Installation

1. Drag OneCut.cs to any gameobject, but remember to add it once because it is a singleton.
2. Drag OneCutObject to gameobject you want to perform actual cutting. Not to apply it to every gameobject if you want better performance.

Class Detail

SpriteCutObject

- It will copy the sprite render you attach on, so once cutting is perform, it won't affect the original sprite.

CutResult

- Include all the information need after cutting
    - mainSprite => Mesh with larger area after cutting
    - subSprite => Mesh with lesser area
    - originalSprite => Mesh before cutting
    - intersectionPoints => Debug usage, see where your cut line intersect with original mesh

OneCut

- Singleton
- Check debugMode, if you want to see where your cut take place
- Method Cut is work under threading to prevent blocking ui thread

How to use

OneCut.Cut

```
Vector3 mouseDownWorld = camera.ScreenToWorldPoint(p_mouseDown);
Vector3 mouseUpWorld = camera.ScreenToWorldPoint(p_mouseUp);

RaycastHit2D[] hit2ds = Physics2D.LinecastAll(mouseDownWorld, mouseUpWorld, layermask);

foreach (var hit in hit2ds) {
    if (hit.transform != null) {

        OneCutObject spriteCutObject = hit.collider.GetComponent<OneCutObject>();
        OneCut.Instance.Cut(spriteCutObject, mouseDownWorld, mouseUpWorld, (OneCut.CutResult result, bool isSuccess) => {


            if (isSuccess) {
                //Call SpriteCutObject.ChangeSpriteMesh to apply new generated mesh
                spriteCutObject.ChangeSpriteMesh(result.mainSprite.triangle, result.mainSprite.meshTrig, result.mainSprite.meshVert);
            }
        });
    }
}
```

Its include in Sample Scene