

## 第六次实验： 输入事件

学号：22920212204392 姓名：黄勛

### 一、 实验目的

- 了解 Unity 处理输入事件的逻辑流程
- 掌握常见的输入处理方法

### 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

➤ 在上个实验的基础上做以下更改

- 登录界面
  - 账号密码输入框，可以输入账号密码，
  - 点击登录在 Console 输出当前所登录账号，切换到主界面
- 游戏界面
  - 主窗口显示若干三维场景元素
  - 点击选中某个物体，并线框高亮
- 延伸任务
  - 点击地面将当前选中物体移到点击的位置

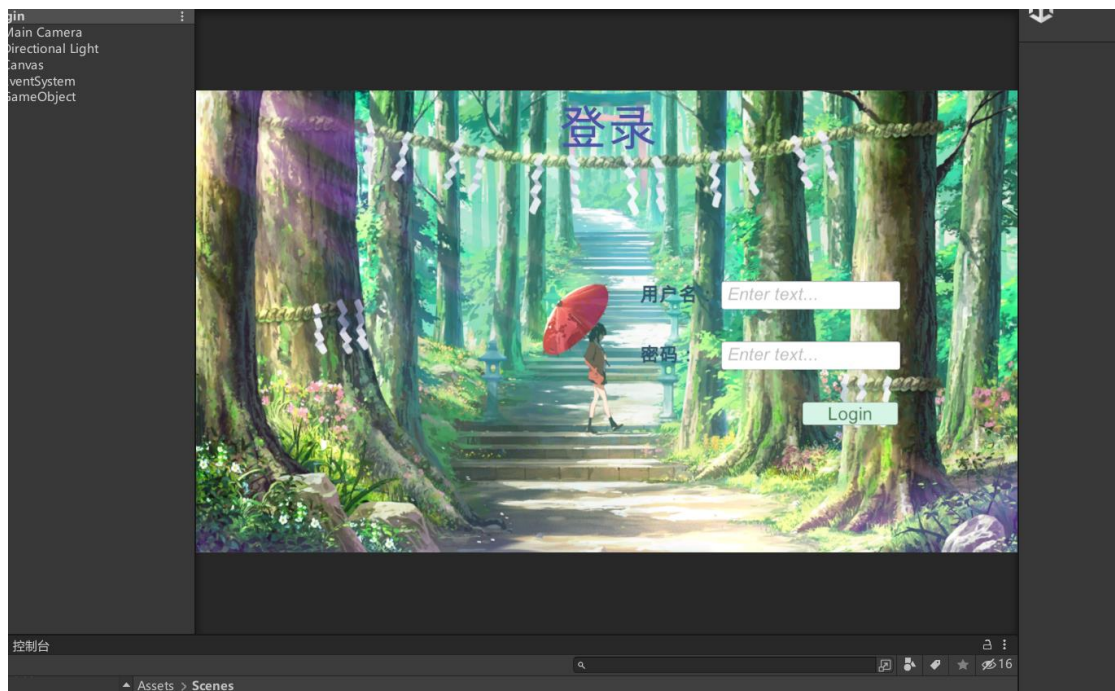
### 四、 实验项目步骤：

为了实现实验内容,主要的工作是编写 2D 模板 UI 项目和拓展移动物体脚本场景，

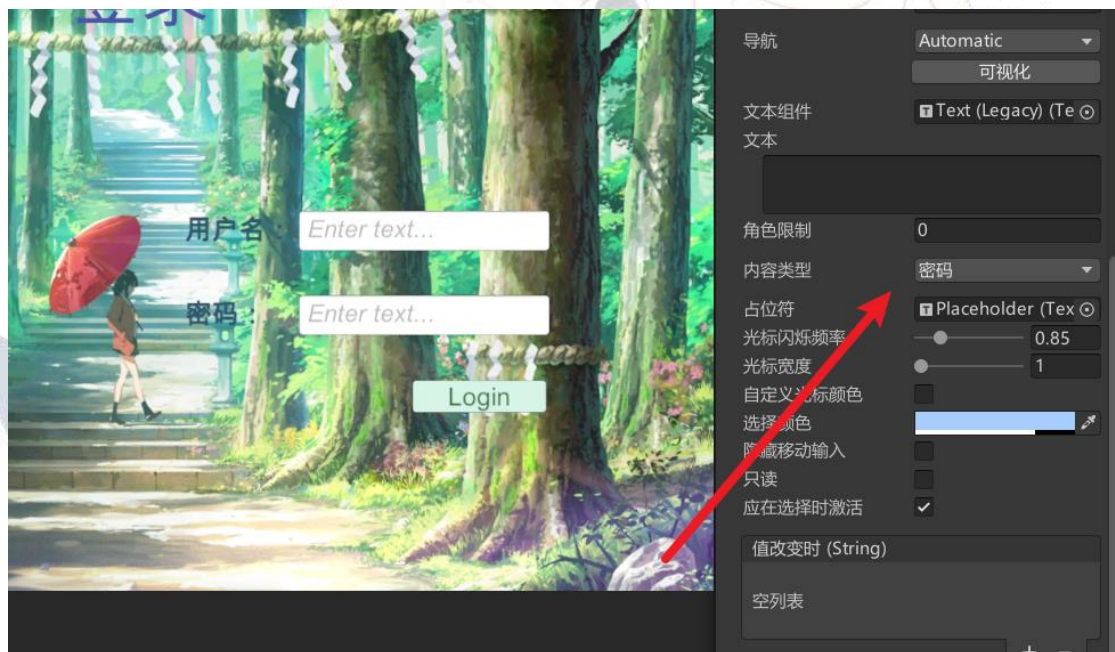
主要内容与注释如下：

#### （一）登陆界面

设计场景背景以及账号密码输入框，可以输入账号密码



对密码进行加密处理

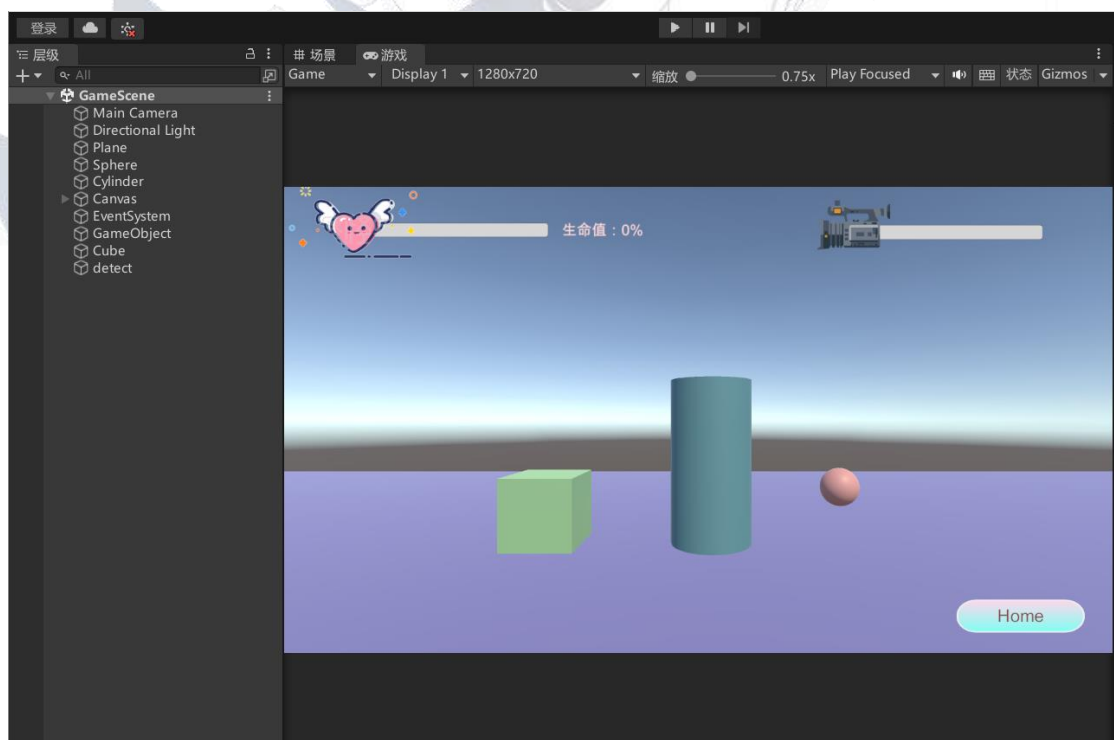


编写脚本，实现点击登录在 Console 输出当前所登录账号，切换到主界面

```
Assets > Scripts > Login.cs > ...
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5 using UnityEngine.SceneManagement;
6
7 0 references
8 public class Login : MonoBehaviour
9 {
10     2 references
11     public InputField UserName;
12     1 reference
13     public InputField Password;
14     0 references
15     public void Loginto()
16     {
17         if(UserName.text.ToString()!=""&&Password.text.ToString()!="")
18         {
19             Debug.Log("UserName :" + UserName.text.ToString());
20             Debug.Log("登录成功");
21             SceneManager.LoadScene(2);
22         }
23         else
24         {
25             Debug.Log("用户名或密码为空");
26         }
27     }
28 }
```

