



Unity 快速入门

By karl07

目录



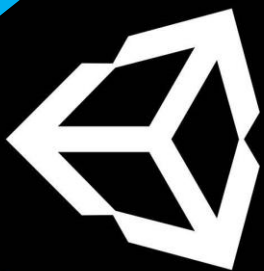
1. 下载与安装

2. Unity界面概览

3. Unity与C#脚本

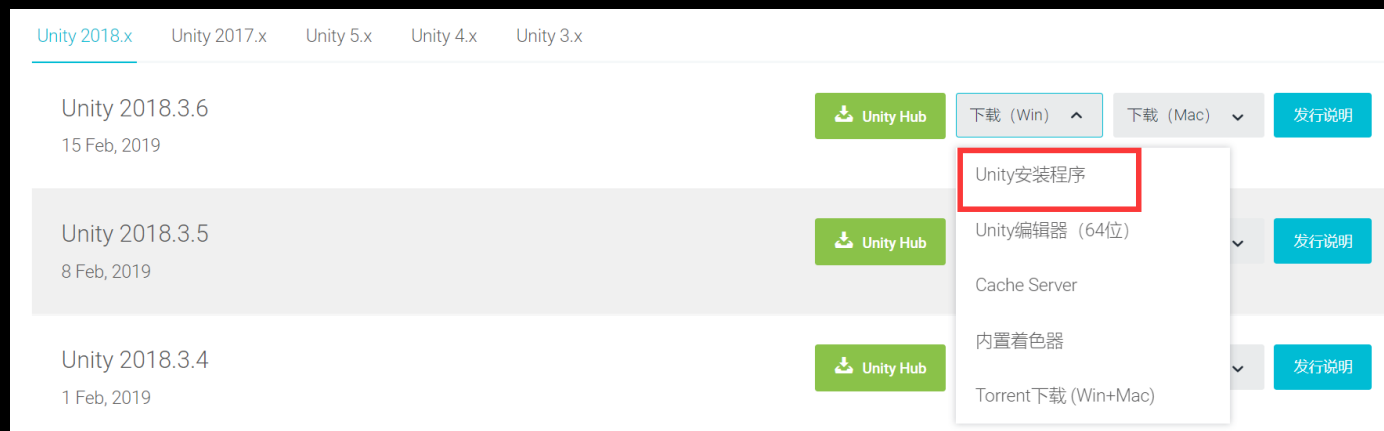
4. Unity从入门到放弃

下载与安装



<https://unity3d.com/cn/get-unity/download/archive>

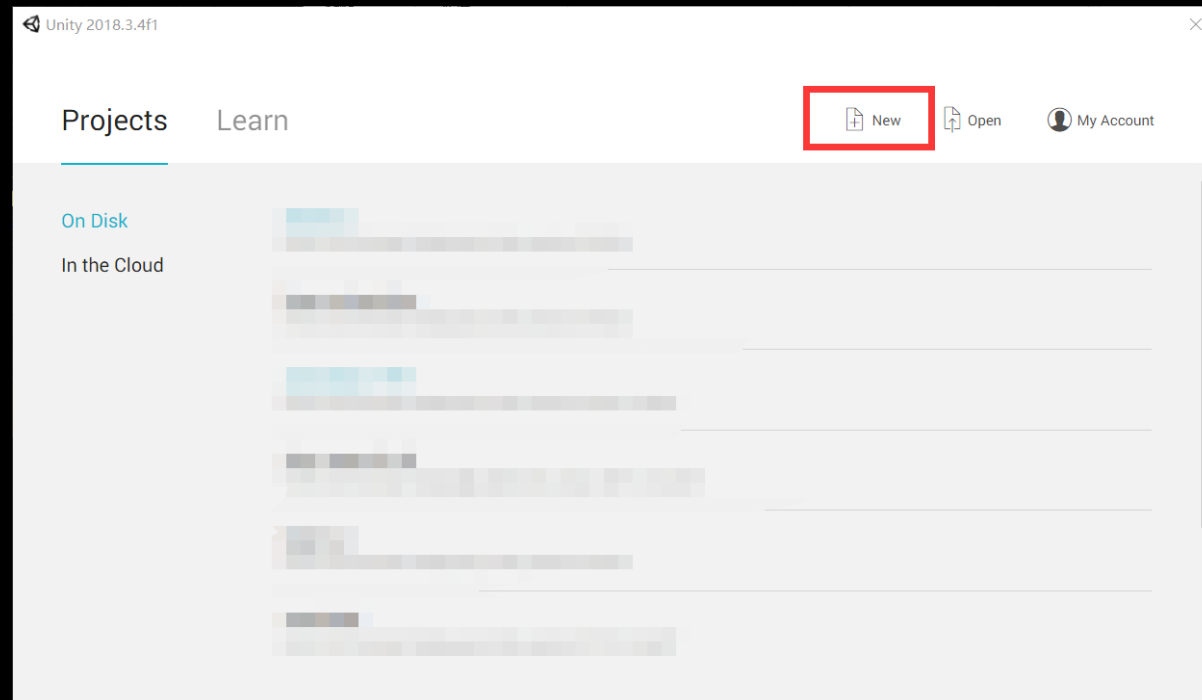
进入链接选择版本操作系统然后直接安装就好了



下载与安装

