实现点击效果。

o 用 Plane 或其他物体做地面， tag 为“Finish”

o 点击地面后，出现一个圆形攻击标记，两秒后自动消失。注意：该攻击标记不能挡住点击。（Primitive Objects / Cylinder）

o 请使用一个简单工厂创建、管理这些的标记，并自动收回这些标记（注意，这些对象创建后，放在列表内，不必释放）。

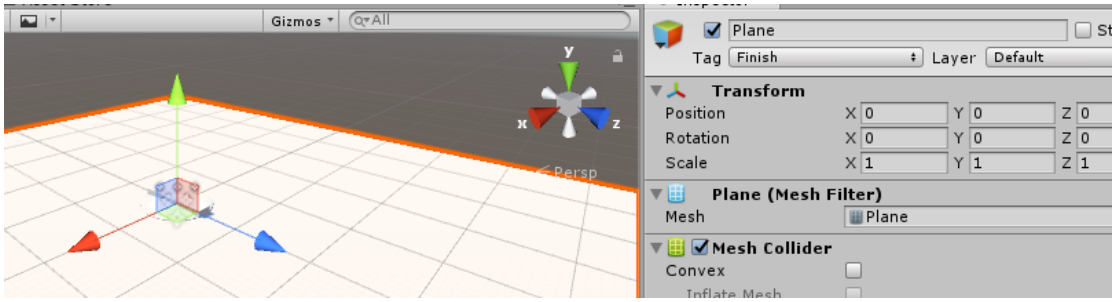
用户仅需申请使用即可

GameObjectmyFactory.placeAttackMark(Vector3 postion)

1. 实现点击效果。

```
if (Input.GetButtonDown ("Fire1")) {
            Debug.Log ("shoot!");
        }
```

2. 用 plane 或其它物体做地面，tag 为 finish



3.点击地面后出现一个圆形标记。

```
if (Physics.Raycast(ray, out hit, 100f))
        {
            if (Input.GetButtonDown ("Fire1")) {
                Instantiate (Resources.Load ("Prefabs/Cylinder
"));
                Debug.Log ("hit the ground");
            }
        }
```

4. 简单工厂模式

Factory.cs

```
public class Factory : MonoBehaviour {
    public static List<GameObject> free = new List<GameObject>
();
    public static List<GameObject> used = new List<GameObject>
();
    private static Factory _instance;
    public static Factory getInstance(){
        if (_instance == null)
            _instance = new Factory ();
```

```
        return _instance;
    }

    public void placeAttackMark(Vector3 postion){
        GameObject t;
        if (free.Count == 0)
            t = Instantiate<GameObject> (Resources.Load<GameOb
ject> ("Prefabs/Cylinder"),Vector3.zero,Quaternion.identity);
        else {
            t = free [0];
            free.Remove (t);
        }
        used.Add (t);
        t.transform.position = postion;
    }
```

工厂只有两个函数，分别是获取实例，以及 placeAttackMark。后者供用户调用。

SceneController.cs

```
public class SceneController : MonoBehaviour {
    private Factory myFactory;
    public GameObject Cam;
    // Use this for initialization
    void Start () {
        myFactory = Factory.getInstance ();
    }

    // Update is called once per frame
    void Update () {
        if (Input.GetButtonDown ("Fire1")) {
            Vector3 mp = Input.mousePosition;

            Camera ca = Cam.GetComponent<Camera>();
            Ray ray = ca.ScreenPointToRay(mp);

            RaycastHit hit;

            if (Physics.Raycast(ray, out hit))
            {
                if (hit.collider.gameObject.tag.Contains("Fini
sh"))
                {
```

```
                    print(mp);
                    myFactory.placeAttackMark(hit.point);
                }
            }
        }
    }
}
```

update 里面的函数负责给用户调用 placeAttackMark.判断逻辑也放在里面，如果点击，则调用放置标签这个函数。由工厂分配

最终效果：