



第3章 Unity3D脚本程序介绍 ——脚本的响应事件与组件访问



调整MainCamera的Tips

■调整Scene和Game视野范围到“好望角”

✓ Scene和Game视野范围一致

- **Hierarchy**: 双击某个GameObject, 快速显现
- 选中GameObject, **Inspector**: Transform精准设置
- 调整Scene→选中MainCamera→“GameObject”菜单下的“Align with view”

✓ 摄像机的旋转（鼠标右键）

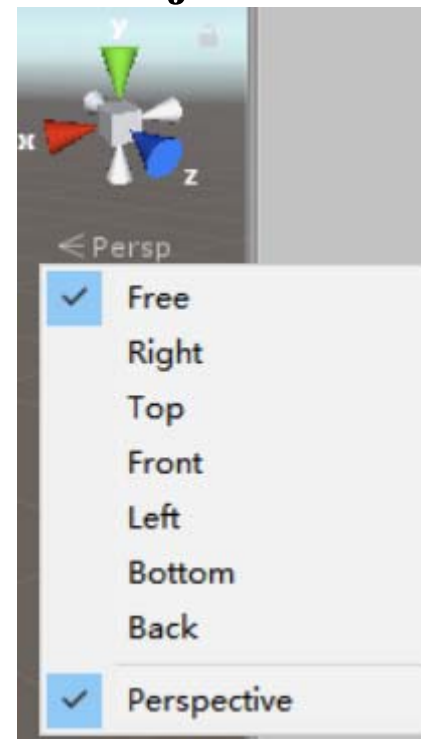
✓ 场景导航（切换视野范围）

- 鼠标右键
+ **W/S(前后)** **A/D(左右)** **Q/E(上下)**

✓ 缩放

- 鼠标中间滑轮(缩放效果)=**W/S(前后)**

模拟实拍摄像机的
“推”、“拉”、“摇”、“移”效果！



游戏呈现给玩家的主要体验

■提升用户粘性和使用体验的部分要素

✓移动、交互性（InputManager）

✓多个GameObject的克隆

➤Instantiate: bullets、批量对象仿真

✓3D物理模拟

➤碰撞器-刚体、触发器-穿越等真实感效果的仿真

✓Audio effects（听觉特效）

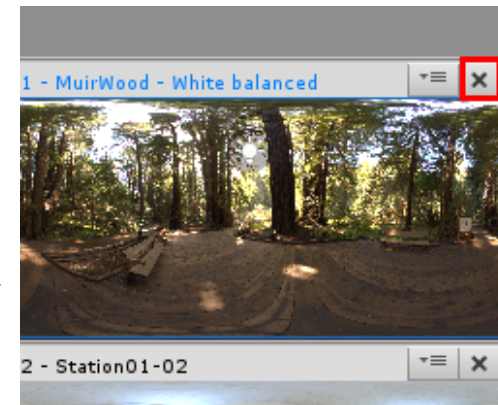
➤高保真、环绕立体声

✓游戏心理学、精美场景、美工

✓Visual effects（视觉特效）

➤High Dynamic Range Image、fps数值高（沉浸感）

<https://docs.unity3d.com/Manual/LookDevHDRVView.html>



本章提要

❖ 3.1 常用脚本API

- GameObject类

- 克隆游戏对象 (Instantiate, Destroy)

- Transform类

- position, Translate, Rotate, eulerAngles

- Time类, Mathf类, Random类

❖ 3.2 脚本开发的输入与控制

- Input (Mouse, button, touch)

- InputManager (Fire1, Jump, ADWS)

- 键盘输入 (Input.GetKey())

3.1.1 GameObject类（克隆游戏对象）

■Instantiate实例化

➤ **Instantiate()** 是Unity提供克隆游戏对象的方法，在游戏中应用比较广泛，而且提高了工作效率，一般常用于发射炮弹、AI敌人等一些完全相同并且数量庞大的游戏对象。

➤格式

- **Instantiate (GameObject)**
- **Instantiate (GameObject, position, rotation)**

➤说明

- **GameObject**指生成克隆的游戏对象，也可以是**Prefab预制体**
- **position**指生成克隆的游戏对象的初始位置，类型是**Vector3**
- **rotation**指生成克隆的游戏对象的初始角度，类型是**Quaternion**