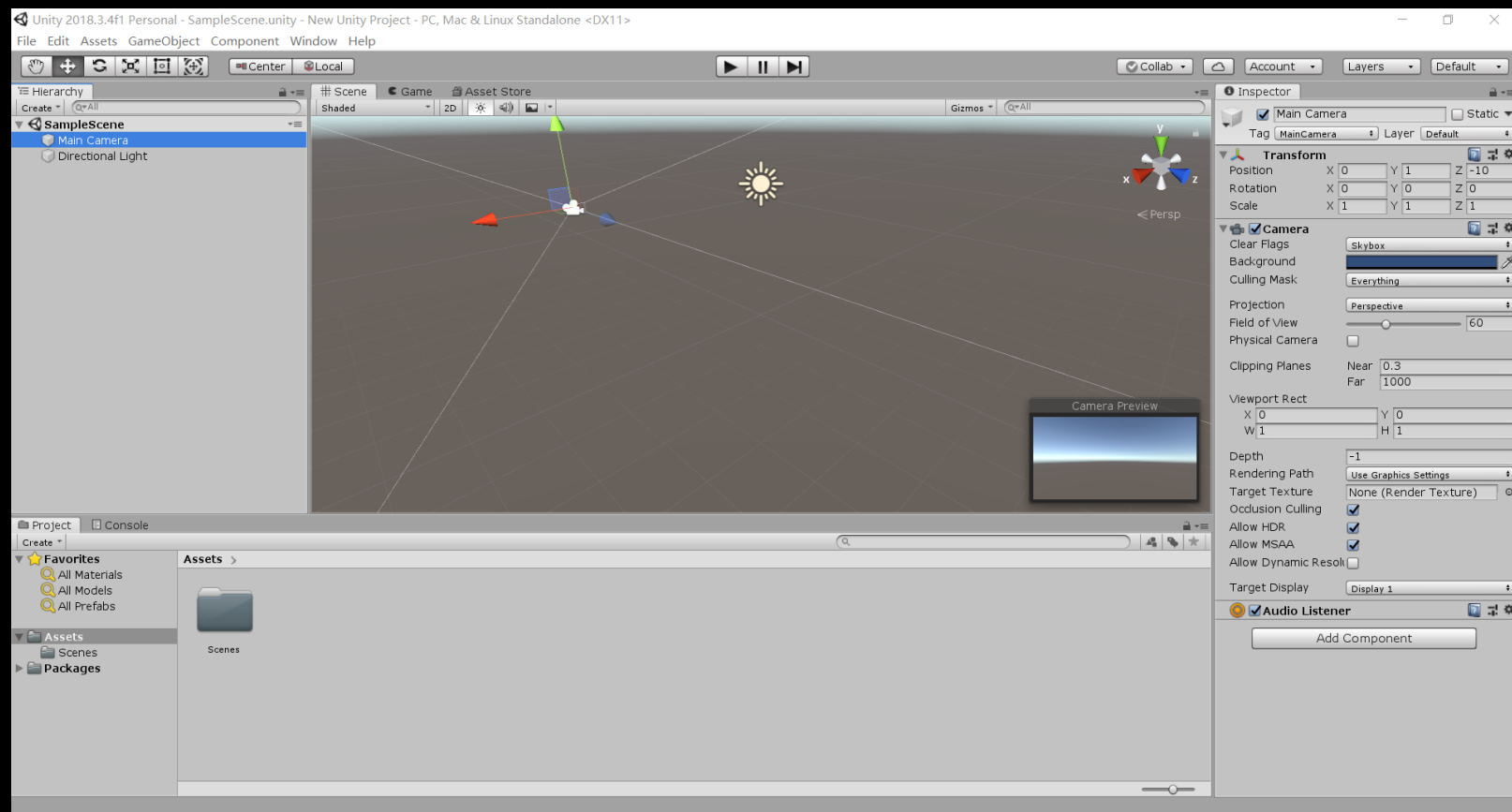
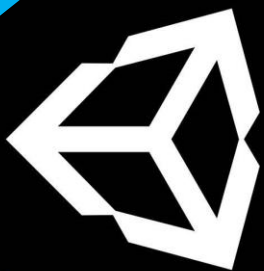


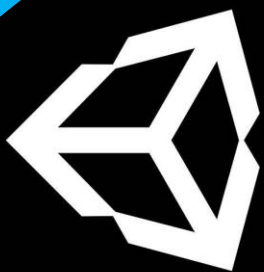
安装完应该就是这样 可能要注册账号什么的 点new创建新项目



Unity界面概览

然后新项目创建完应该长这样

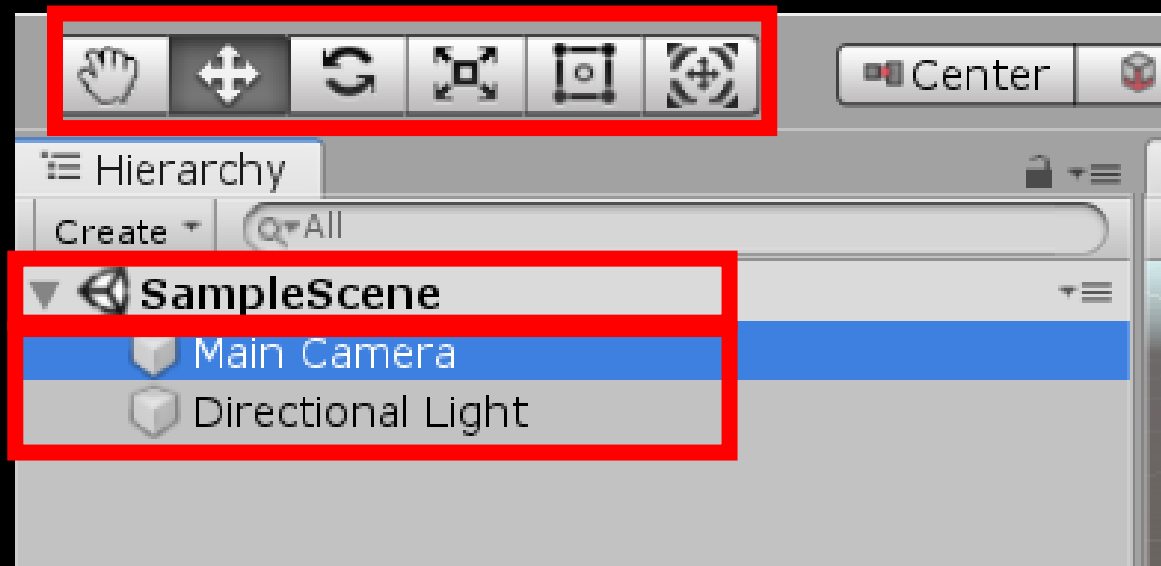




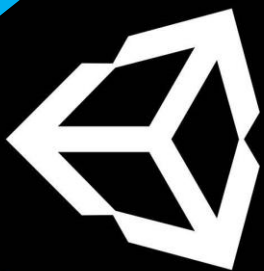
Unity界面概览

上面一排是用来操作场景和场景里的物体的

场景
(Scenes)
物体
(GameObject)

