



*Virtual Reality (VR)*  
&  
*Augmented Reality (AR)*

# 虚拟现实与增强现实技术

刘石坚 博士  
C2-306 计算机科学与技术教研室  
liusj2003@126.com



# 第4讲 脚本编程初步

---



# 一、基本概念

---

# 概念

- **游戏对象 (GameObject)**
  - Unity中的基本类型
- **脚本 (Script)**
  - 控制游戏对象的代码
  - `GameObject ← AddComponent(Script)`
  - 将**类**以**组件**的形式关联至游戏对象: 运行App时该脚本中定义的类将被实例化执行.
- **继承/重载**
  - `Awake()`, `Start()`, `Update()` .....



# C# (读作C Sharp) 脚本

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class MobileCamera : MonoBehaviour
6  {
7      // Start is called before the first frame update
8      void Start()
9      {
10         -----
11     }
12
13     // Update is called once per frame
14     void Update()
15     {
16         -----
17     }
18 }
```

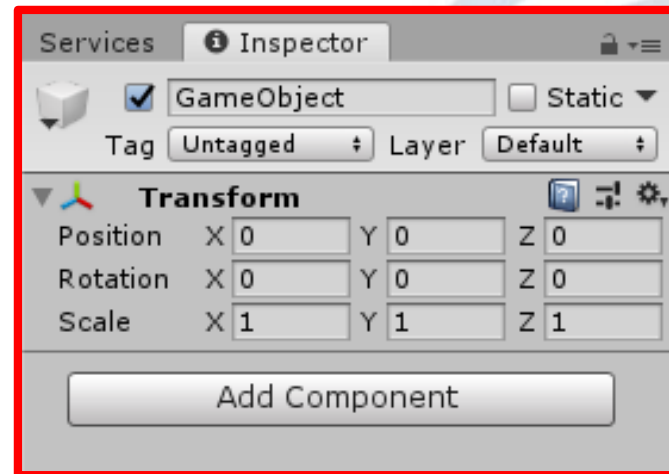
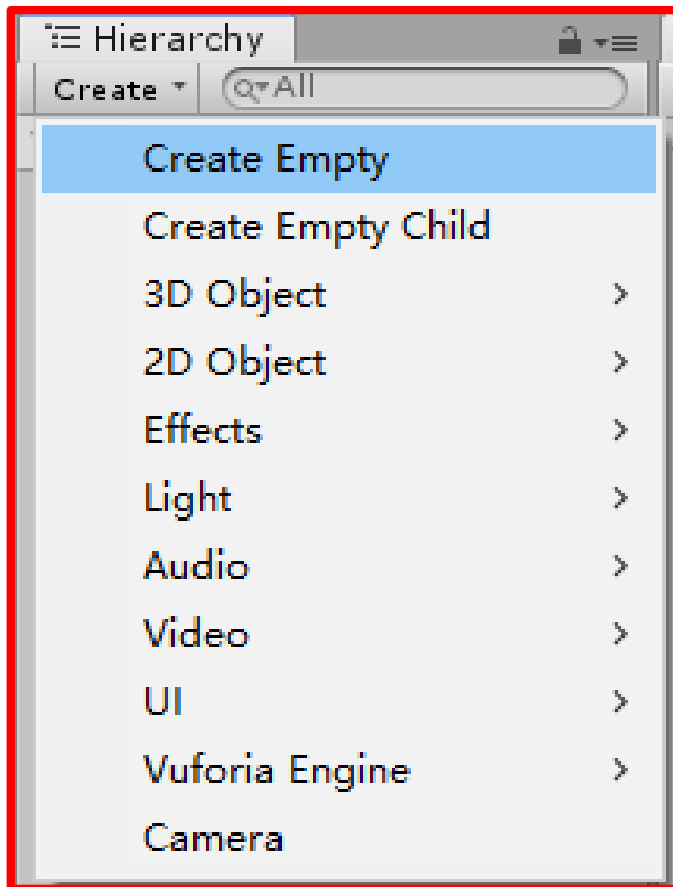


## 二、自定义组件

---

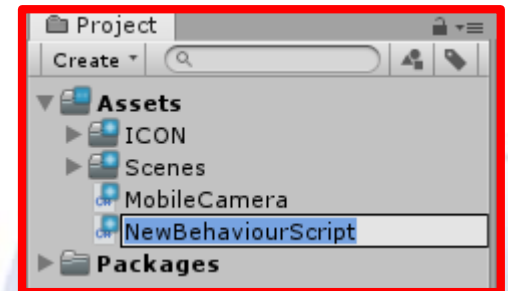
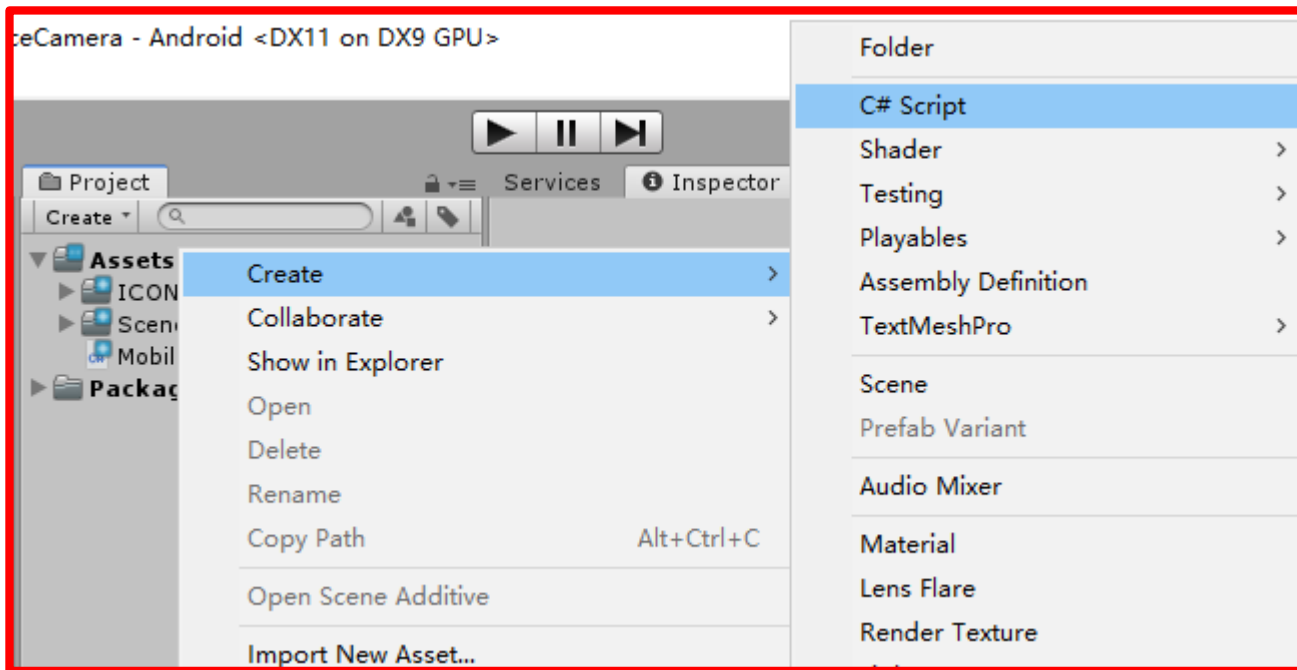
# 1 创建对象

- Hierarchy窗口→Create→ .....



## 2 创建脚本

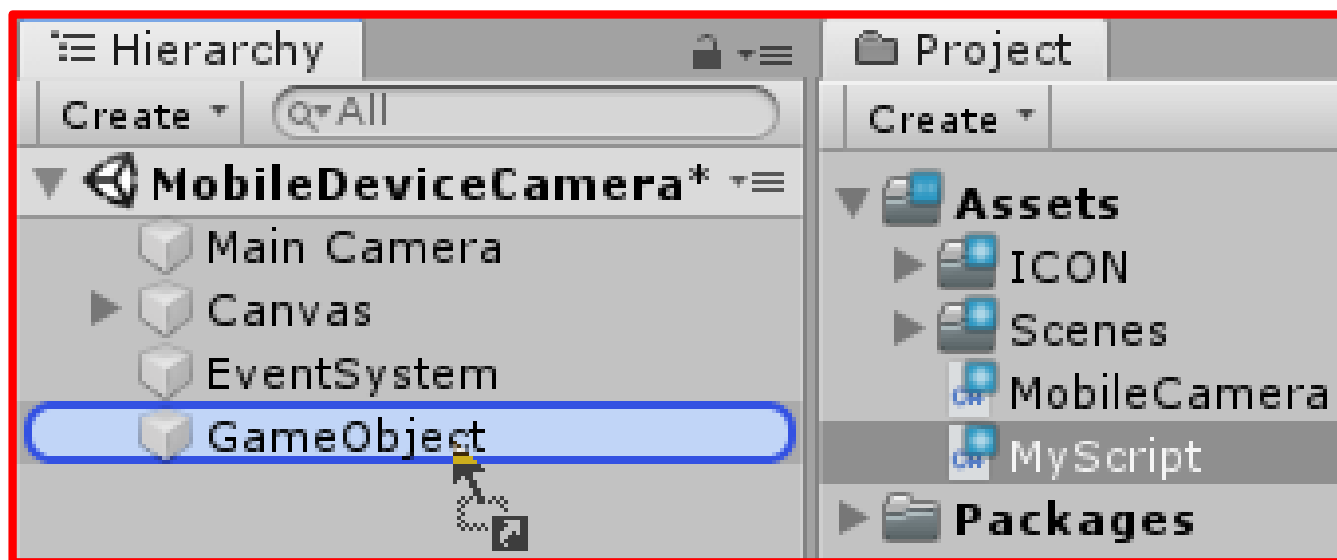
