject;
    public GameObject bottomGameObject;
}
```

topGameObject is part of the original object that was on top of the slice plane. topGameObject has exactly 4 components attached: GameObject, Transform, Mesh Filter, Mesh Renderer.
Notably, the Transform is at the root of the hierarchy and has the same position, rotation, scale as the original game object;
the Mesh Filter uses the sliced mesh;
the Mesh Renderer uses the original material(s) in addition to the intersection material.

Same story for bottomGameObject.

Note that the slicer does not modify any of the input parameters. If you need to destroy the original game object after the slice, you need to do it yourself (I am sure you got this).

Things That Should Work, But I Have Not Tested Them:

Should be able to slice meshes with multiple sub meshes.
Should be able to slice meshes with vertices that have all 8 UV channels.
Should be able to slice meshes with vertices that have color.

Things That Should Not Work:

Should not slice meshes with blend shapes and bind poses.
If the original mesh is weird (like clipping into each other, or a plane), then do not expect Mesh Slicer Free to slice correctly.

Possible Extensions:

I would like to add the following extension to this product, but need help and incentive:
Able to slice deformable meshes (need someone to teach me what is blend shape and bind pose).
Need a better way to give uv coordinates to intersection vertices (right now all intersection vertices have uv coordinates of (0,0)).
Asynchronous?
Multithreading?

Contact Author:

hanzemeng2001518@gmail.com