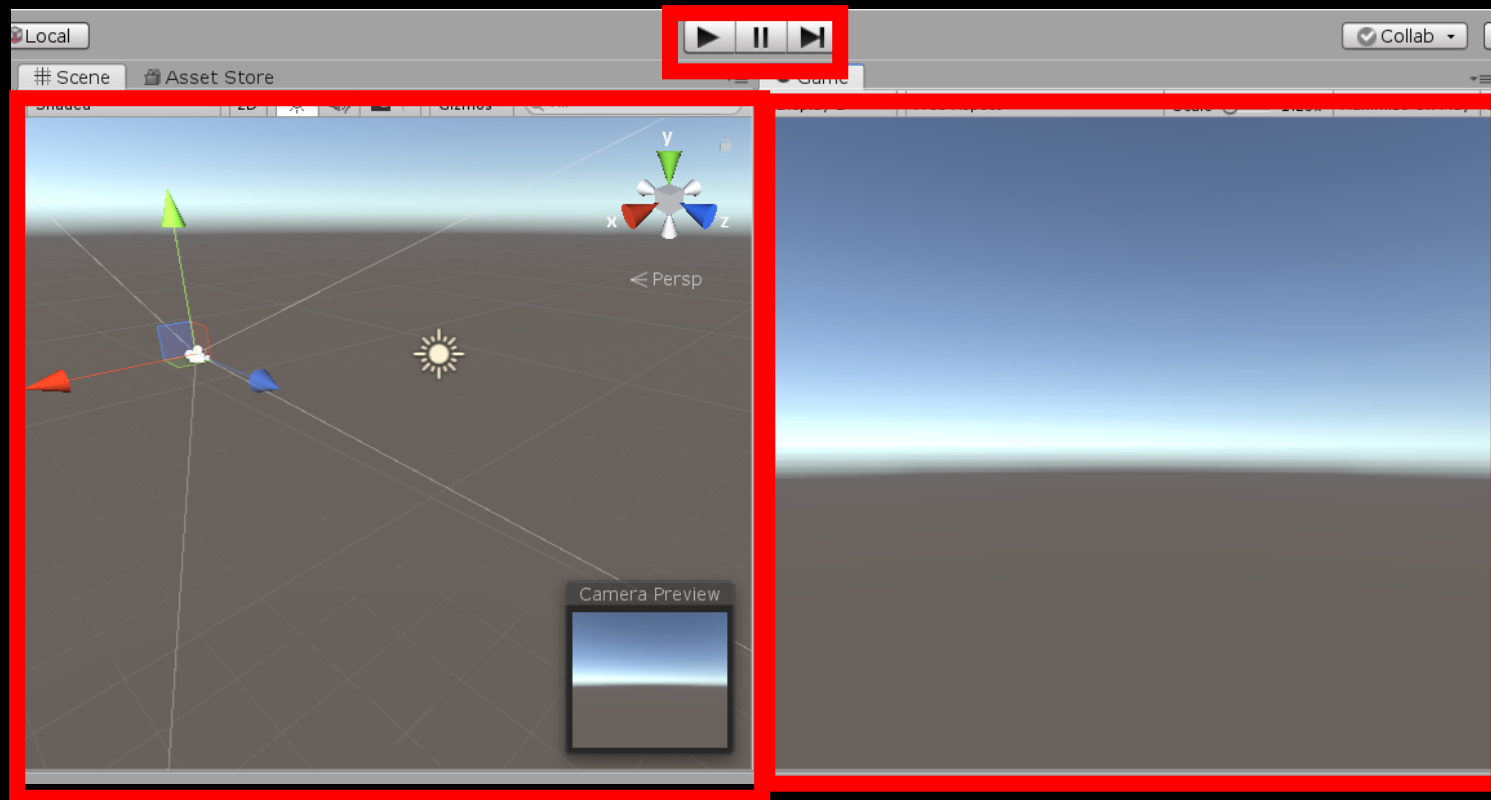


Unity界面概览



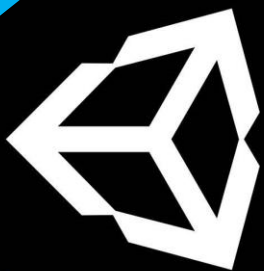
Scenes窗口
整个游戏场景

运行 暂停 断点



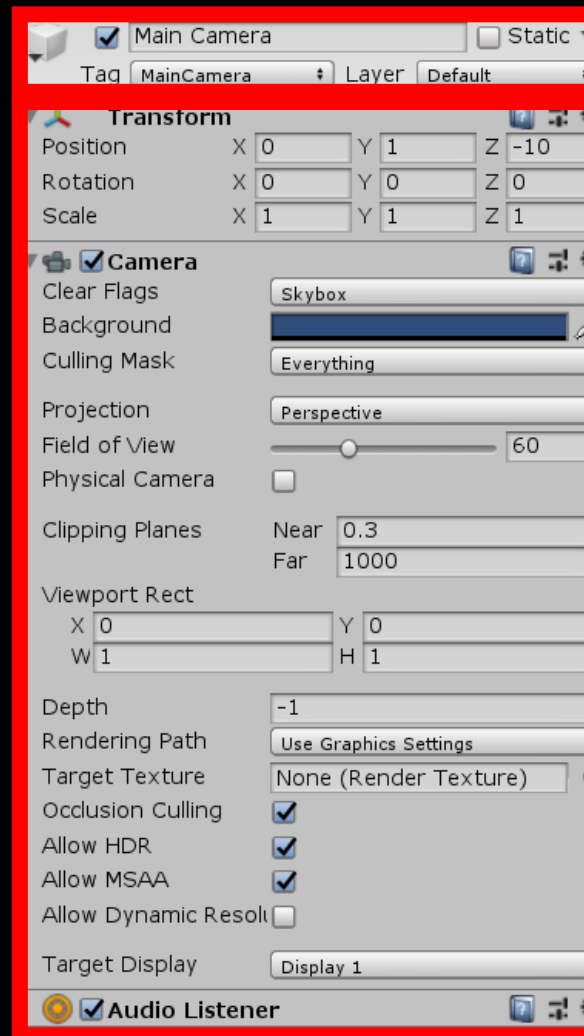
Game窗口
游戏运行时
看到的窗口

Unity界面概览

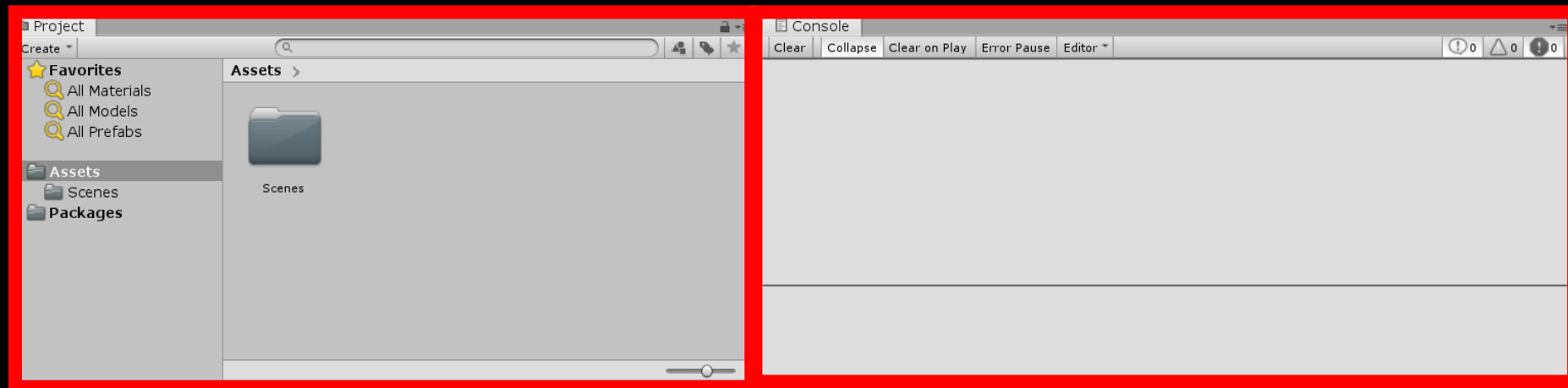
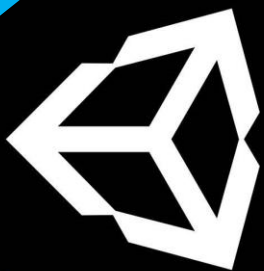


GameObject

组件
(Components)