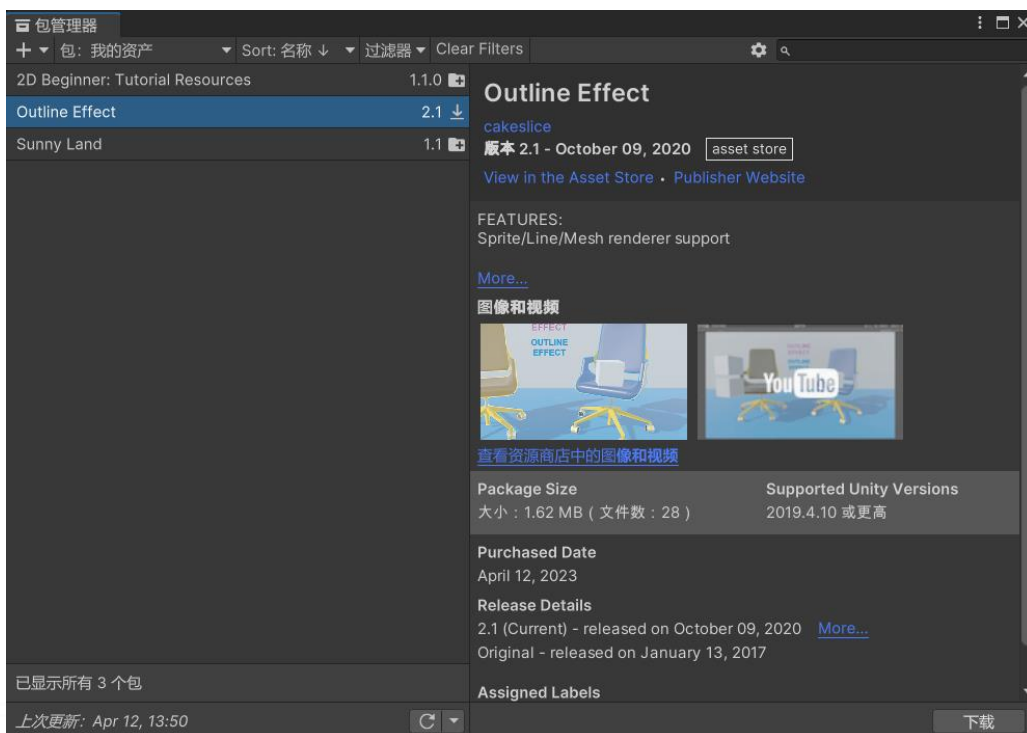
## (二) 游戏界面

### 主窗口显示若干三维场景元素

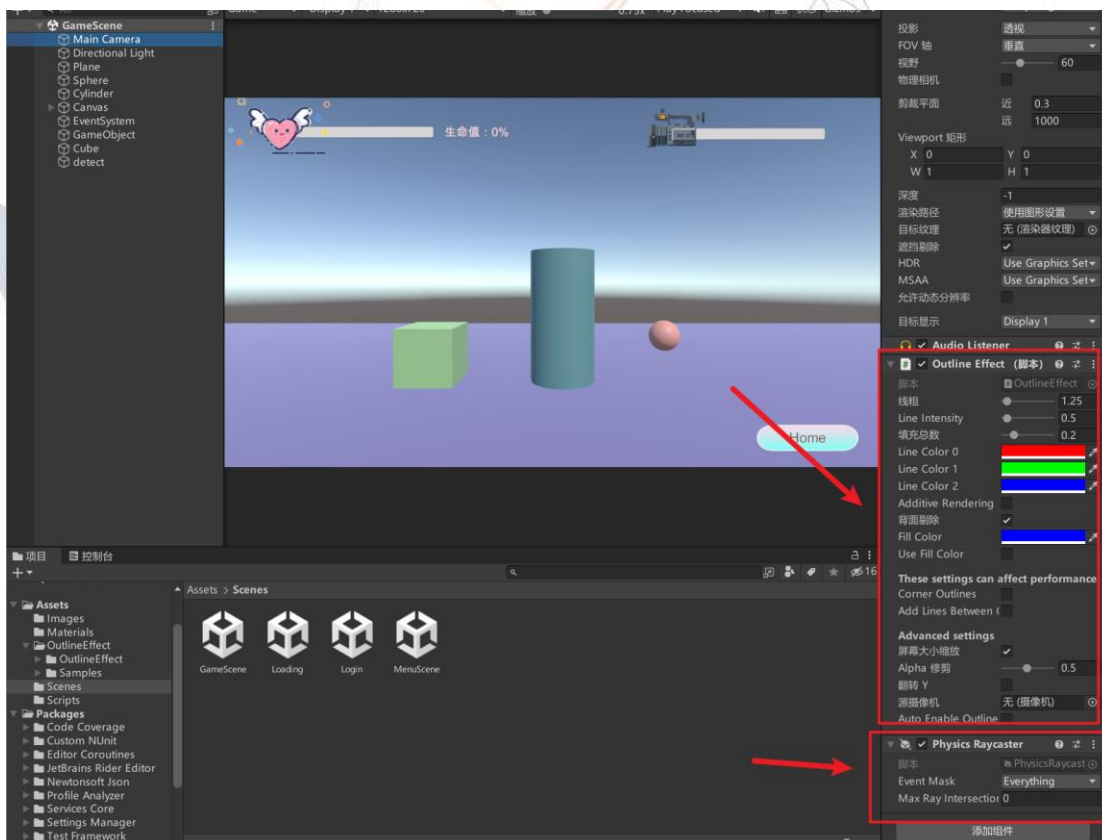


实现点击选中某个物体，并线框高亮：

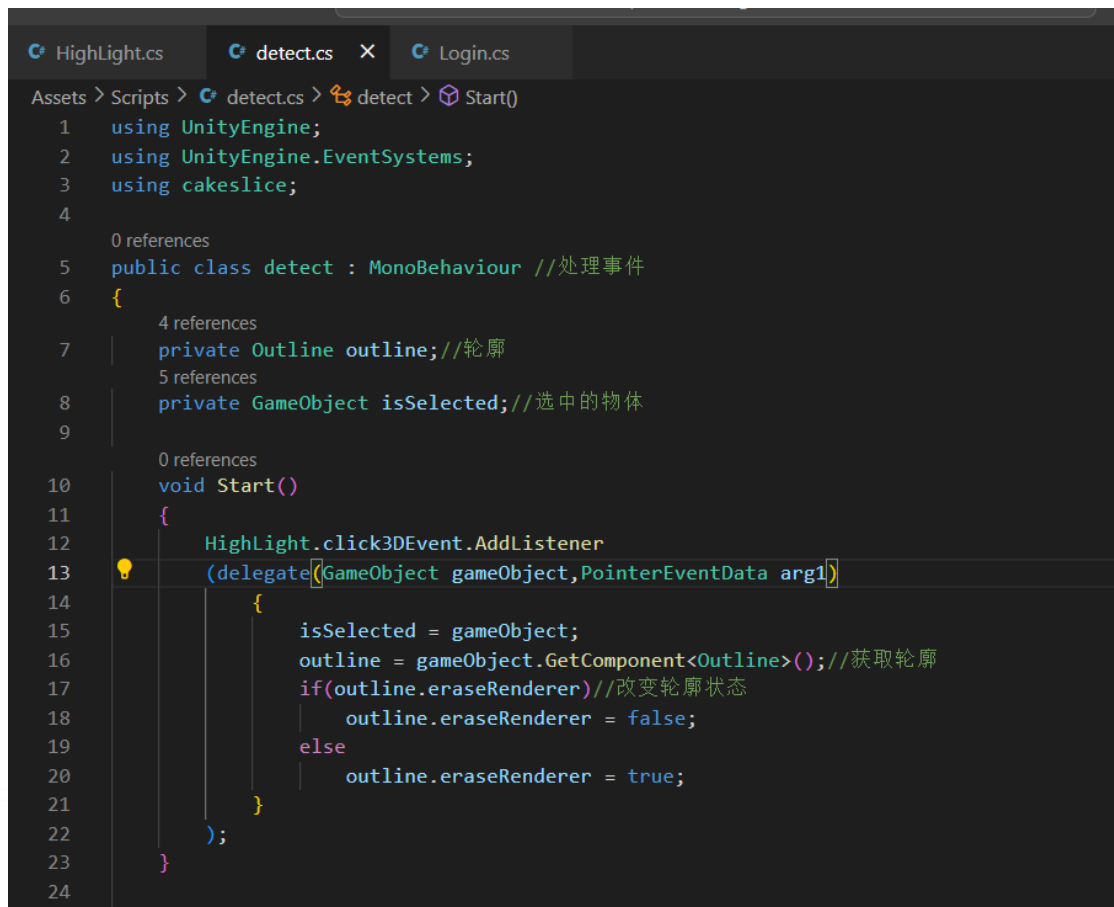
1) 从 github 下载并导入 Outline Effect 包



2) 对摄像头加入 Outline Effect 组件和 Physics Raycaster 用于射线功能

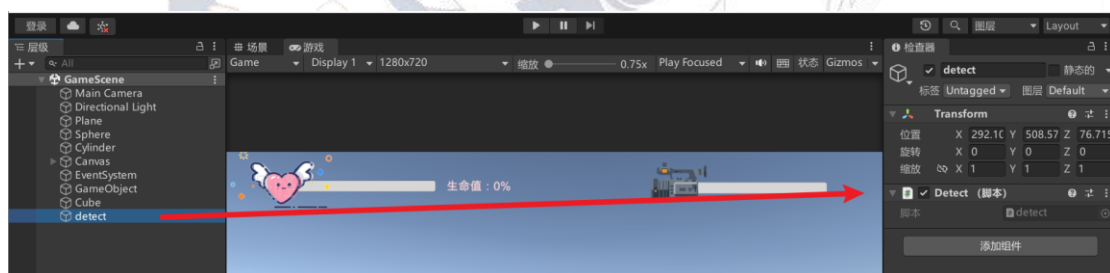


### 3) 编写 detect 脚本，用于监听点击事件并修改轮廓是否显示的状态



```
Assets > Scripts > detect.cs > detect > Start()
1  using UnityEngine;
2  using UnityEngine.EventSystems;
3  using cakeslice;
4
5  0 references
6  public class detect : MonoBehaviour //处理事件
7  {
8      4 references
9      private Outline outline;//轮廓
10     5 references
11     private GameObject isSelected;//选中的物体
12
13     0 references
14     void Start()
15     {
16         HighLight.click3DEvent.AddListener
17         (delegate(GameObject gameObject,PointerEventData arg1)
18         {
19             isSelected = gameObject;
20             outline = gameObject.GetComponent<Outline>();//获取轮廓
21             if(outline.eraseRenderer)//改变轮廓状态
22                 outline.eraseRenderer = false;
23             else
24                 outline.eraseRenderer = true;
25         });
26     }
27 }
```