Quaternion.identity: 该四元数,相当于"无旋转":这个物体完全对齐于世界或父轴。

3.1.1 GameObject类（克隆游戏对象）

■ Instantiate步骤

- Assets: 右键“Create”→“Prefab”
- Assets: 右键“Create”→“Material”（设置颜色等）
- Scene中，创建并调整GameObject（如Sphere1），在Hierarchy菜单中将Sphere1推拽到Assets下的Prefab即可
- 可删除Sphere1
- Material既可以赋给静态创建的GameObject或Prefab

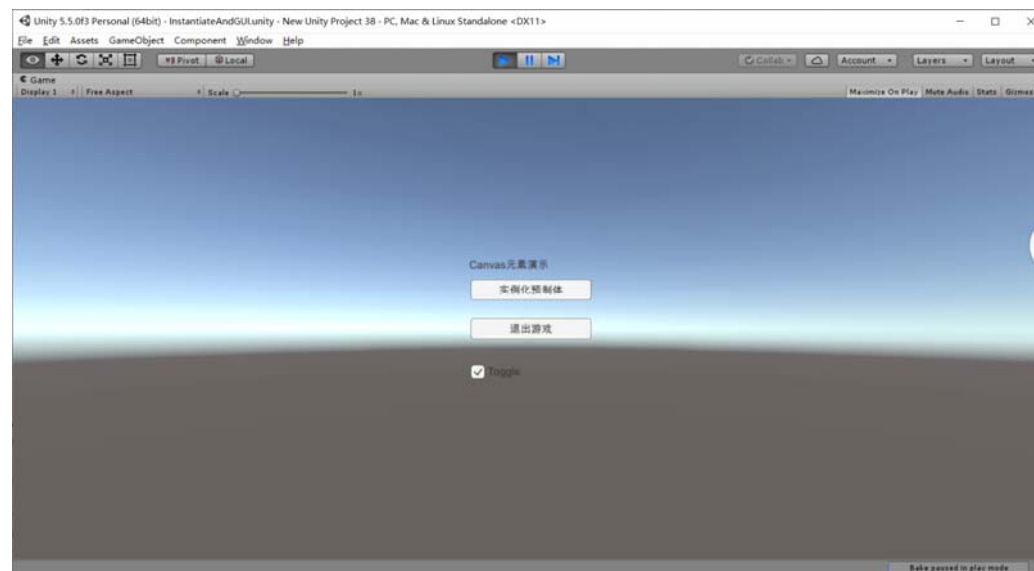
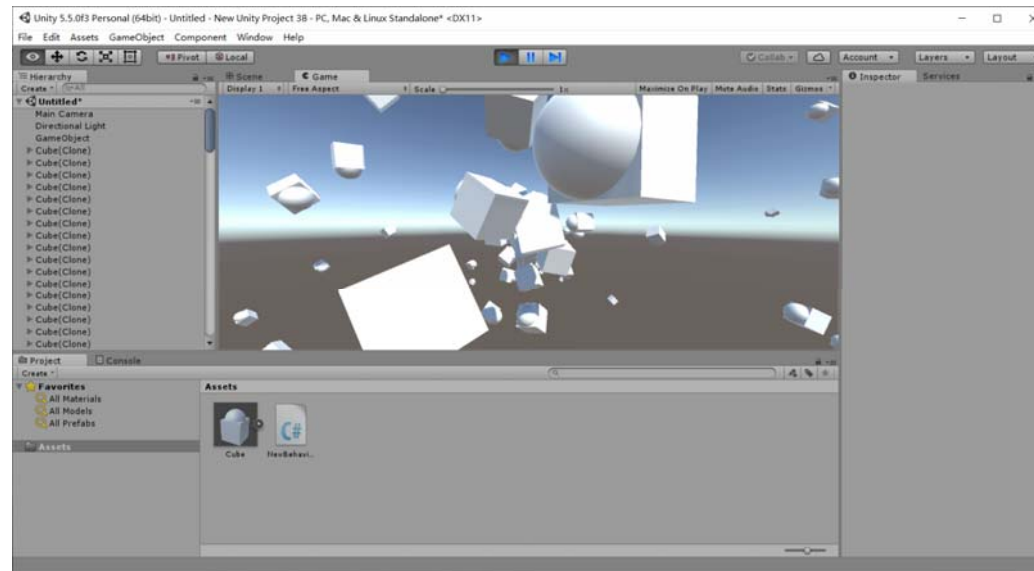
代码演示

Project 38: “NewBehaviourScript.cs”

