

第十二次实验： 寻路系统

学号：22920212204392 姓名：黄勛

一、 实验目的

- 掌握 Navigation 寻路系统

二、 实验条件

- 系统环境：Windows 10 21H2
- 软件环境：Unity 3D 2021.3.14f1c1

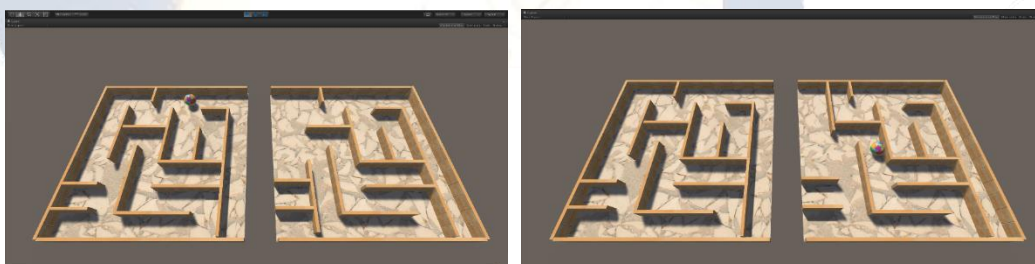
三、 实验内容

➤ 小球寻路

- 使用多个 Plane 和 Cube 对象搭建一个简易的迷宫:迷宫分为两部分，相互之间没有连接。
- 运行时可以通过鼠标点击场景，选择小球需要移动到的位置
- 小球能够在两个迷宫之间移动