### 4) 挂载在空物体上

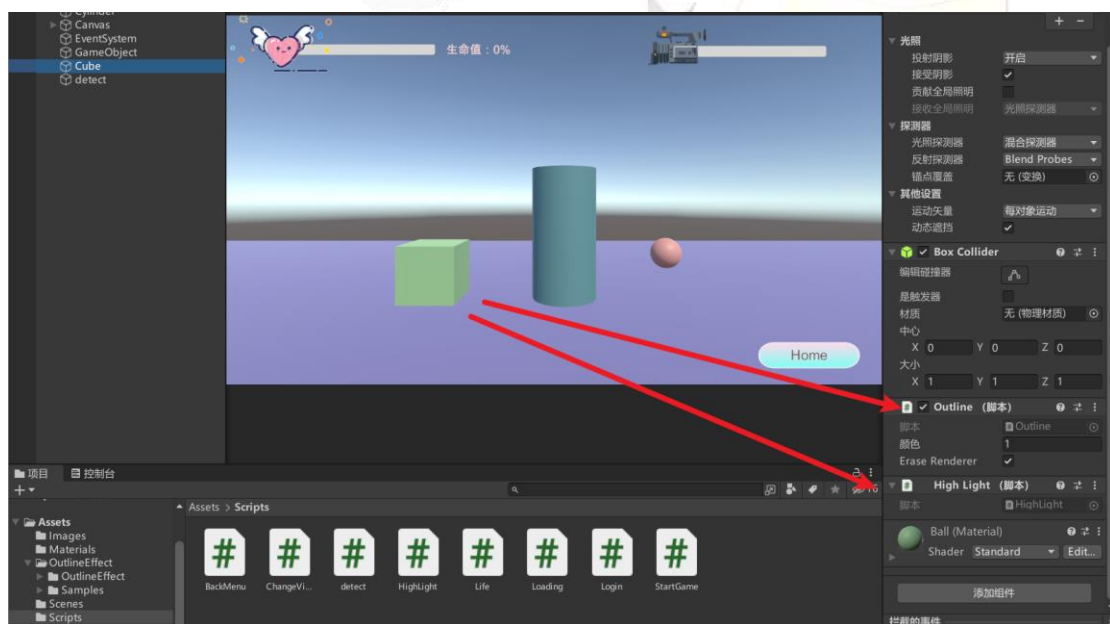


### 5) 编写 HighLight 脚本，与 Outline 一同挂载在需要添加轮廓的物体上，批量添加监听

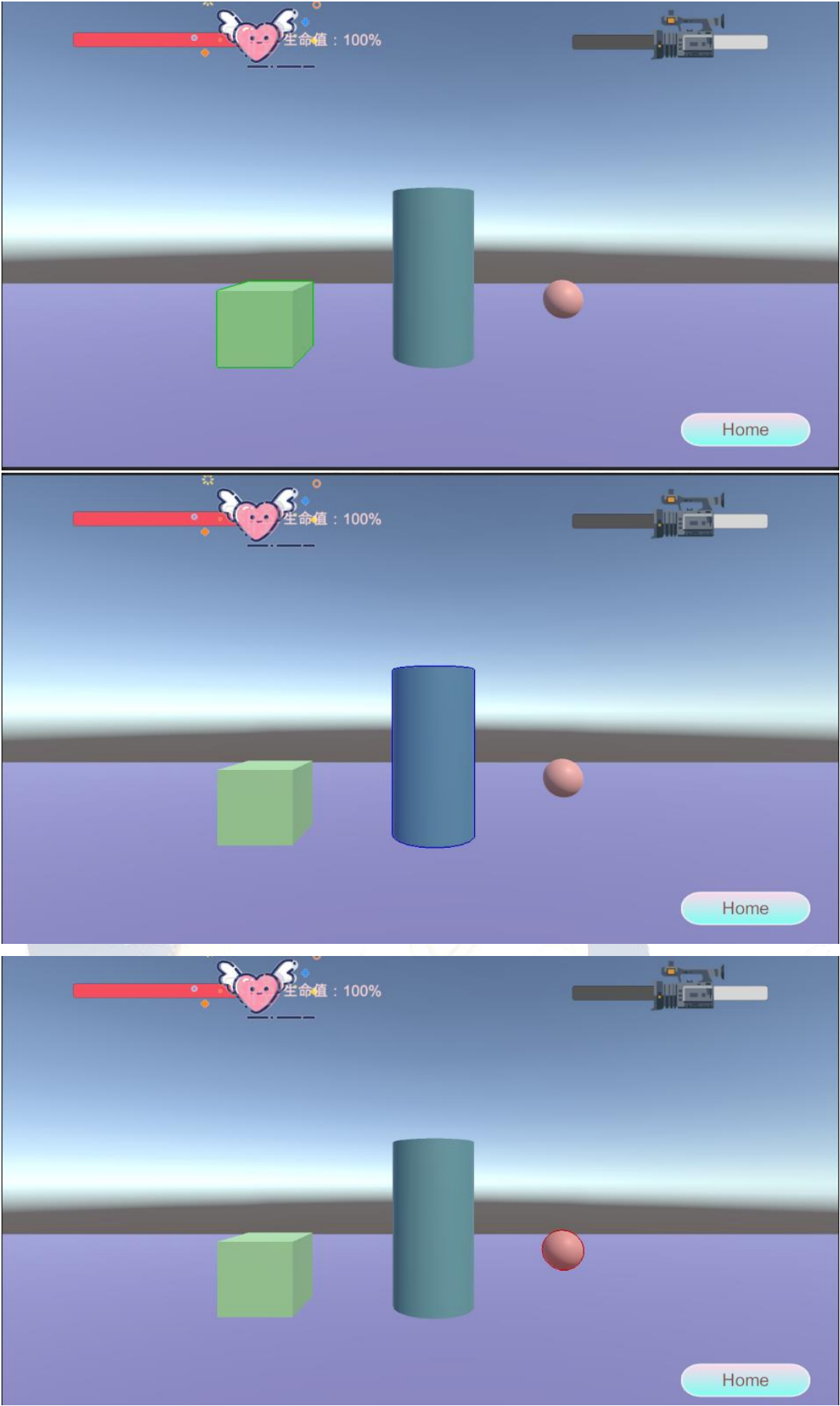


```
Assets > Scripts > HighLight.cs > ...
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.EventSystems;
5  using UnityEngine.Events;
6
7  public class Click3DEvent : UnityEvent<GameObject,PointerEventData>{}
8
9
10 public class HighLight:MonoBehaviour,IPointerClickHandler
11 {
12     public static Click3DEvent click3DEvent = new Click3DEvent();//static event
13
14     public void OnPointerClick(PointerEventData eventData)//event handler
15     {
16         click3DEvent.Invoke(gameObject, eventData);//invoke event
17     }
18 }
19
20
```