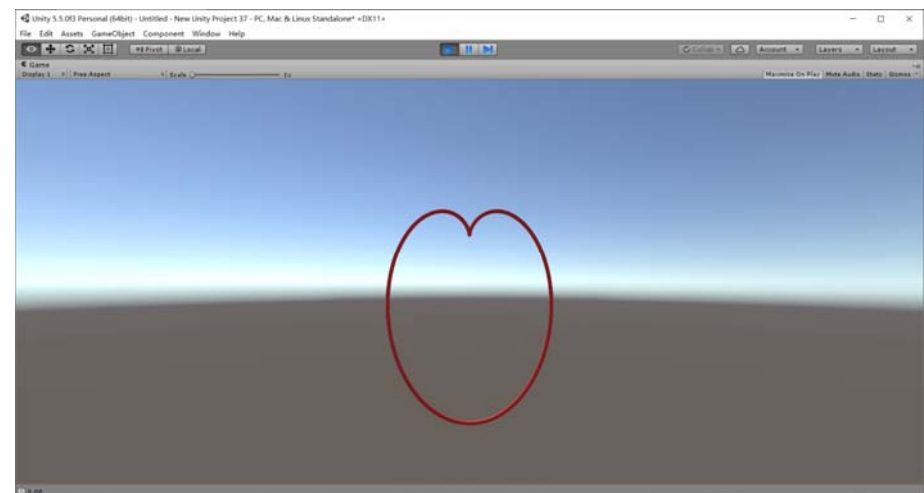
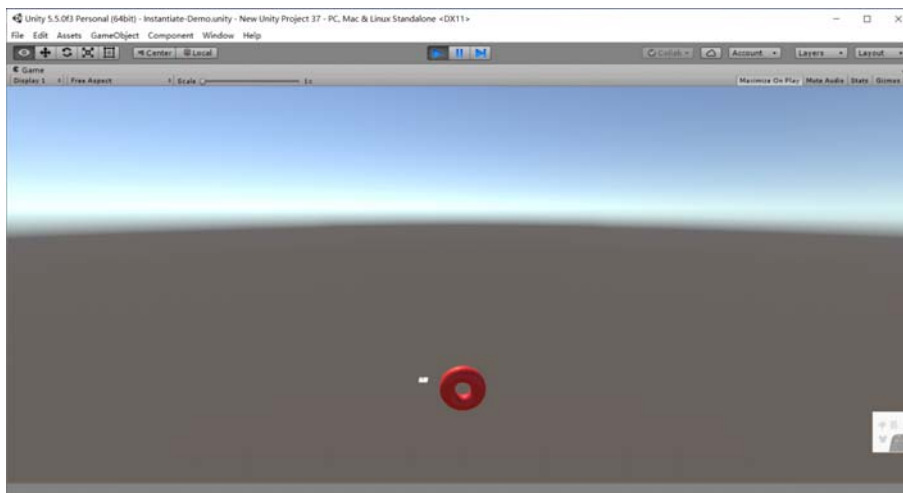
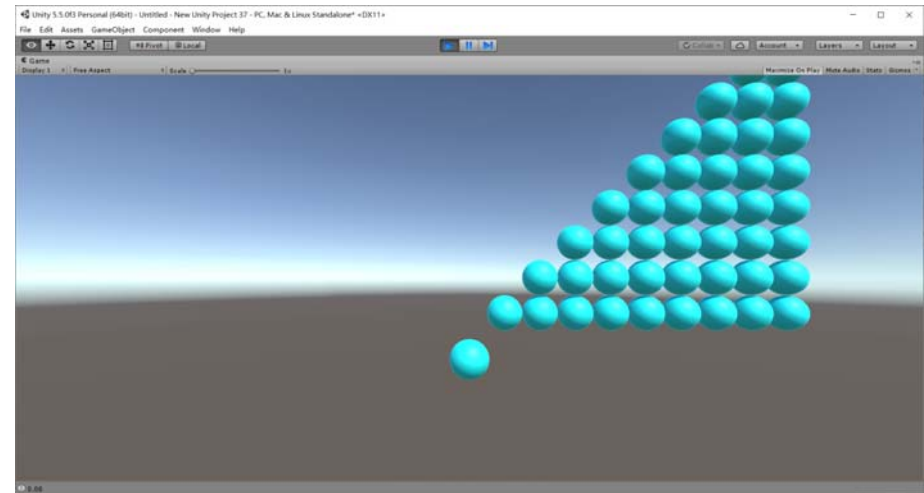
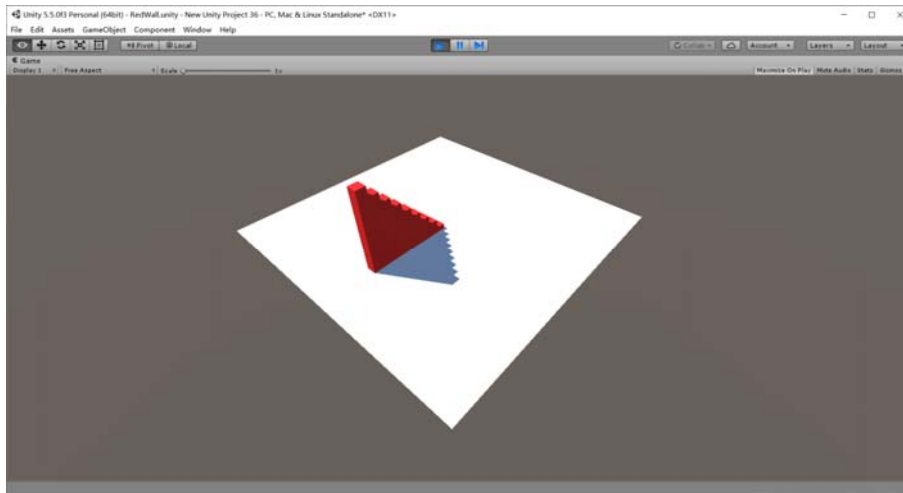
Project 37: “SpecialEffectWithInstantiate.cs”