- **Project窗口→Create→C# Script**





## 3 挂载脚本

- 方法1
  - 拖拽

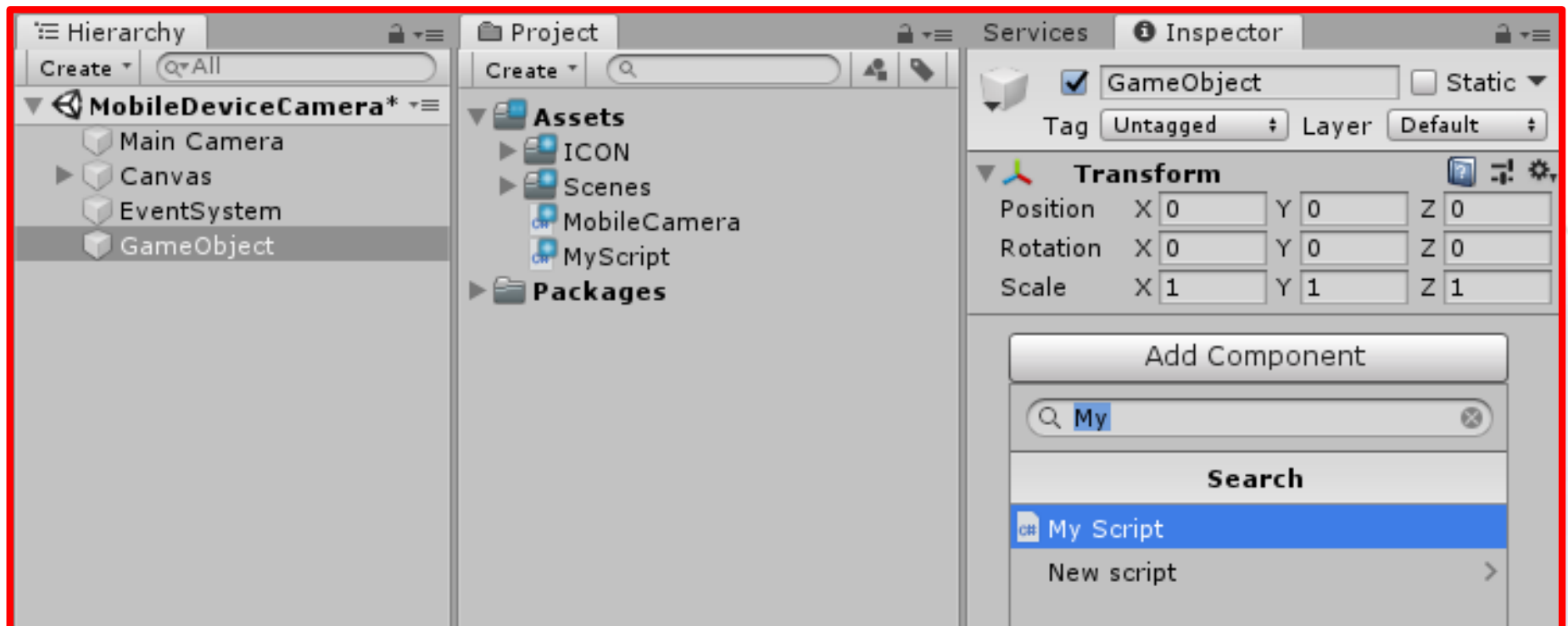


### 3 挂载脚本

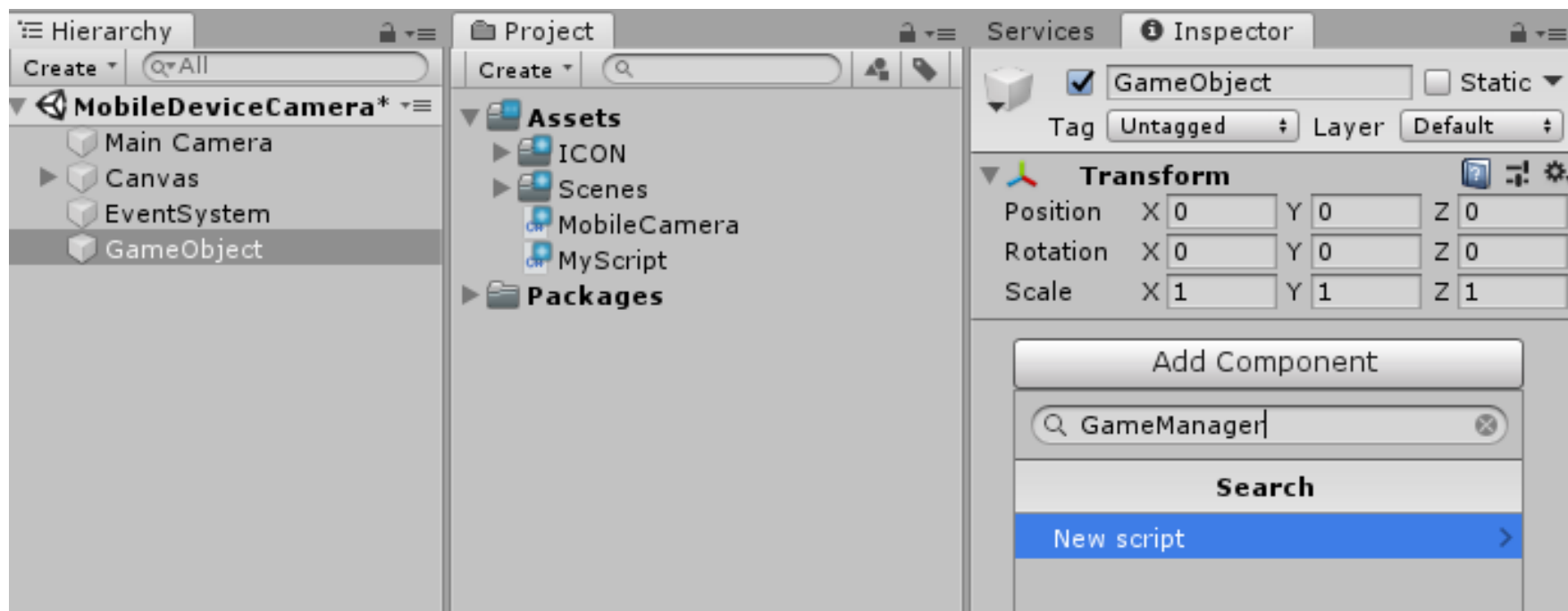
### 方法3?

手工交互实现 vs 编码实现

- 方法2
  - Inspector窗口



# 一步到位



## 思考:

- 同一个脚本是否可以挂载到多个不同的游戏对象?
- 自定义脚本 vs 系统脚本(例如Button).





## 三、脚本编程

---



# 目标

- **场景由很多游戏对象组成**
  - 如何获取对象？
- **游戏对象具有脚本组件来实现**
  - 如何获取指定组件？
- **游戏对象以层次结构进行组织**
  - 如何获取孩子/父亲对象的组件？

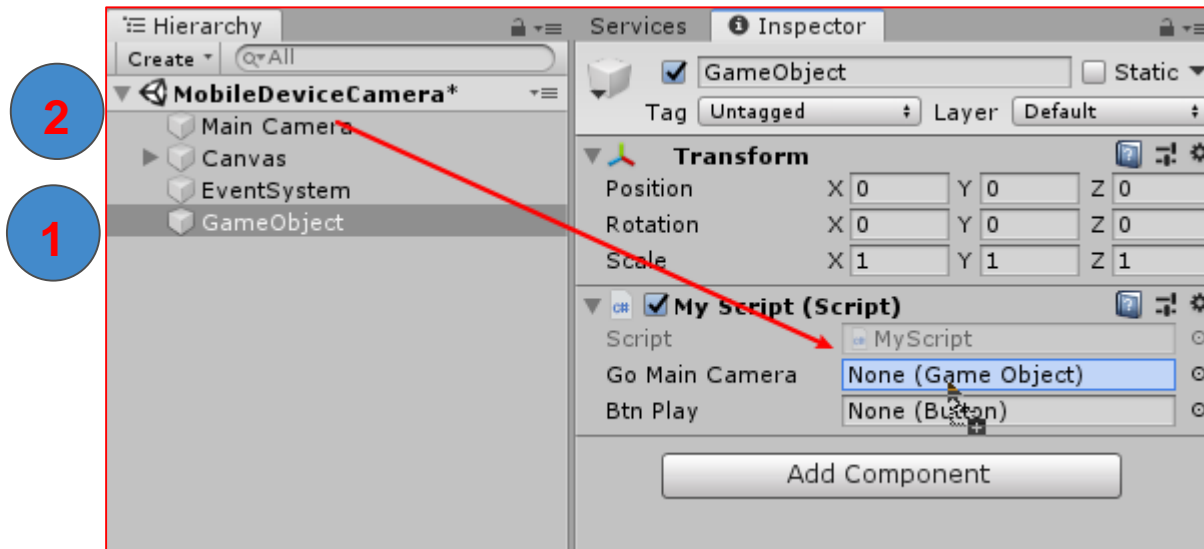


# 1 获取场景中的对象

- 在脚本中申明目标对象