Unity界面概览



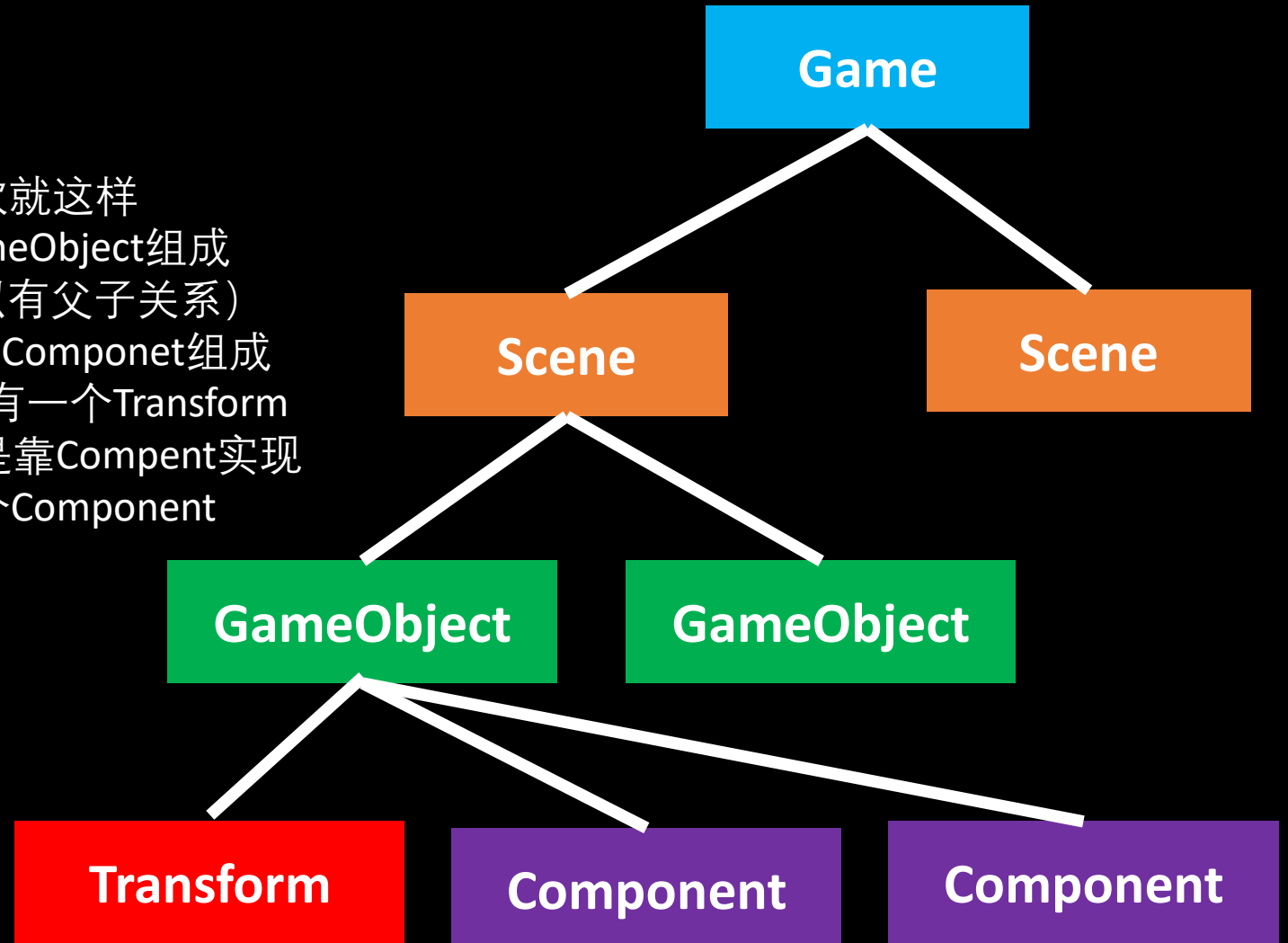
资源窗口
游戏用到的资源都在这里
在这里右键可以创建脚本

控制台
调试时候用的

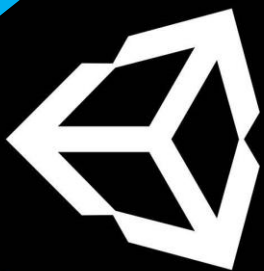
Unity与C#脚本



Unity大概的层次就这样
一个Scene由多个GameObject组成
(GameObject本身可以有父子关系)
然后每个GameObject由Componet组成
GameObject本身有且只有一个Transform
而GameObject的功能都是靠Compent实现
C#脚本也是作为一个Component



Unity与C#脚本

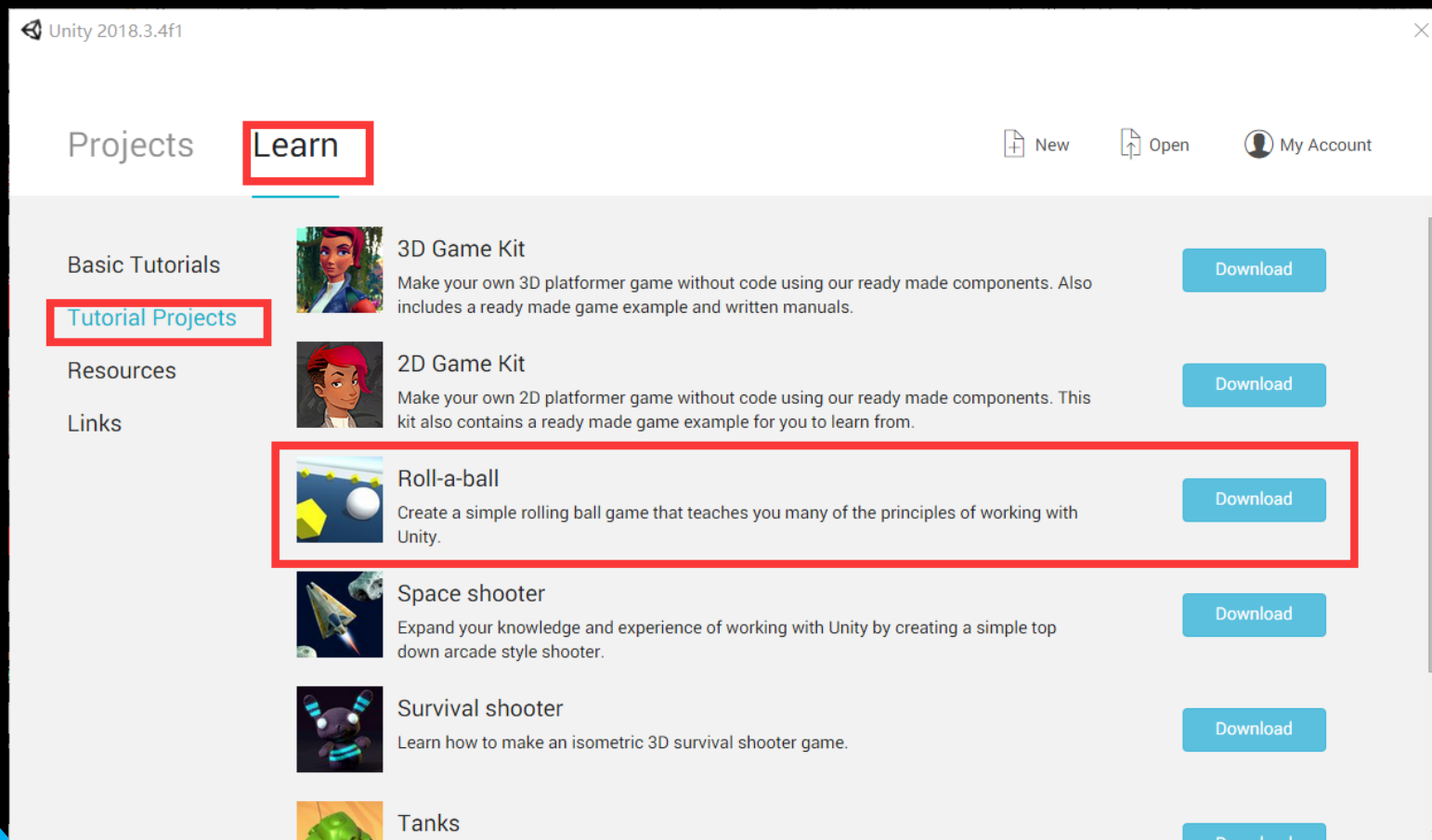
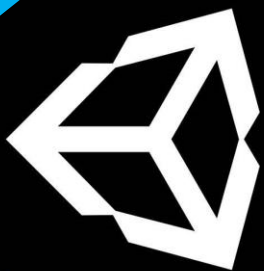


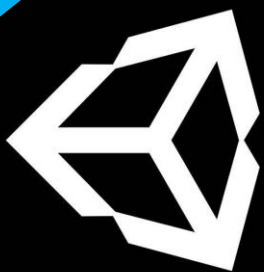
C#脚本一般继承自MonoBehaviour
一个.cs文件对应一个类
当这个脚本被挂在GameObject上时会实例化
实例化的对象作为一个Component
Awake可以理解为这个类的构造函数
Start是这个Component的初始化函数
Update是在游戏循环内每一帧执行的函数

```
5 public class Test : MonoBehaviour
6 {
7     private void Awake()
8     {
9     }
10 }
11 // Start is called before the first frame update
12 void Start()
13 {
14 }
15
16 // Update is called once per frame
17 void Update()
18 {
19 }
20
21 }
22 }
```

