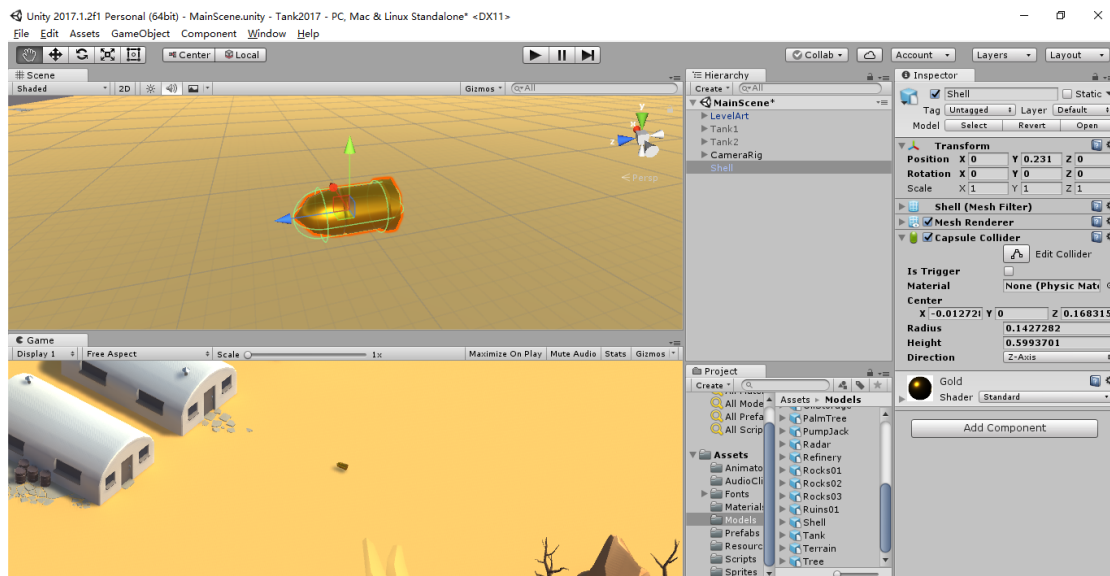


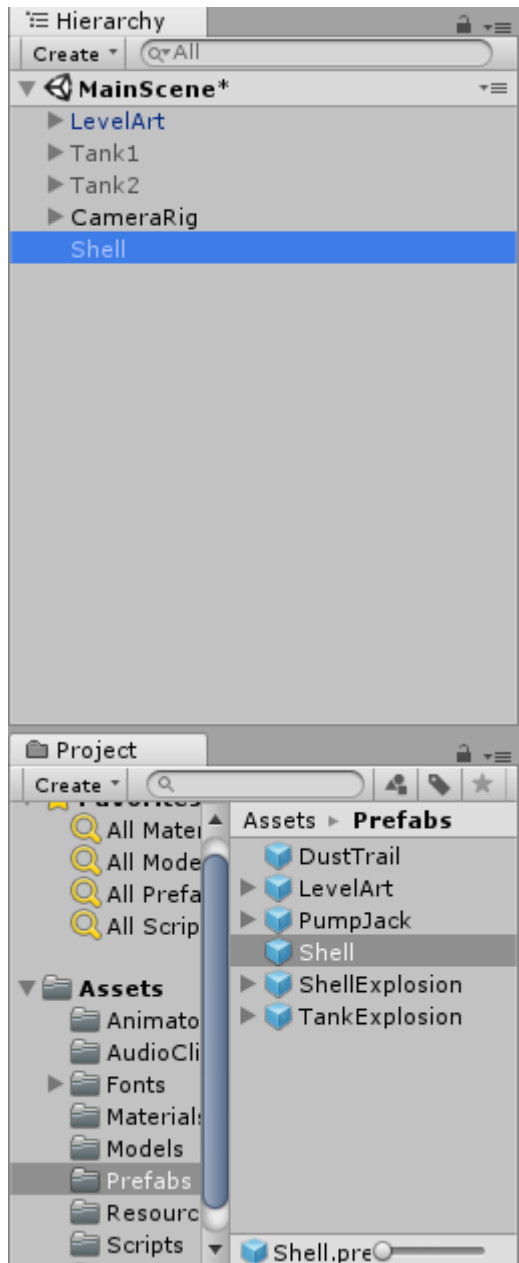
## Course 3 : Shoot it!

