

## 实验 6：学习使用 unity3D 软件

### 1. 为物体添加射击动作

1) 根据以下步骤创建带有刚性的小球。

1. 创建 sphere，调整 Y 坐标至合适值使其出现在视野内。
2. 添加 plane，调整至合适大小。

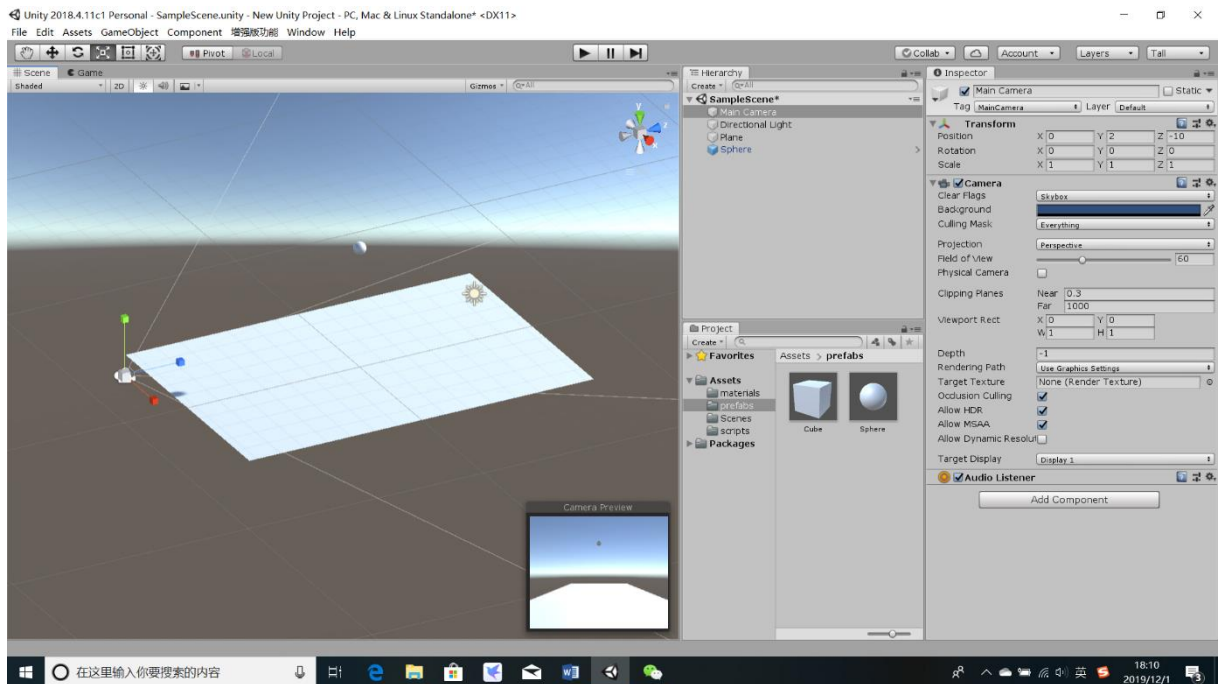



图 1：地板和小球

3. 创建的 sphere 预制体，在 inspector 中单击 Add Component，添加 Rigidbody 属性，可使用搜索框进行搜索，并将其实例化。
4. 此时按下顶部播放键 ，观察小球的运动，小球做一次自由落体运动。
5. 创建一个 cube 预制体,在场景中进行实例。实例位置在地面高度，且在小球的正下方。此时单击播放键，观察物体的运动，小球会模拟实际场景，自由落体后撞击立方体并反弹至一定高度。