Unity与C#脚本

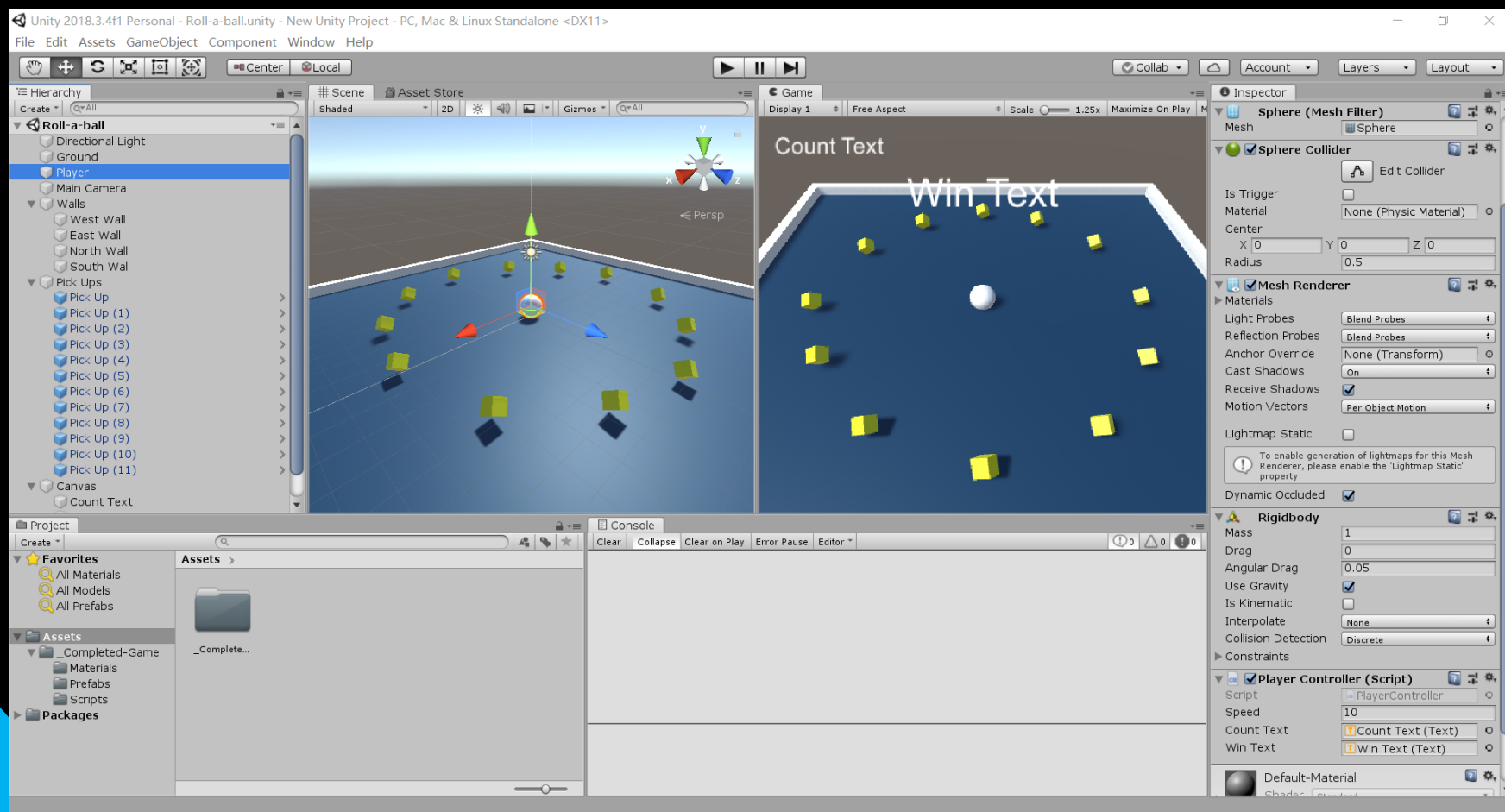
举个栗子



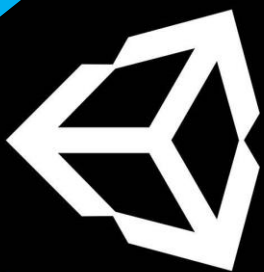


Unity与C#脚本

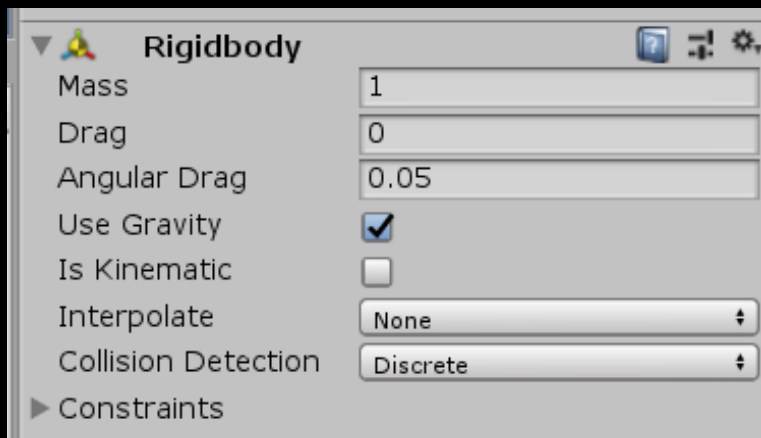
举个栗子
分析一下这个项目里的Player



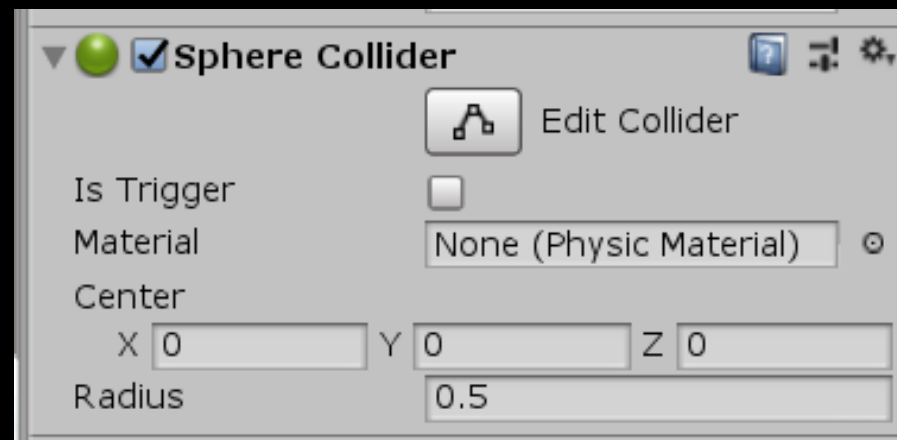
Unity与C#脚本



举个栗子
分析一下这个项目里的Player



Rigidbody是一个让小球有物理属性的Component

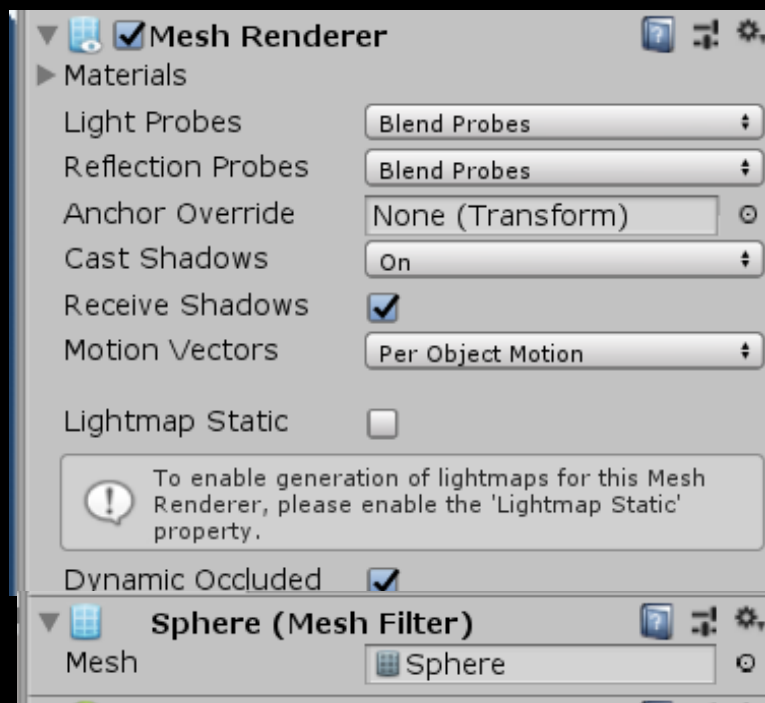