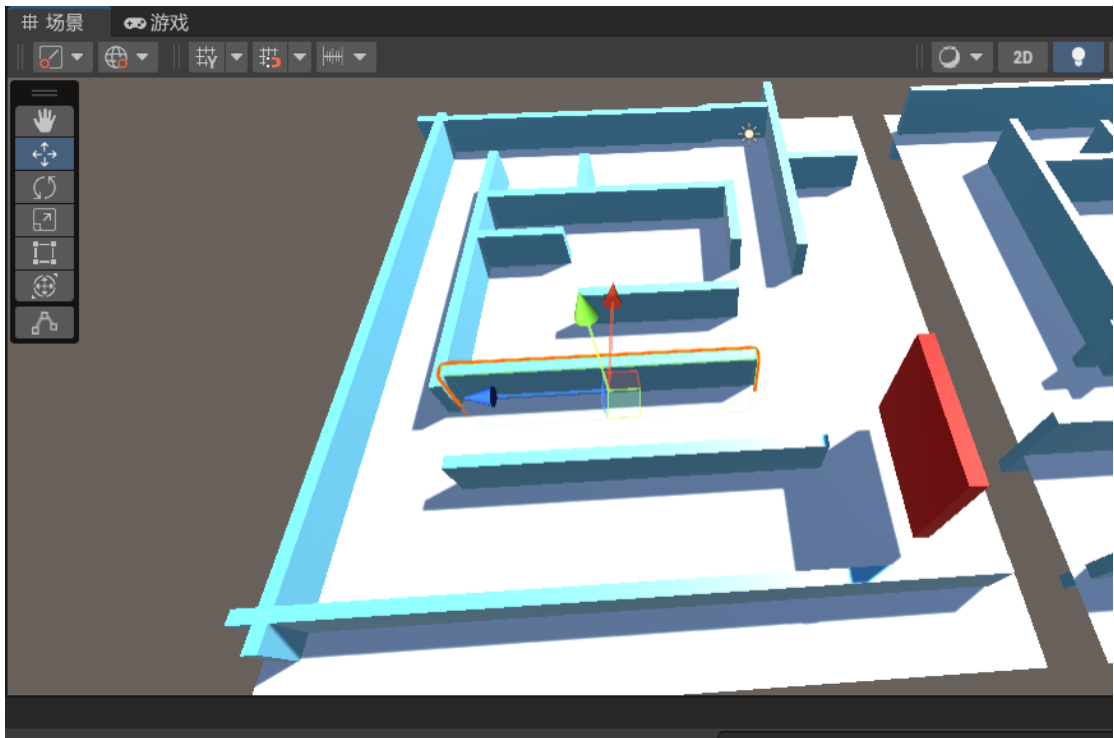
➤ 延伸任务

- 设置自动移动的障碍物，障碍物会与小球碰撞影响其移动

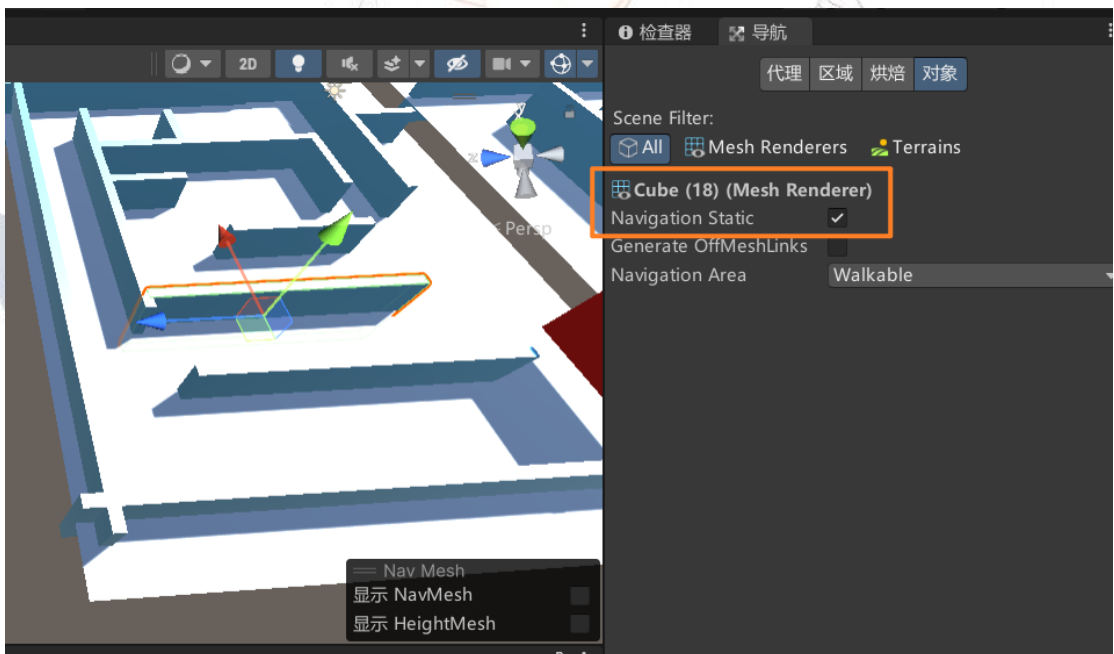


四、 实验项目步骤:

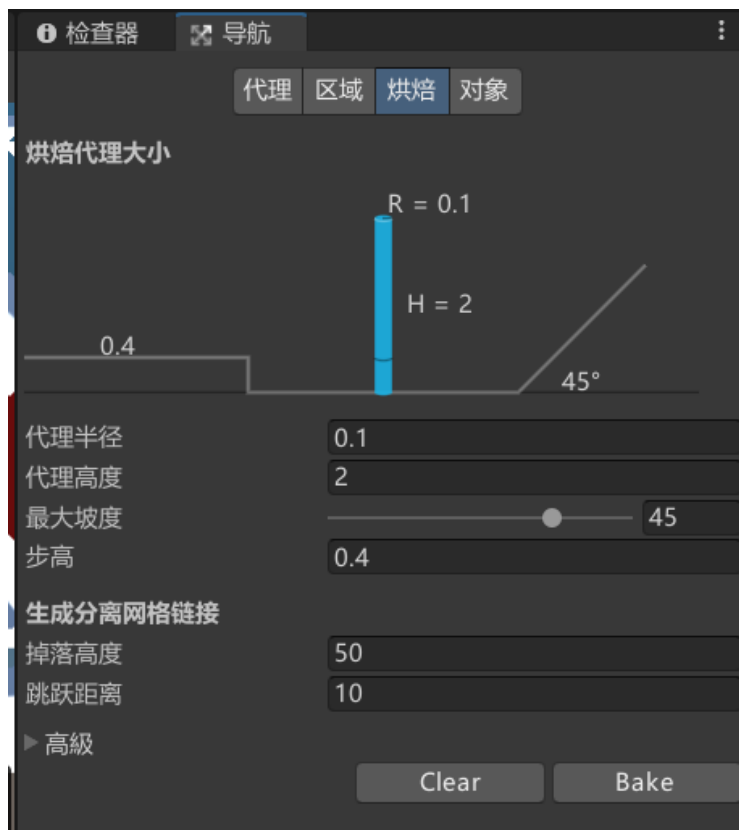
(一) 1. 摆一个迷宫的模型



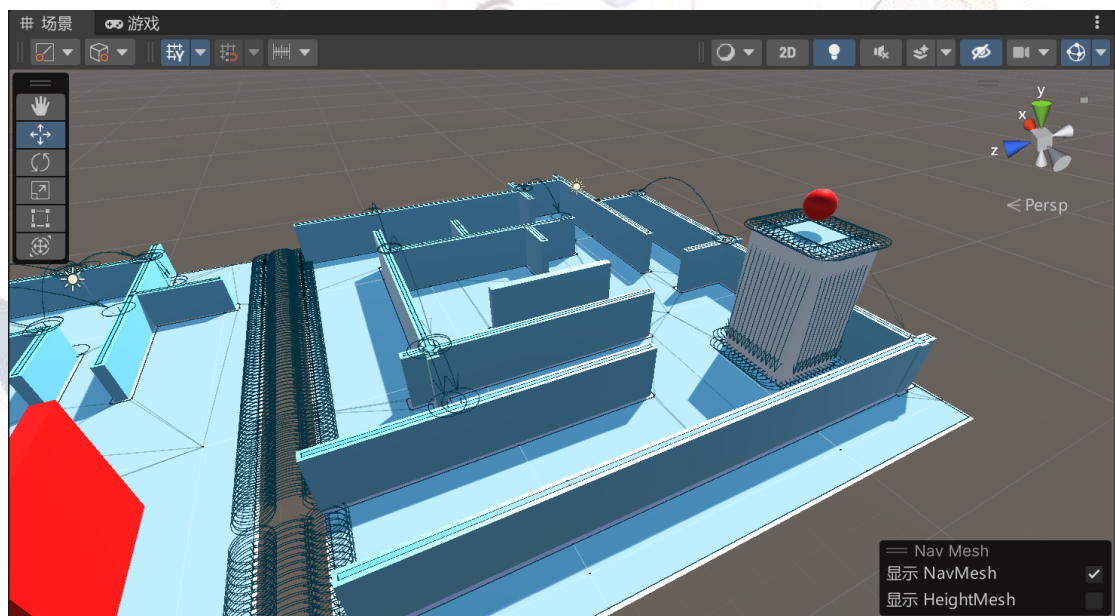
2. 设置迷宫障碍的静态属性为 Navigation Static



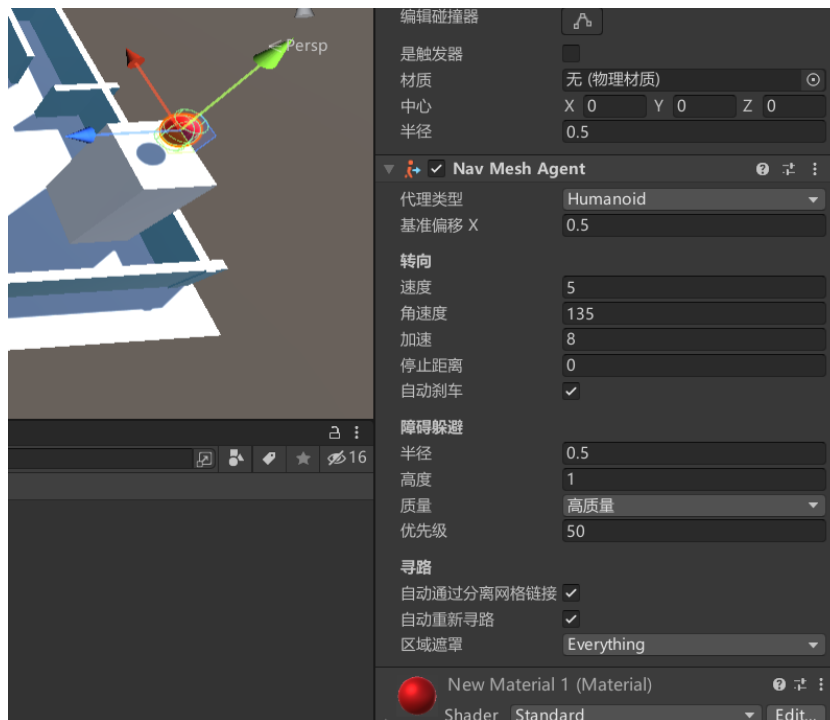
3. (1) 打开 AI→Navigation 窗口



(2) 烘焙



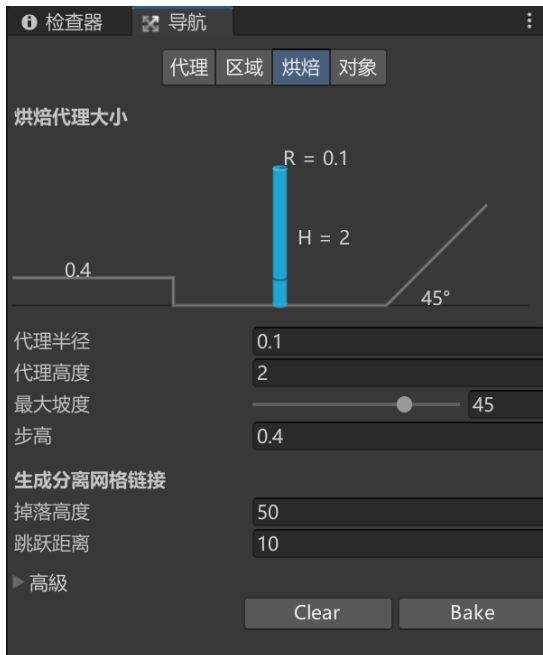
4. 新建小球，并为其添加 Nav Mesh Agent 组件



5. 为小球添加移动脚本

```
Assets > NavController.cs > ...
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.AI;
5
6 public class NavController : MonoBehaviour
7 {
8     NavMeshAgent agent;
9
10    public GameObject tar;
11    // Start is called before the first frame update
12
13    void Start()
14    {
15        agent = GetComponent<NavMeshAgent>();
16    }
17
18    // Update is called once per frame
19
20    void Update()
21    {
22        if (Input.GetMouseButtonDown(0))
23        {
24            //需要碰撞到物体才可以
25            Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);
26            RaycastHit hit;
27            bool isCollider = Physics.Raycast(ray, out hit);
28            if (isCollider)
29            {
30                Debug.Log("射线检测到的点是" + hit.point);
31                agent.SetDestination(hit.point);
32            }
33        }
34    }
35}
```