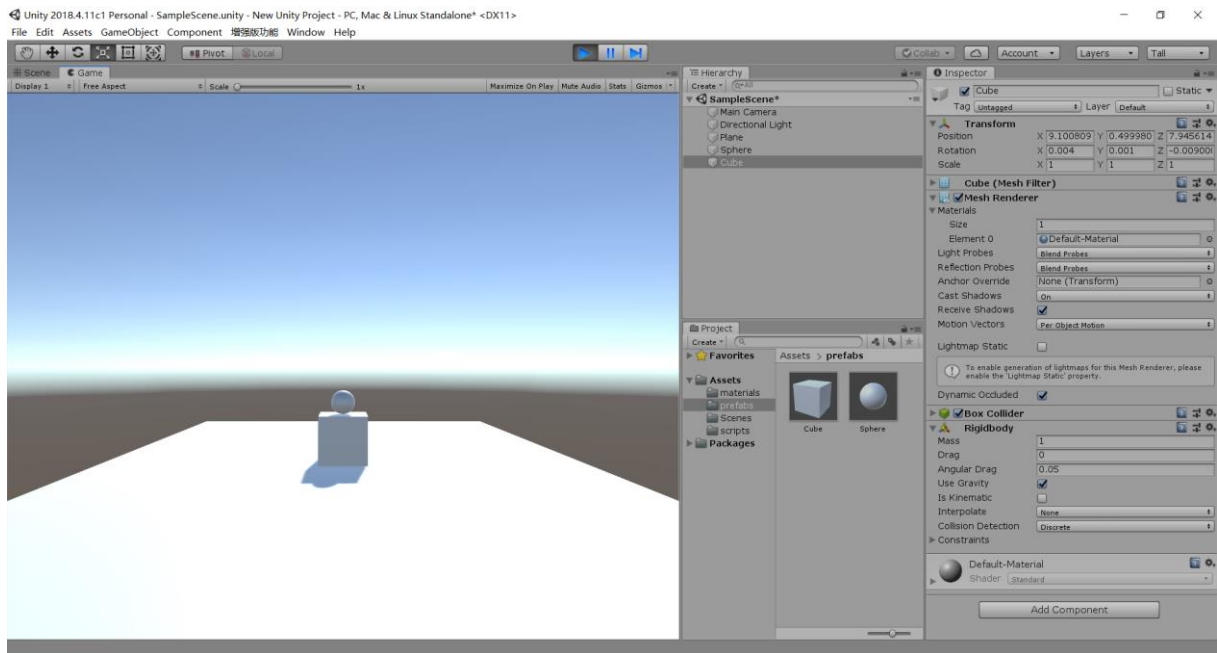


图 2 地板小球和立方体

## 2) 赋予小球发射属性。

1. 在 Assets 文件夹下新建 scripts 文件夹，进入 scripts 文件夹，并在其中新建名为 shoot 的 C#脚本文件。
2. 双击新建的脚本文件，软件会自动调用之前安装好的 visual studio。我们需要在 vs 中完成 shoot 脚本。