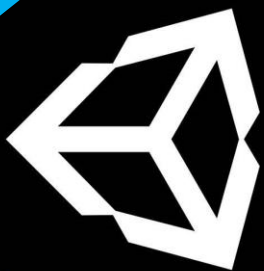


Colider是一个碰撞检测器
Is Trigger就是这个碰撞器能否穿透
(比如这个栗子的黄色方块的colider就是trigger)

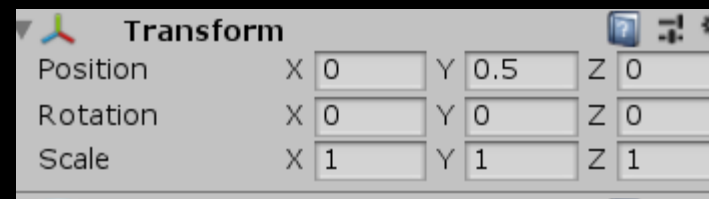
Unity与C#脚本

举个栗子

分析一下这个项目里的Player



这两个组件描述了Player的形状
并且渲染在摄像机上



Transform是位置旋转大小等信息

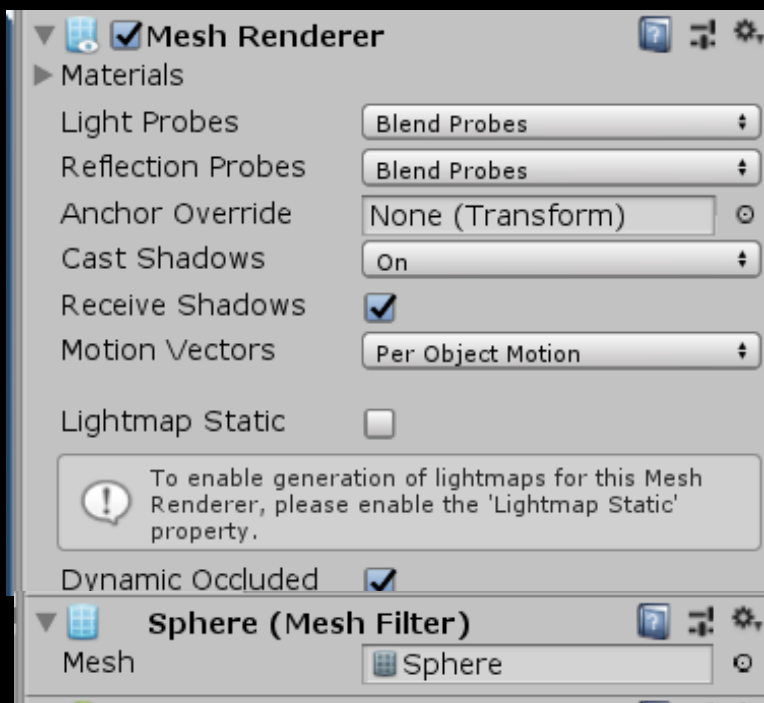
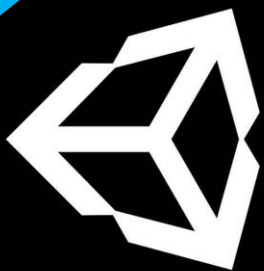