```
public GameObject myGameObject;  
public Button btnPlay;  
Text txtMyText;  
...
```

- 在Editor中将目标对象拖拽至组件的指定位置



问：  
txtMyText  
去哪了？

2

1

3

# 1 获取场景中的对象

- 在脚本中申明目标对象

```
public GameObject myGameObject;  
public Button btnPlay;  
Text txtMyText;  
...
```

- 使用Find方法 (点击查看API文档) `GameObject.Find();`

```
GameObject GameObject.Find(string name)  
Finds a GameObject by name and returns it.
```

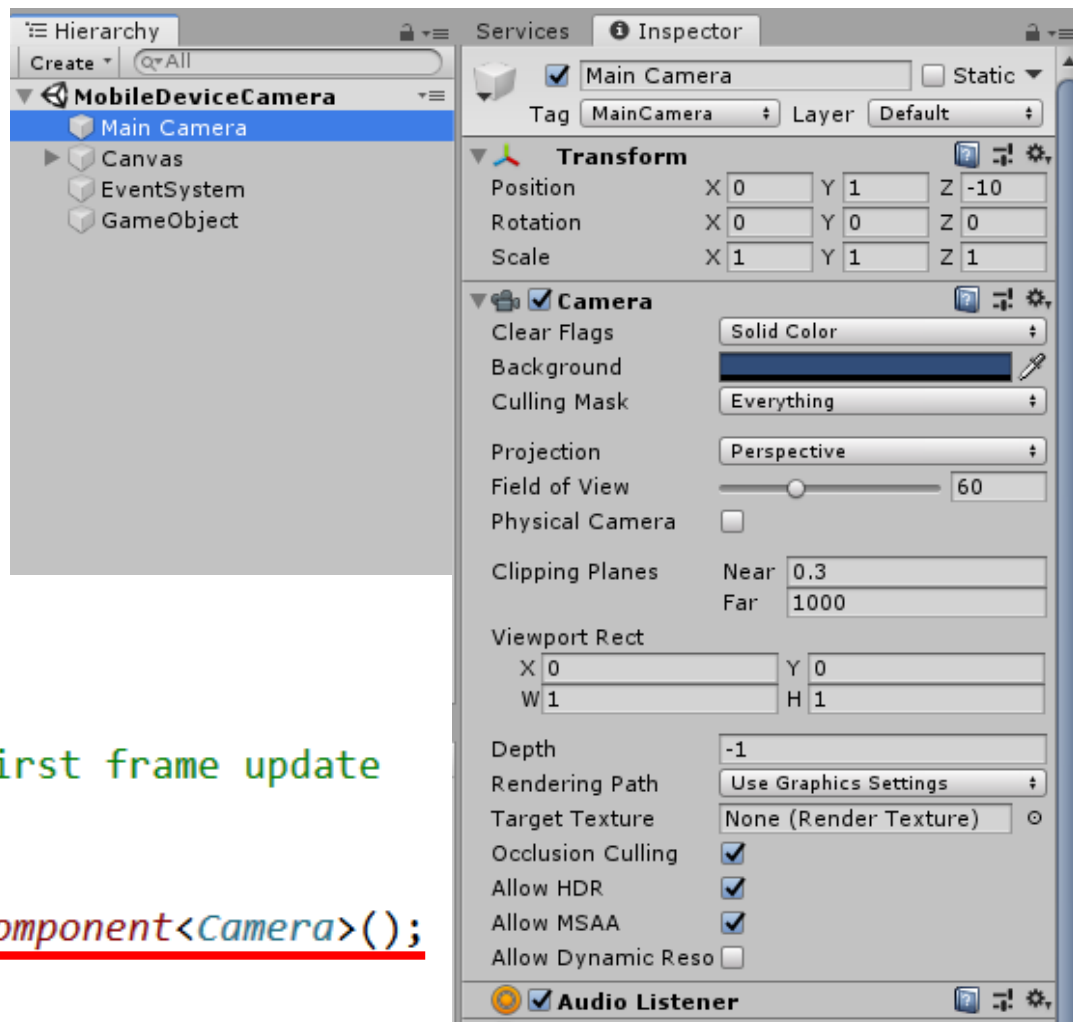
```
Text txtMyText;  
// Start is called before the first frame update  
void Start()  
{  
    txtMyText = GameObject.Find("txtMyText").GetComponent<Text>();  
}
```

## 2 获取指定对象的组件

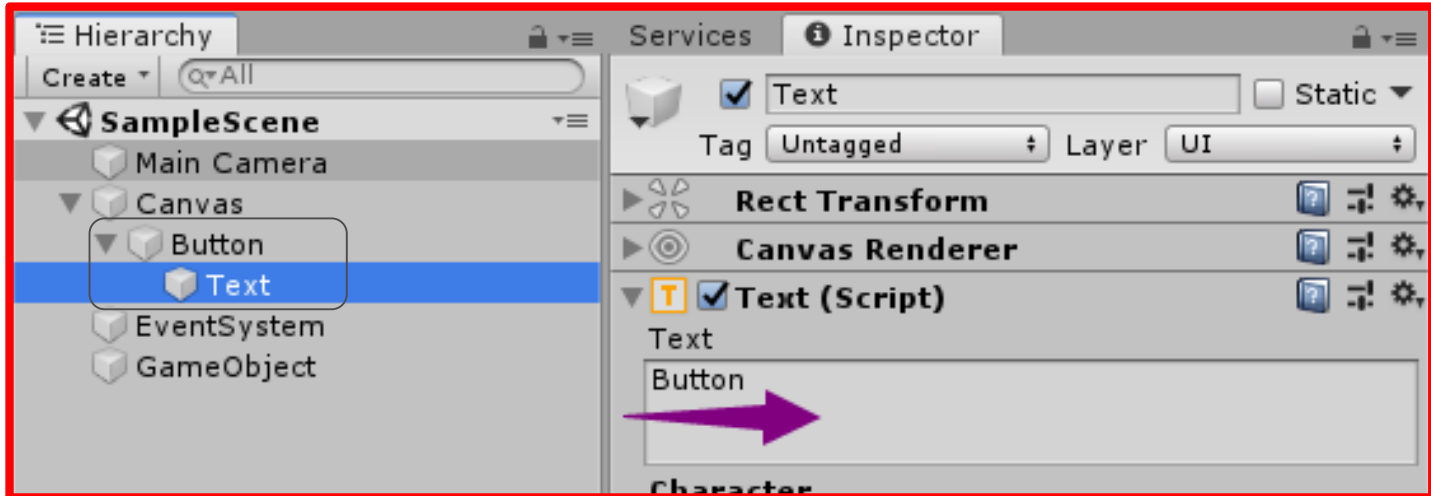
- 添加组件
  - AddComponent
- 获取组件
  - GetComponent