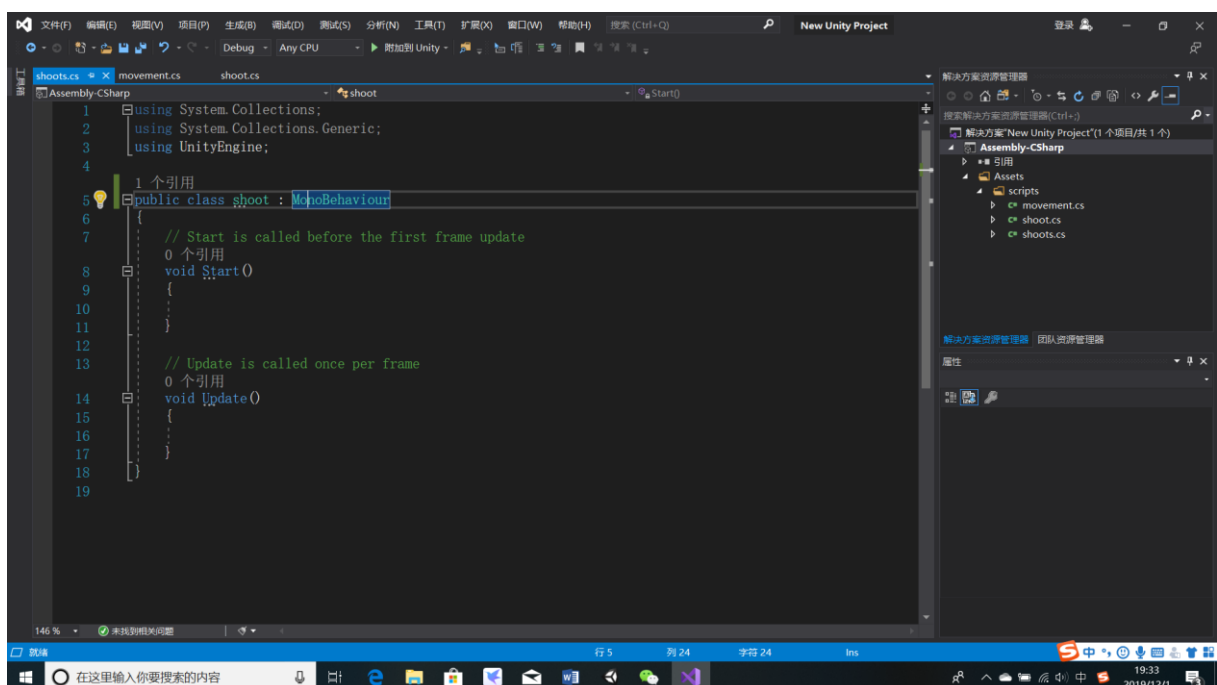


图 3 shoot C#脚本

3. Unity 编程需要我们具备一些 C#知识。脚本大致分为两个部分，一个是初始化函

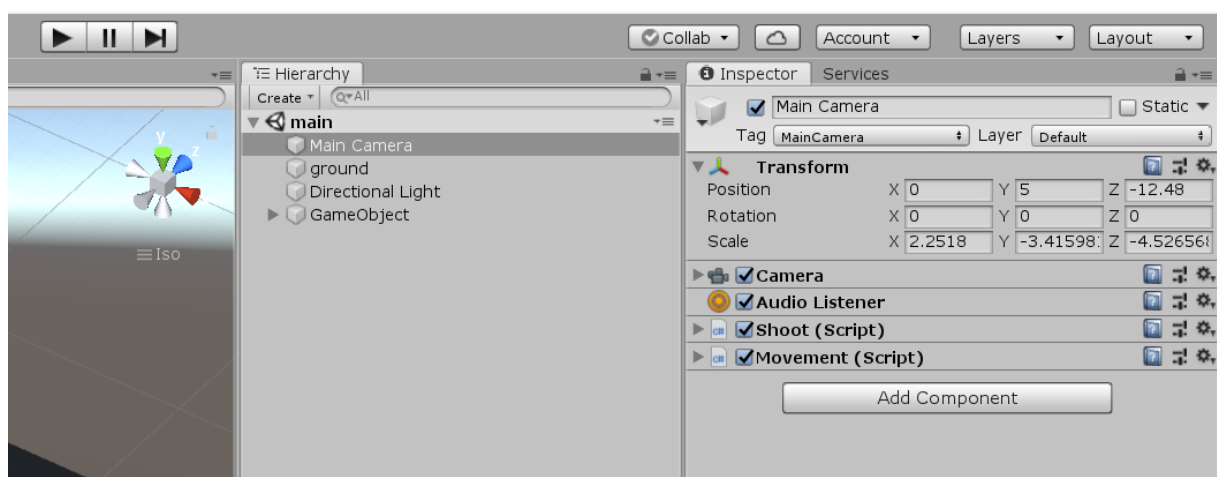
数（void start（）），一个是更新函数（void update（））。前者调用一次，后者一秒调用次数与动画帧数相同。