修改 Erase Renderer，初始时不添加轮廓（还可以修改不同的颜色）



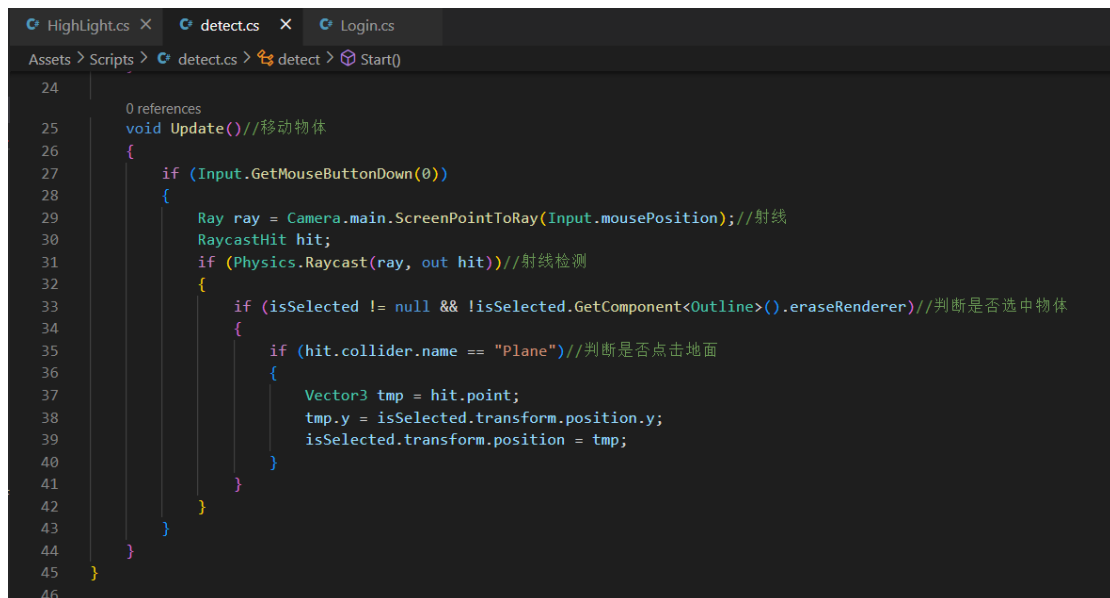
测试，可以正确实现点击时添加轮廓：



### (三) 延伸任务

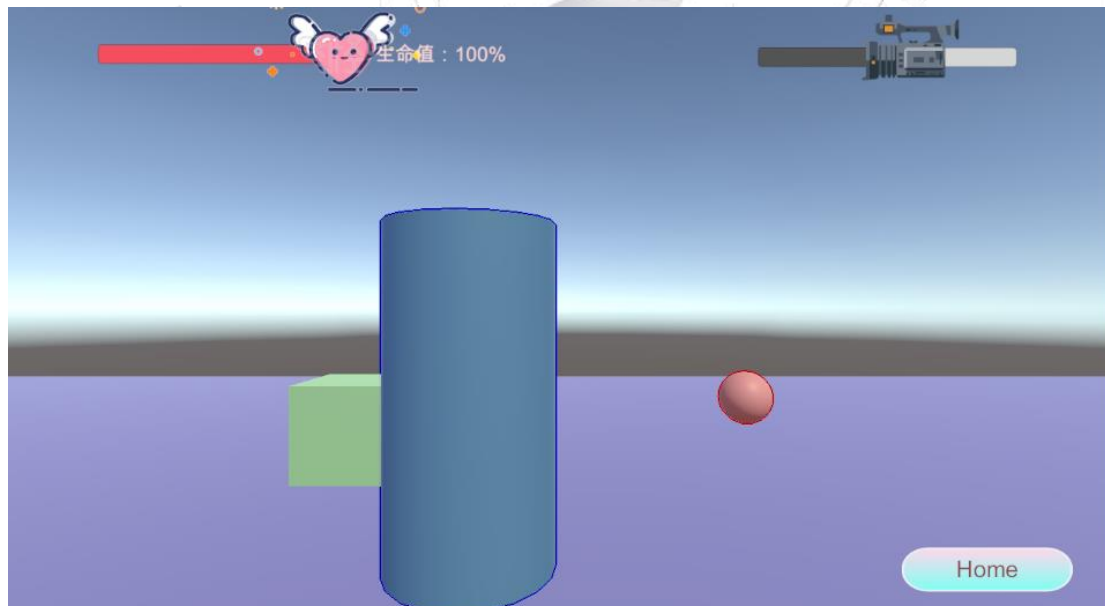
点击地面将当前选中物体移到点击的位置

修改 detect 脚本，添加判断点击是否在 plane 的代码，若点击到 plane 则移动选中的物体。



```
24
25 0 references
26 void Update()//移动物体
27 {
28     if (Input.GetMouseButtonDown(0))
29     {
30         Ray ray = Camera.main.ScreenPointToRay(Input.mousePosition);//射线
31         RaycastHit hit;
32         if (Physics.Raycast(ray, out hit))//射线检测
33         {
34             if (isSelected != null && lisSelected.GetComponent<Outline>().eraseRenderer)//判断是否选中物体
35             {
36                 if (hit.collider.name == "Plane")//判断是否点击地面
37                 {
38                     Vector3 tmp = hit.point;
