

第三次实验： 游戏脚本

学号：22920212204392 姓名：黄勛

一、 实验目的

- 初步掌握 Unity 脚本编程

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

- 在程序启动时随机创建 5 个物体分布于随机位置
- 启动后再随机创建 5 个物体，每隔 2 秒创建一个
- 通过 OnGUI 在屏幕右上角显示最新创建物体的名称、位置，大小
- 延伸任务：让相机每次都对准新创建的物体，让该物体做随机运动或缩放

四、 实验步骤：

为了实现实验内容，主要的工作是编写 GlobalCoordinateSystem.cs 编辑器拓展脚本，主要内容与注释如下：

①编写 start 方法

```
5 public class RandomObjectSpawner : MonoBehaviour
6 {
7     private GameObject latestObj;
8     void Start()
9     {
10         for (int i=0;i<5;i++)
11         {
12             //在程序启动时随机创建5个物体分布于随机位置
13             if(Random.Range(0,1.0f)>0.5f){
14                 latestObj = GameObject.CreatePrimitive(PrimitiveType.Cube);//生成立方体
15             }else{
16                 latestObj = GameObject.CreatePrimitive(PrimitiveType.Sphere);//生成球
17             }
18             //随机设置立方体位置
19             latestObj.transform.position = new Vector3(Random.Range(-8, 8), Random.Range(-3, 3), Random.Range(-5, 5));
20             //随机设置立方体颜色
21             latestObj.GetComponent<Renderer>().material.color = new Color(Random.Range(0.0f, 1.0f), Random.Range(0.0f, 1.0f), Random.Range(0.0f, 1.0f));
22         }
23     }
24 }
```

完成了在程序启动时随机创建 5 个物体分布于随机位置，形状包括立方体和球（也可以增加更多形状，包括 PrimitiveType.Plane，PrimitiveType.Cube，PrimitiveType.Sphere，PrimitiveType.Capsule，PrimitiveType.Cylinder）

主要使用的方法包括：

`latestObj = GameObject.CreatePrimitive(PrimitiveType.XXX);`创建物体

`latestObj.transform.position = new Vector3();`生成随机位置

`latestObj.GetComponent<Renderer>().material.color = new Color();`生成颜色

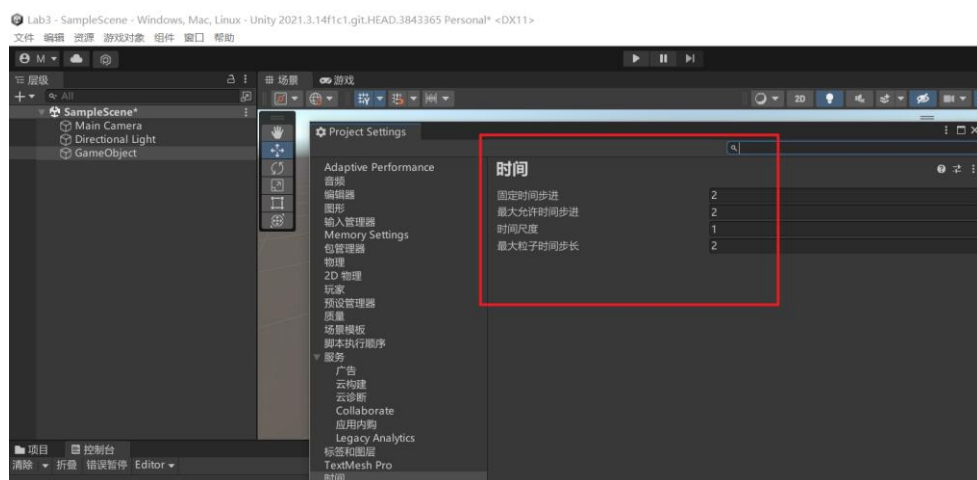
②编写 Update 方法，让目前生成且摄像机对准的物体移动和缩放

```
25 void Update()
26 {
27     // 让相机对准的物体做随机运动或缩放
28     latestObj.transform.position += new Vector3(Random.Range(-0.05f, 0.05f), Random.Range(-0.05f, 0.05f), Random.Range(-0.05f, 0.05f));
29     latestObj.transform.localScale += new Vector3(Random.Range(-0.05f, 0.05f), Random.Range(-0.05f, 0.05f), Random.Range(-0.05f, 0.05f));
30 }
```