4. 编辑脚本文件如图 4 所示。并按 Ctrl+s 同步到 unity 工程。

```
0 个引用
5 public class shoot : MonoBehaviour
6 {
7     public GameObject bullet;
8     public float speed = 25;
9     // Start is called before the first frame update
10    0 个引用
11    void Start()
12    {
13    }
14
15    // Update is called once per frame
16    0 个引用
17    void Update()
18    {
19        Debug.Log("aa");
20        if(Input.GetButtonDown("Fire1")) //如果鼠标左键单击
21        {
22            Debug.Log("bb");
23            GameObject b = GameObject.Instantiate(bullet, transform.position, transform.rotation);
24            // 从bullet类实例化b, 并获得bullet的位置和方向
25            Rigidbody rgd = b.GetComponent<Rigidbody>();
26            //赋予 b 刚体属性
27            rgd.velocity = transform.forward * speed;
28            //赋予b 一个初速度
29        }
30    }
31 }
```

图 4 代码

3) 单击要赋予射击动作的小球，将 shoot 脚本拖动至 inspector 窗口，add component 上方，直至鼠标从‘禁止’标志变成‘释放’标志。（后续的 movement，移动脚本在编写完成时也要进行相同操作）。



4) 将实例 sphere 拖曳到 inspector/shoot/Bullet 右侧的矩形框中，如图 5 所示。

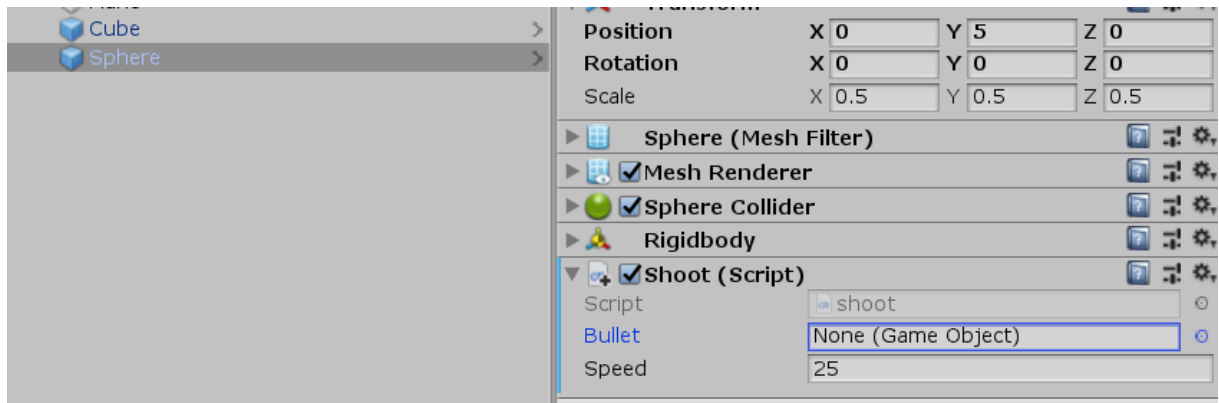


图 5 使 sphere 成为 bullet

- 5) 此时单击播放键进行游戏，发现小球可以进行自由落体，该过程中进行单击鼠标左键会在小球当时位置发射一颗小球。他的速度可以由 inspector 中的 speed 来控制。  
(该接口在脚本中实现，请同学们仔细推敲)

## 2. 为物体添加键盘控制移动动作

- 1) 在 scripts 下新建 C#脚本文件 movement，并双击打开，添加如下内容：

```

movement.cs  shoot.cs
Assembly-CSharp  movement  Update()
4
0 个引用
5  public class movement : MonoBehaviour
6  {
7      public float speed = 2;
8      // Start is called before the first frame update
9      0 个引用
10     void Start()
11     {
12     }
13
14     // Update is called once per frame
15     0 个引用
16     void Update()
17     {
18         float h = Input.GetAxis("Horizontal");
19         float v = Input.GetAxis("Vertical");
20         //Debug.Log(h);
21         //Debug.Log(v);
22         transform.Translate(new Vector3(h, v, 0)* Time.deltaTime );
23     }
24

```

- 2) 编写完成后同样将其拖动到要赋予该动作的物体的 inspector 窗口上，并运行游戏查看效果。

3. 同学们还可以添加其他的 **component**，如果没有（如 **shoot**）我们就需要自己编写脚本来实现该组件。
4. 将场景进行保存，并编译运行，查看自己的第一个 **unity** 游戏的效果。