到目前为止,我们已经完成了 tank 的移动与摄像机移动的逻辑编写,本次课我们将实现的是子弹的发射。

1.从 model 文件夹中,找到 Shell,将其拖到场景中,作为我们的子弹,并为其添加一个 capsule collider, 调试其形状大小, 以正好包裹住该子弹的物体。并加上 Rigidbody 组件。(下图只是显示碰撞体的形状,还未添加 Rigidbody 组件)



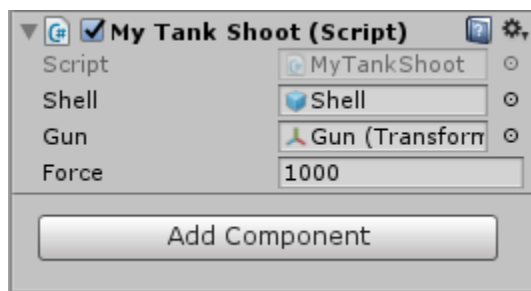
2.将该物体拖入 prefab 文件夹中, 让其成为一个 prefab 物体。



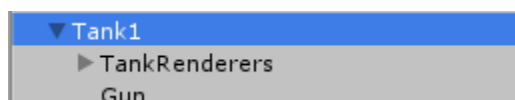
3.然后在 Scripts 文件中创建一个新的脚本,名为 MyTankShoot, 打开,  
进入编辑界面:  
键入以下代码:

```
Tank2017 MyTankShoot Start()
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class MyTankShoot : MonoBehaviour {
6
7     public GameObject Shell;           //子弹预制体
8     public Transform Gun;              //子弹发射的位置
9     public float force;               //添加的力
10    // Use this for initialization
11    void Start () {
12
13    }
14
15    // Update is called once per frame
16    void FixedUpdate () {
17        if (Input.GetKeyDown(KeyCode.Space)) //空格发射
18        {
19            GameObject temp = Instantiate(Shell, Gun);
20            temp.GetComponent<Rigidbody>().AddForce(force * transform.forward);
21        }
22    }
23
24 }
25
```