改变的有 `latestObj.transform.position` 以及 `latestObj.transform.localScale`

③编写 FixedUpdate，每隔两秒调用一次，实现启动后再随机创建 5 个物体，每隔 2 秒创建一个。

```
32 int count = 0;
33 private void FixedUpdate()
34 {
35     // 启动后再随机创建5个物体，每隔2秒创建一个
36     if(count < 5)
37     {
38         if(Random.Range(0,1.0f)>0.5f){
39             latestObj = GameObject.CreatePrimitive(PrimitiveType.Cube); //生成立方体
40         }else{
41             latestObj = GameObject.CreatePrimitive(PrimitiveType.Sphere); //生成球
42         }
43         latestObj.transform.position = new Vector3(Random.Range(-8, 8), Random.Range(-3, 3), Random.Range(-5, 5));
44         latestObj.GetComponent<Renderer>().material.color = new Color(Random.Range(0.0f, 1.0f), Random.Range(0.0f, 1.0f), Random.Range(0.0f, 1.0f));
45         count++;
46         // 让相机每次都对准新创建的物体。
47         GameObject.Find("Main Camera").transform.LookAt(new Vector3(latestObj.transform.position.x, latestObj.transform.position.y, latestObj.transform.position.z));
48     }
49 }
```



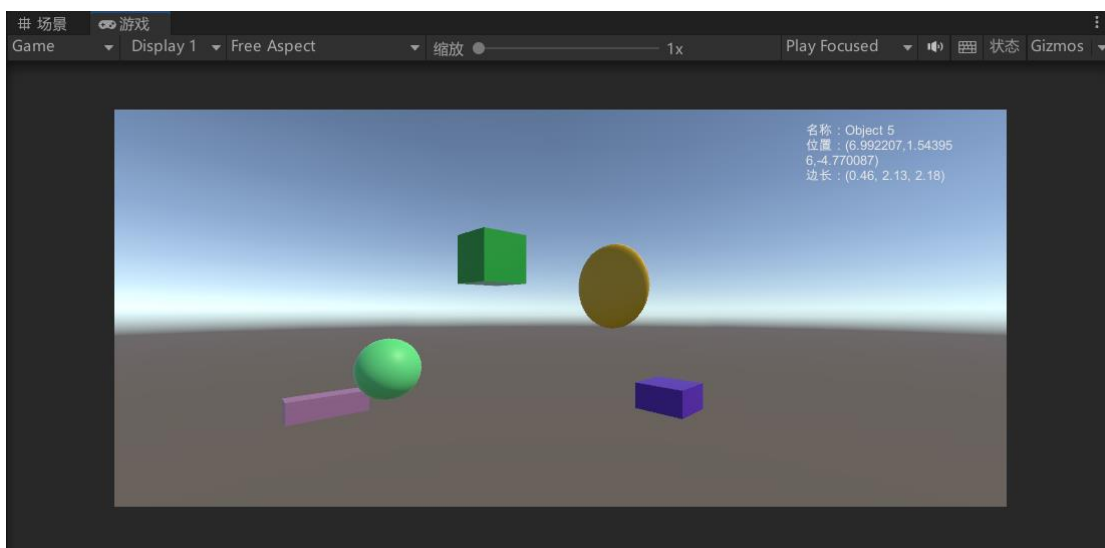
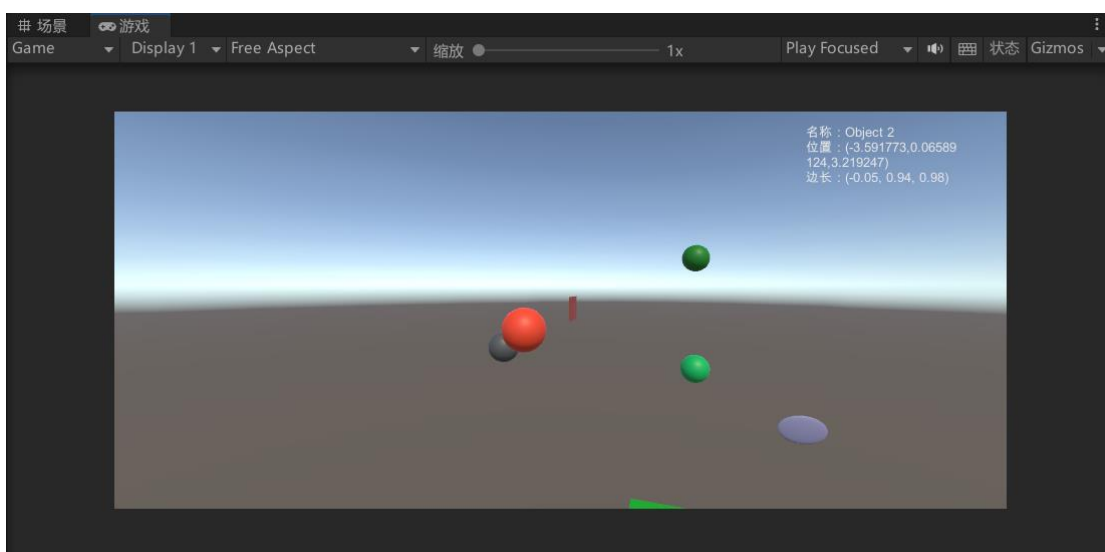
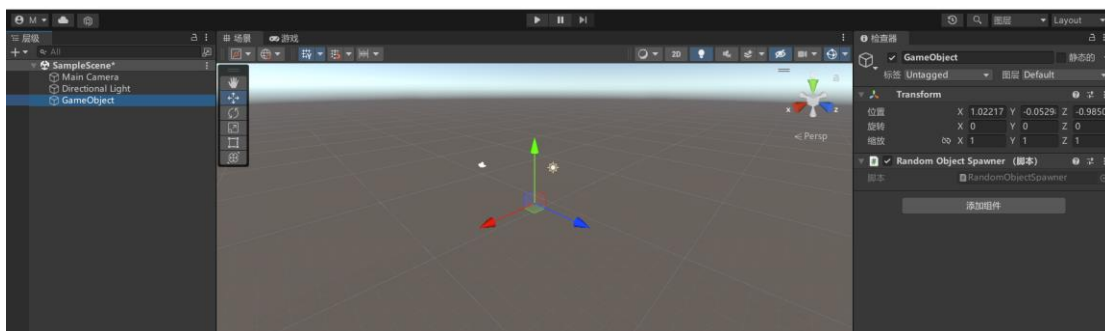
主要用到的新方法是 `GameObject.Find("Main Camera").transform.LookAt ()`,