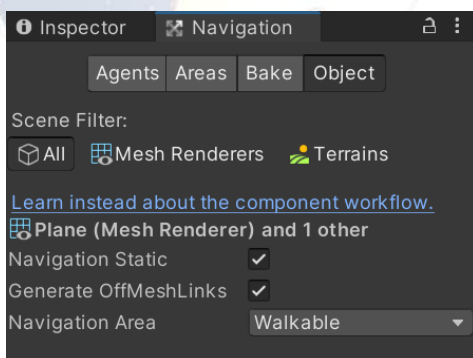
6. 设置小球可以从高处掉落

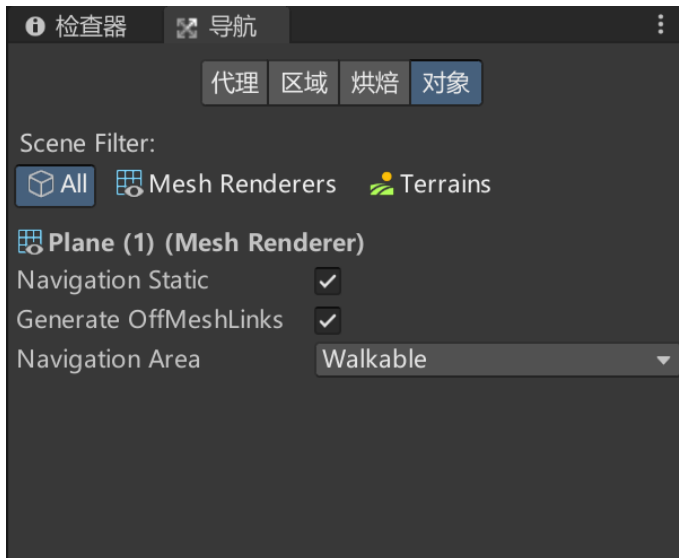


- (1) 改变 DropHeight
- (2) 勾选高台和地面之间的链接
- (3) 重新烘焙

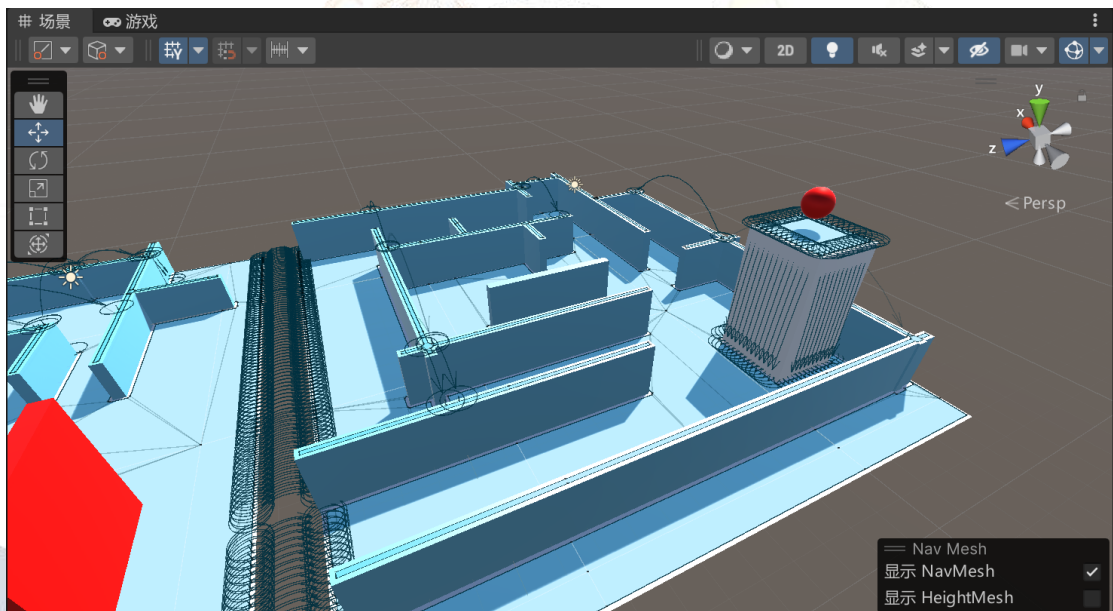
7. 使小球可以跳跃缝隙

- (1) 改变 JumpDistance
- (2) 勾选地面和地面之间的链接



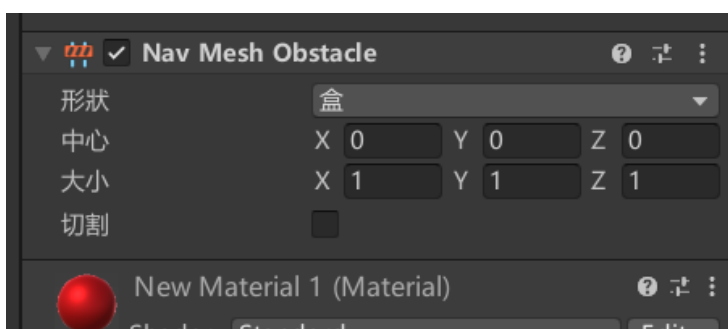


(3) 重新烘焙



7. 移动的障碍物

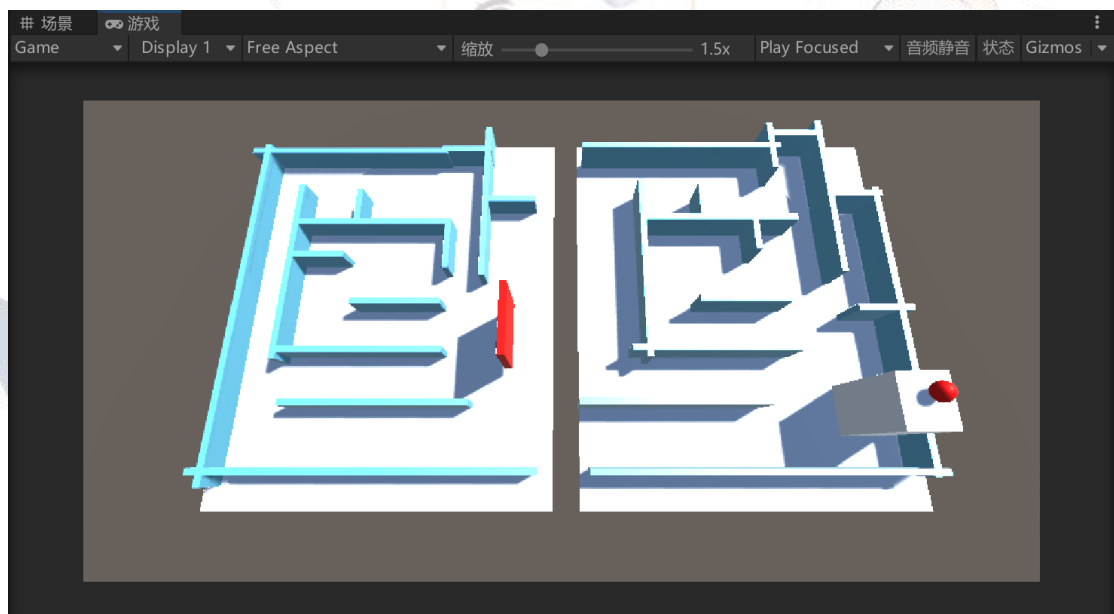
(1) 添加 Nav Mesh Obstacle 组件

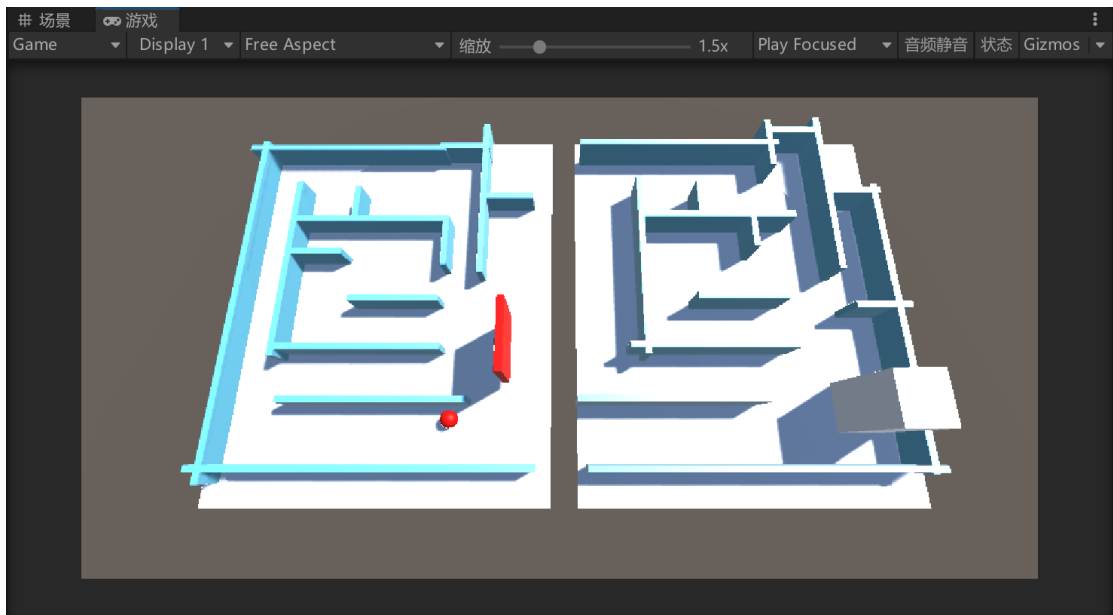


(2) 为其添加脚本，用两个空物体标记其变换的位置循环动画移动其位置

```
move.cs x NavController.cs
Assets > move.cs
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using DG.Tweening;
5 0 references
6 public class move : MonoBehaviour
7 {
8     1 reference
9     public GameObject p0;
10    1 reference
11    public GameObject p1;
12    // Start is called before the first frame update
13    0 references
14    void Start()
15    {
16        Tweener move0 = transform.DOMove(p0.transform.position, 4);
17        Tweener move1 = transform.DOMove(p1.transform.position, 4);
18        Sequence mySequence = DOTween.Sequence();
19        mySequence.Append(move0);
20        mySequence.Append(move1);
21        mySequence.SetLoops(-1);
22    }
23 }
```

8. 最终效果





此时即可实现实验所要求的全部内容!

最终效果详见视频演示

五、 实验心得总结：

在本次实验中，我学习了 Unity 的 Navigation 寻路系统，学会了 NavMesh Agent 组件的用法，跟着演示视频学会了小球寻路的功能。