将其绑在 Tank1 上，并设好初值如下：

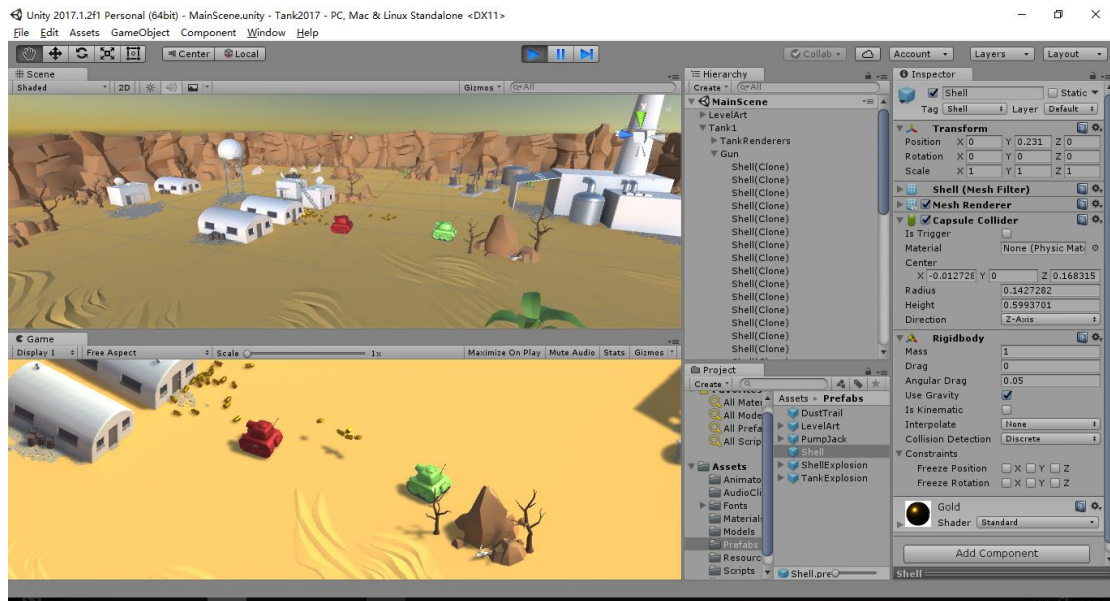


其中，Shell 是 prefab 文件夹中的 shell，Gun 是在 Tank1 下新建一个子物体，位置在炮管口处，是用于初始发射子弹的地方：



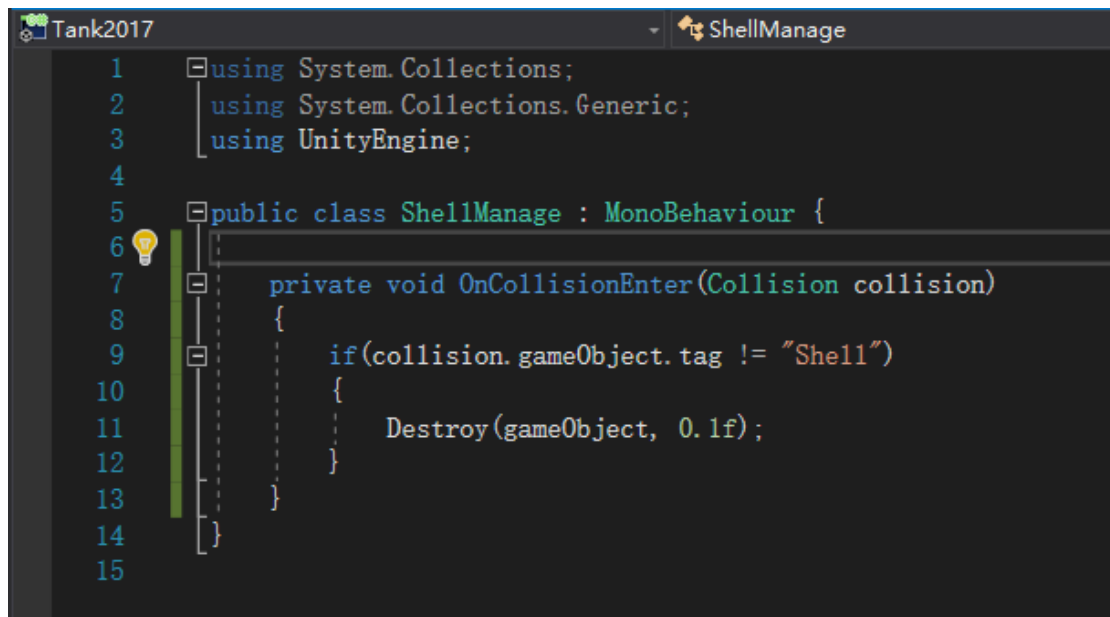
Force 是施加的力的大小。

现在我们应该可以运行

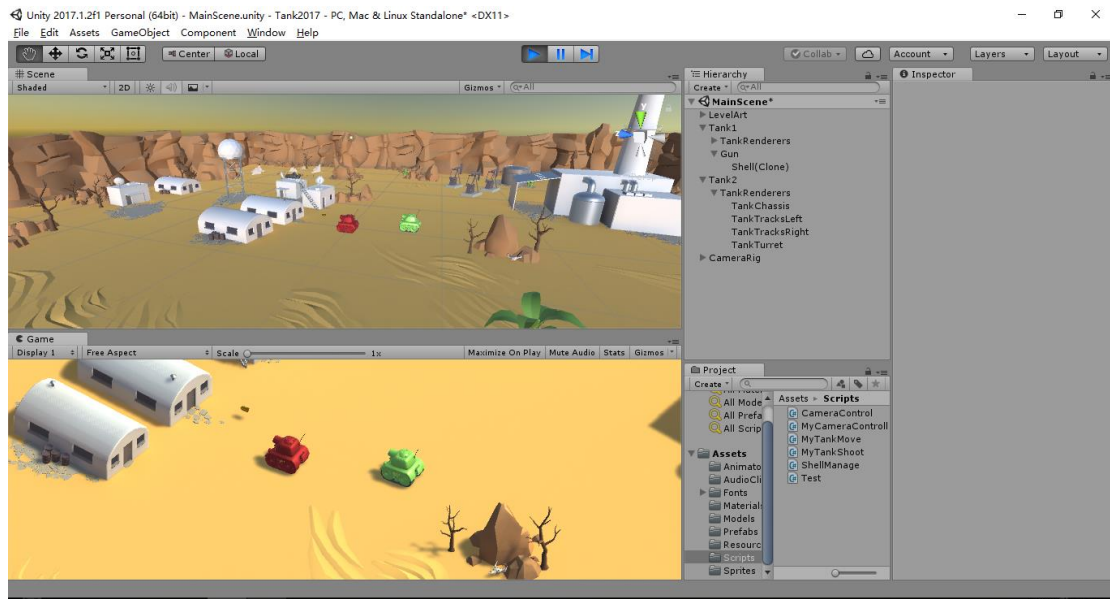


可以源源不断的射出很多的子弹啦！

但是撞到东西的子弹应该被销毁掉，所以我们在 Scripts 文件夹中再建一个脚本，名为 ShellManage。键入以下代码：



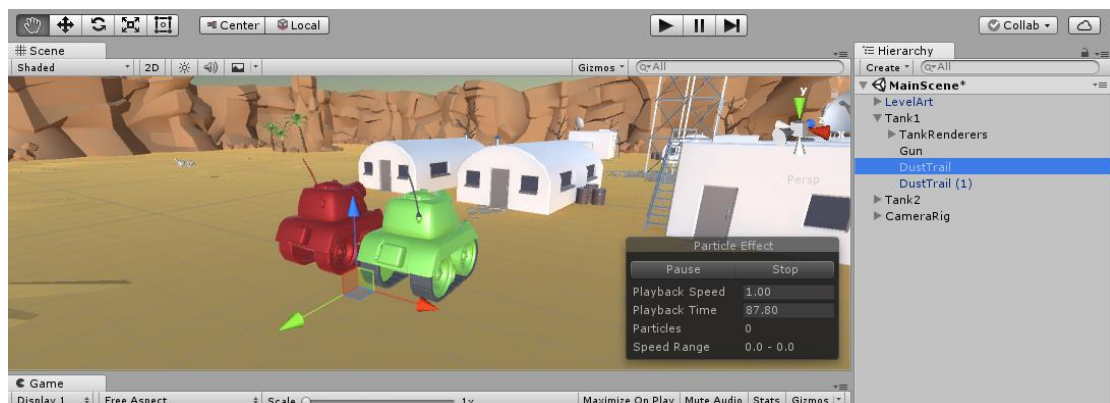
将其拖到 Shell 物体上，再次运行：



子弹撞到物体 0.1s 后就会自动销毁掉啦~

接下来是轻松愉快的特效添加时间：

找到 Prefab 文件夹中的 DustTrail，将其拖到 tank 子物体下，分别置于左右两边：



然后同样，将 ShellExplosion 添加到 Shell 上，在 ShellManage 中稍作修改：

```
public class ShellManage : MonoBehaviour {  
  
    private void OnCollisionEnter(Collision collision)  
    {  
        if(collision.gameObject.tag != "Shell")  
        {  
            this.transform.GetChild(0).GetComponent<ParticleSystem>().Play();  
            Destroy(gameObject, 0.25f);  
        }  
    }  
}
```

再次运行，我们就能看到较为酷炫的粒子效果啦~