```
public GameObject goMainCamera;  
Camera mainCam;
```

```
// Start is called before the first frame update  
void Start()  
{  
    mainCam = goMainCamera.GetComponent<Camera>();  
}
```

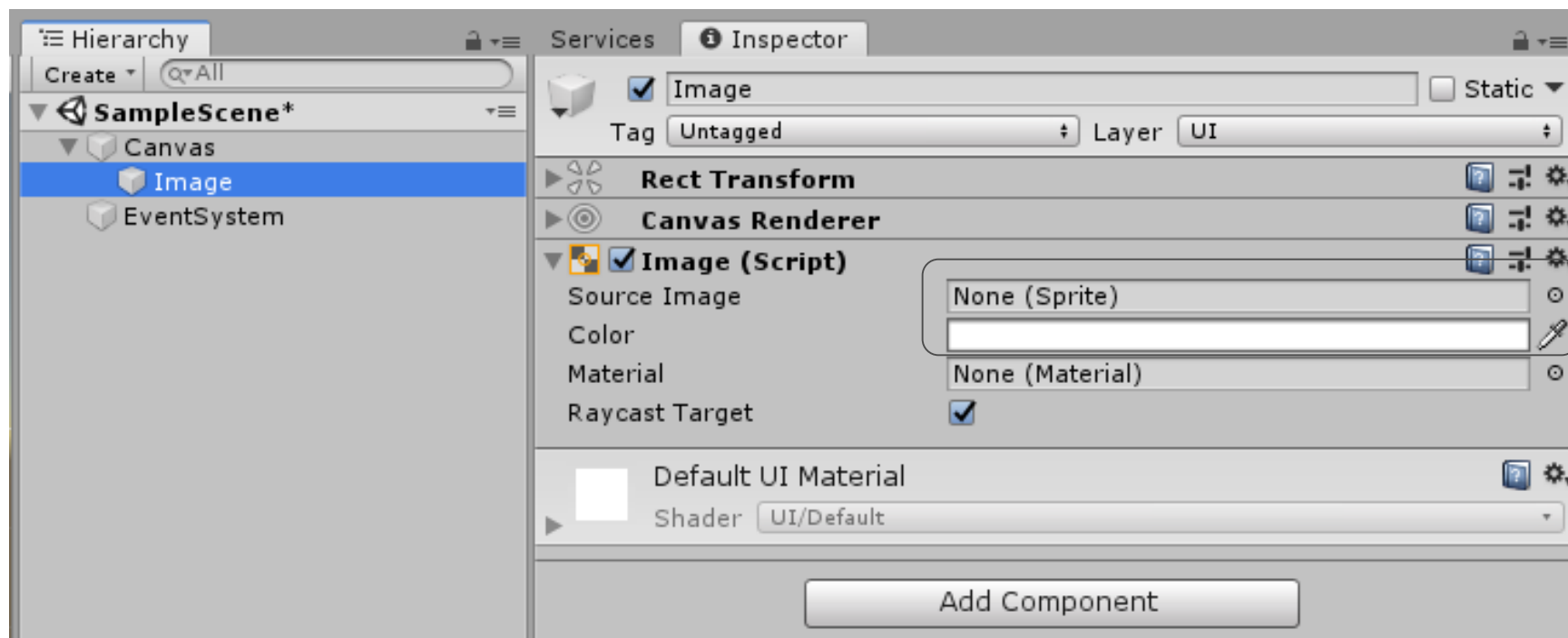


# 扩展

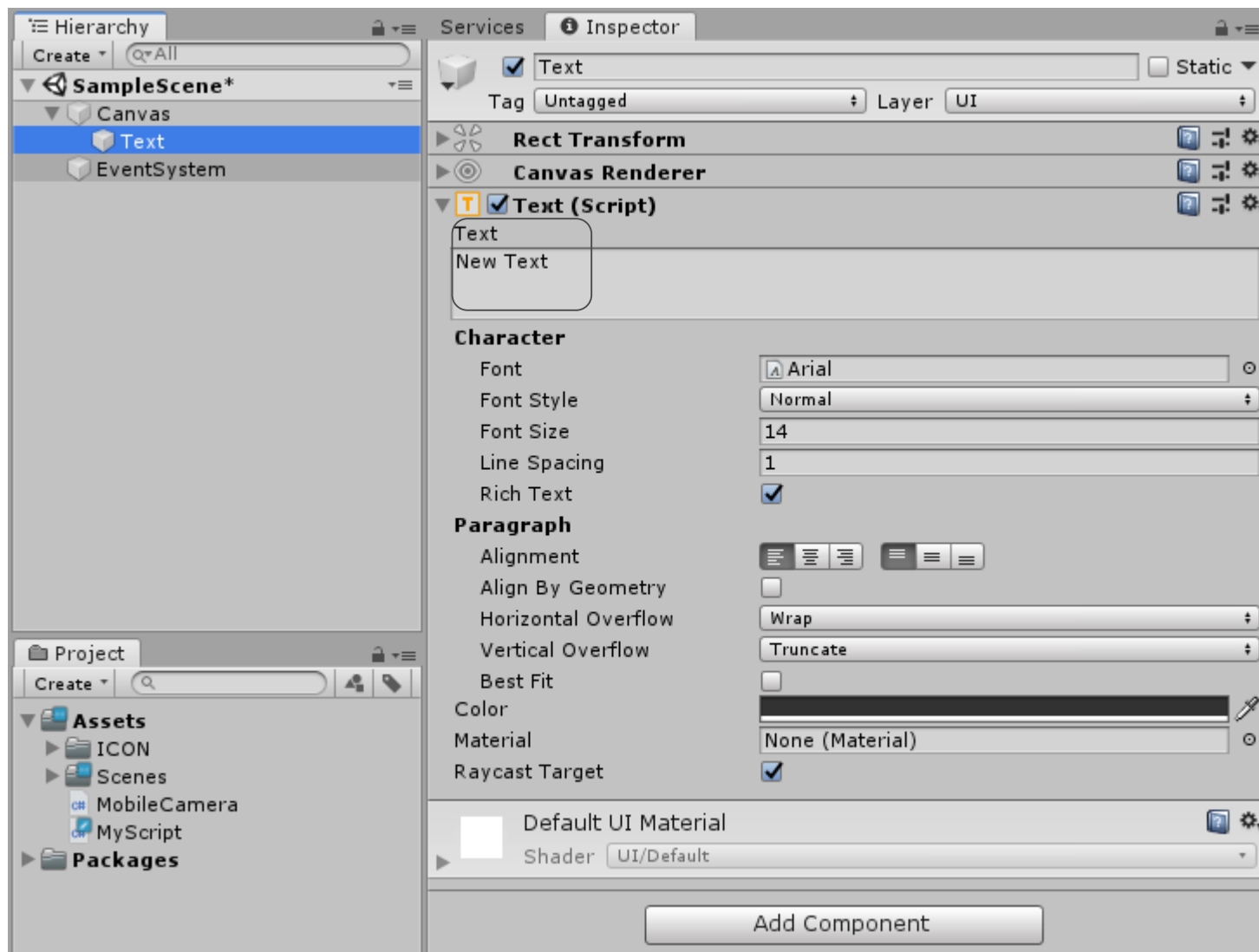