39                     tmp.y = isSelected.transform.position.y;
40                     isSelected.transform.position = tmp;
41                 }
42             }
43         }
44     }
45 }
46
```

测试可以移动物体：



具体展示参见随报告附的演示视频！



## 五、 实验心得总结:

这次实验主要是关于 Unity 3D 游戏界面基本设计流程以及 UGUI 元素的应用。通过实验,我掌握了 Unity 3D 游戏界面基本设计的流程。首先,在 Unity 中创建一个新项目,然后选择相应的场景并添加游戏对象。接下来,根据需要设计游戏界面的布局和功能,使用 Unity 的图形用户界面 (UGUI) 元素来实现这些功能。同时,我了解了各种 UGUI 基础元素,例如按钮,文本框,图像等等,并且熟悉了 UGUI 事件系统。

UGUI 事件系统是 Unity 中非常重要的一部分,因为它允许开发者将交互性添加到游戏界面中。在实验中,我学会了如何设置和使用 UGUI 事件系统来响应用户的输入。例如,当用户点击按钮时,可以使用 UGUI 事件系统来检测到并触发相应的事件。除此之外,我还理解了 Canvas 组件,它是实现 UGUI 的一个基础元素。Canvas 可以被认为是一个平面,我们可以在其上面添加游戏对象,它们可以作为 UI 元素来显示。

在这次实验中,我还学习了添加物体轮廓和移动物体的方法,这让我在未来对物体交互的设计更加得心应手。

通过本次实验,我进一步了解了 Unity 3D 游戏界面设计的基本流程和 UGUI 元素的应用。我相信这些知识将对我未来的游戏开发工作有很大的帮助。