来实现摄像头对准新生成的物体。

④编写 OnGUI，在界面展示名称、位置和大小。

```
51 private void OnGUI()
52 {
53     // 通过OnGUI在屏幕右上角显示最新创建物体的名称、位置、大小
54     GUI.Label(new Rect(Screen.width - 200, 10, 150, 200), "名称: Object "+count+"\n");
55     GUI.Label(new Rect(Screen.width - 200, 25, 150, 200), "位置: (" + latestObj.transform.position.x + ", "+ latestObj.transform.position.y + ", "+ latestObj.transform.position.z + ")");
56     GUI.Label(new Rect(Screen.width - 200, 55, 150, 200), "边长: " + latestObj.transform.localScale);
57 }
```

⑤最后在场景创建一个空物体并挂载编写好的脚本，即可运行！



五、 实验心得总结：

本次实验使用 Unity 引擎，通过 C#编写脚本我实现了以下功能：

1. 在程序启动时随机创建 5 个物体分布于随机位置。

通过使用 `Random.Range` 方法生成 0 到 1 的随机数，当随机数大于 0.5 时创建一个立方体，否则创建一个球体。然后通过设置 `transform.position` 属性，将物体随机分布于范围

内。

2. 启动后再随机创建 5 个物体，每隔 2 秒创建一个。

通过在 `FixedUpdate` 函数中设置计数器，控制随机创建物体的数量。通过使用 `Random.Range` 方法生成 0 到 1 的随机数，当随机数大于 0.5 时创建一个立方体，否则创建一个球体。然后通过设置 `transform.position` 属性，将物体随机分布于范围内。计数器加 1，直到创建完 5 个物体。

3. 通过 OnGUI 在屏幕右上角显示最新创建物体的名称、位置，大小。

通过在 OnGUI 函数中使用 `GUI.Label` 方法，设置标签的位置和内容，实现显示最新创建物体的名称、位置和大小。其中名称是通过计数器来设置，位置和大小则是通过访问 `latestObj` 的 `transform` 组件属性来获取。

4. 延伸任务：让相机每次都对准新创建的物体，让该物体做随机运动或缩放。

在 `Update` 函数中，使用 `transform.position` 和 `transform.localScale` 属性，对最新创建的物体进行随机运动或缩放。在 `FixedUpdate` 函数中，使用 `GameObject.Find` 方法来获取场景中的相机对象，通过调用 `transform.LookAt` 方法，实现让相机每次都对准新创建的物体。这样可以让用户在运行程序时，更加直观地看到新物体的变化过程。

总的来说，本次实验通过使用 Unity 引擎和 C# 编写脚本，实现了随机创建物体、显示物体信息和让相机对准新创建物体等功能，这些学习提高了我的 Unity 使用技巧，使得学习设计变得更加高效和方便。在今后的开发中，我将继续使用这些技巧来提高我的效率。