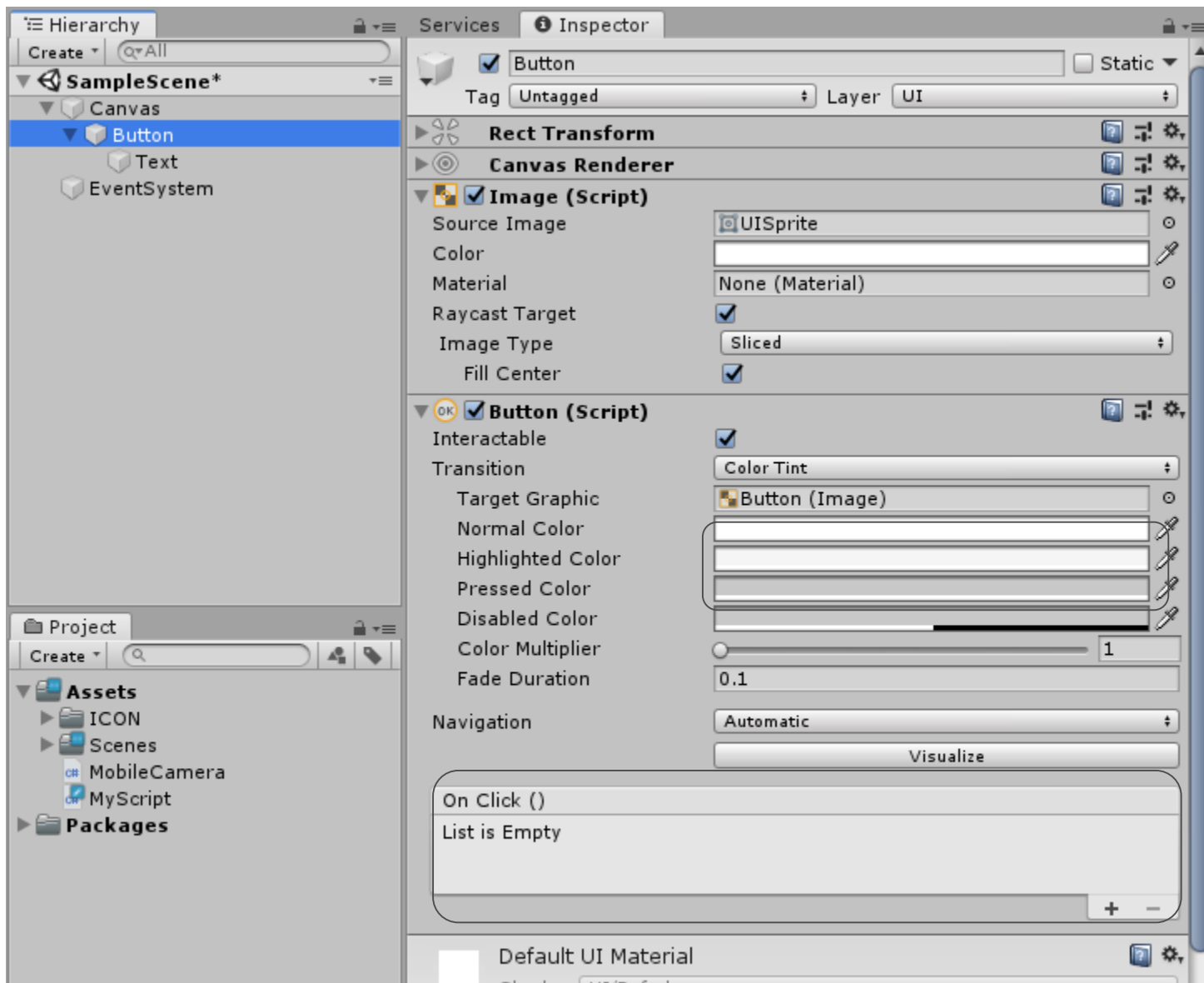
# 常见系统组件 - GUI



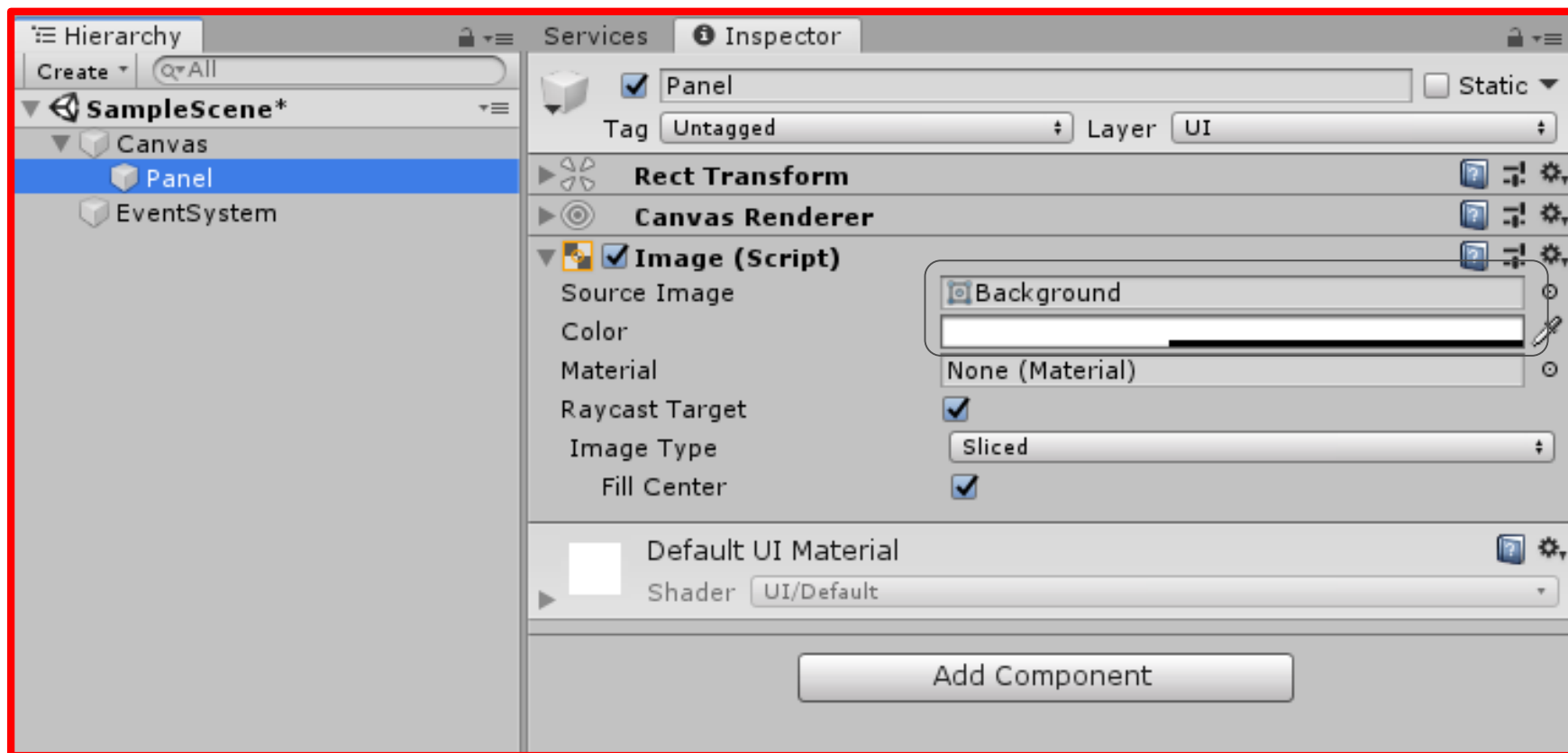
# 常见系统组件 - GUI



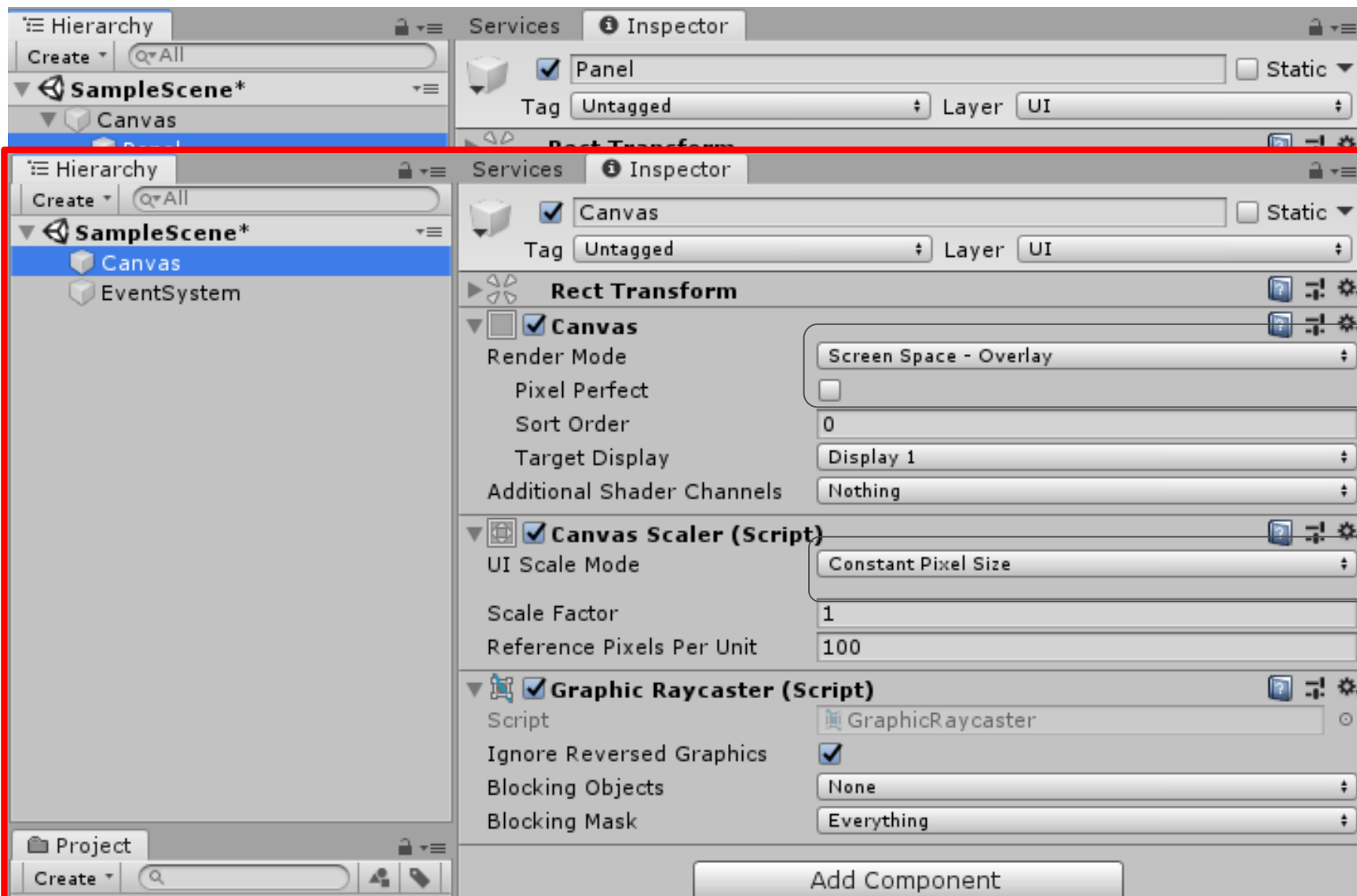