脚本也是作为组件

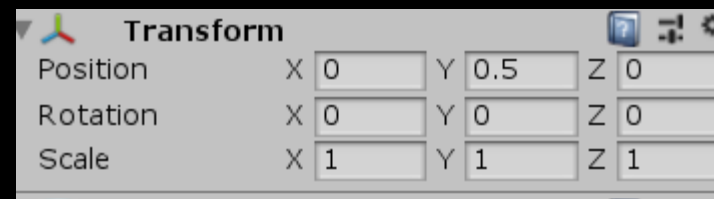
Unity与C#脚本

举个栗子

分析一下这个项目里的Player



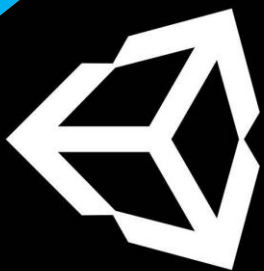
这两个组件描述了Player的形状
并且渲染在摄像机上



Transform是位置旋转大小等信息



脚本也是作为组件



Unity与C#脚本

注释很全没啥好解释的

C#语法跟C++还算类似吧

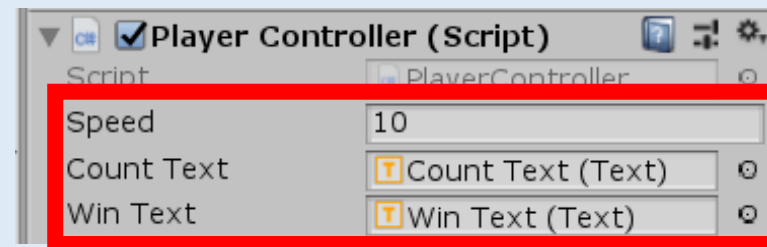
(C#似乎没有指针 所有对象默认引用类型)

不知道为啥缩进被吃了

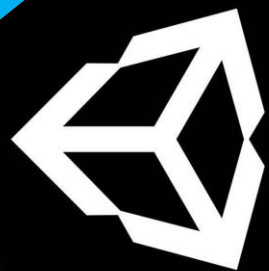
为什么不直接在VS里面看呢 (因为我翻译了一下)

```
using UnityEngine;
// Include一些Unity封装的名空间
using UnityEngine.UI;
using System.Collections;

public class PlayerController : MonoBehaviour {
// 创建public数据 public的成员可以在编辑器编辑拖动配置
public float speed;
public Text countText;
public Text winText;
//创建private数据
private Rigidbody rb;//物理刚体
private int count; //分数计数器
```



Unity与C#脚本



```
//初始化函数
void Start ()
{
    //这是和其他Component交互的方式
    rb = GetComponent<Rigidbody>(); //获取Rigidbody的引用方便操作
    count = 0; // 分数初始化
    SetCountText (); // 初始化UI
    winText.text = ""; // 初始化UI
}
```

Unity与C#脚本



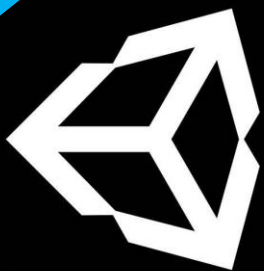
```
// 这是一个每固定时间更新一次的update 一般用在物理计算
void FixedUpdate ()
{
// 获取输入信息具体的可以在菜单栏->Edit->ProjectSetting里面找到
float moveHorizontal = Input.GetAxis ("Horizontal");
float moveVertical = Input.GetAxis ("Vertical");
// 根据输入信息产生一个向量
Vector3 movement = new Vector3 (moveHorizontal, 0.0f,
moveVertical);
// 对Rigidbody加一个力
rb.AddForce (movement * speed);
}
```

Unity与C#脚本



```
// 碰撞检测 之前说过黄色的方块的Collider就是一个Trigger
// OnTriggerEnter是碰撞后会执行的函数 other就是另一个碰撞
// 物体的Collider
void OnTriggerEnter(Collider other)
{
    //判断碰到的Collider的gameObject是不是黄色方块
    if (other.gameObject.CompareTag ("Pick Up"))
    {
        // 使物体消失
        other.gameObject.SetActive (false);
        //分数++
        count = count + 1;
        // 更新UI
        SetCountText ();
    }
}
```

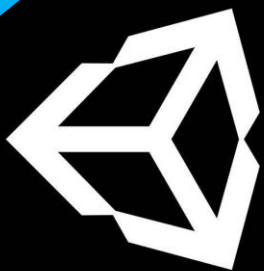
Unity与C#脚本



```
// 更新UI的函数
void SetCountText ()
{
    // 修改UI的文字
    countText.text = "Count: " + count.ToString ();

    // 获胜条件判断
    if (count >= 12)
    {
        // 修改UI文字
        winText.text = "You Win!";
    }
}
}
```

Unity从入门到放弃



更详细的实现教程（上）：

<https://www.cnblogs.com/OctoptusLian/p/8418534.html>

更详细的实现教程（下）：

<https://www.cnblogs.com/OctoptusLian/p/8529313.html>

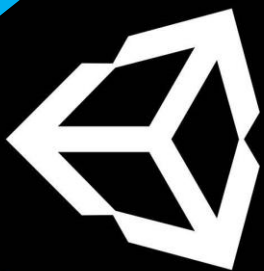
类图（也可以在VS里F12查看继承关系这种）：

http://www.cnblogs.com/bitzhuwei/p/unity-core-types.html#_label0

函数执行顺序：

<https://blog.csdn.net/u012130706/article/details/79393527>

Unity从入门到放弃



官方教程:

<https://unity3d.com/cn/learn/tutorials>

官方API:

<https://docs.unity3d.com/ScriptReference/index.html>

其他教程:

<https://www.xuanyusong.com/>

<http://www.taikr.com/course/explore/Unity3d>

相关书籍:

《Unity 3D游戏开发（第2版）》

<https://book.douban.com/subject/30327646/>