# 常见系统组件 - GUI



# 常见系统组件 - GUI

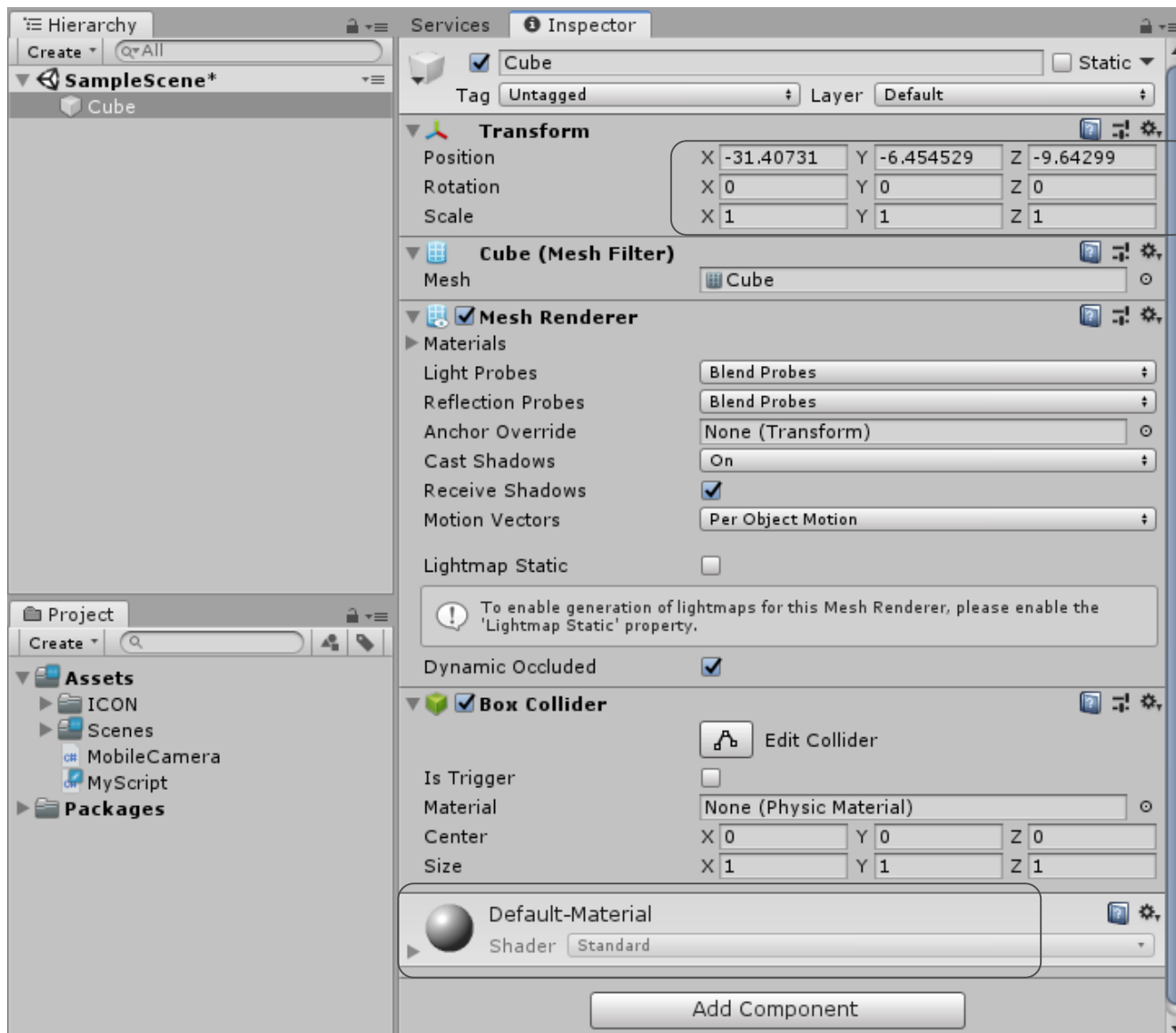


# 常见系统组件 - GUI





# 常见系统组件 - 3D对象





**Thank you !**