Application.Quit()、UI、阶梯效果、环形效果、心脏线（爱心线）

3.1.1 GameObject类（克隆游戏对象）-续1



3.1.1 GameObject类 (克隆游戏对象) -续2



<http://www.fjscdsxxh.com/news/show-54-146.html>

3.1.1 GameObject类 (销毁游戏对象)

■ Destroy

- **Destroy()** 主要用于销毁游戏对象以及组件，但不会再引用那个被销毁的对象
- **格式**
 - **Destroy (GameObject)**
 - **Destroy (GameObject, time)**
- **说明**
 - **GameObject**是销毁的游戏对象，也可以是**Prefab**预制体
 - **time**是销毁游戏对象的指定时间

3.1.1 GameObject类（显示/隐藏游戏对象）

■SetActive显示/隐藏游戏对象

➤ 在Unity中，要激活游戏对象的方法就是使用**SetActive()**，就是说通过此方法让游戏对象显示或者隐藏

➤ 格式

- **GameObject.SetActive (value)**

➤ 说明

- **GameObject**是定义**GameObject**游戏对象的变量名
- **value**是让物体是否显示或隐藏，类型是**bool**

代码演示

“Project 37: ActiveSwitch.cs”

3.1.2 Transform类

■ 场景里的每个对象都含有**Transform**，用来存储并控制物体的位置、旋转和缩放

➤ **transform.position**: 指定物体在世界坐标下的位置

- `transform.position=new Vector3 (1,0,0)`

➤ **transform.Translate**: 指物体相对位移的单位

- `transform.Translate (0,2,0)`

➤ **transform.Rotate**: 指物体旋转

- `transform.Rotate (0,60,0)`

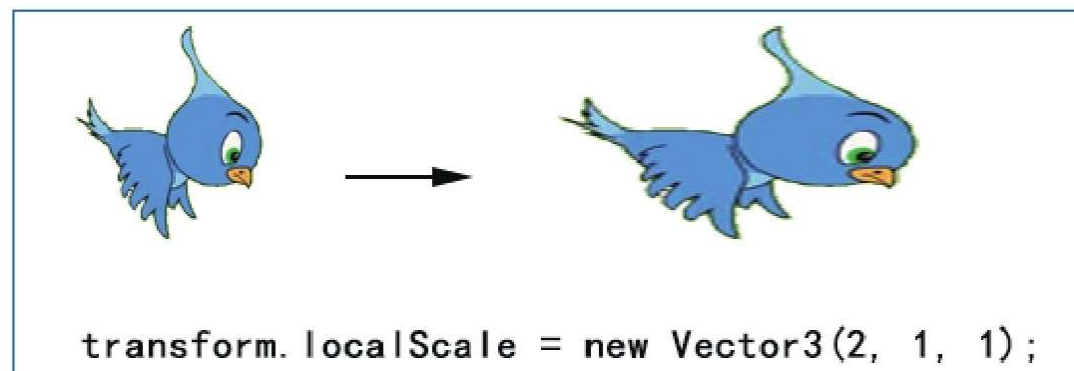
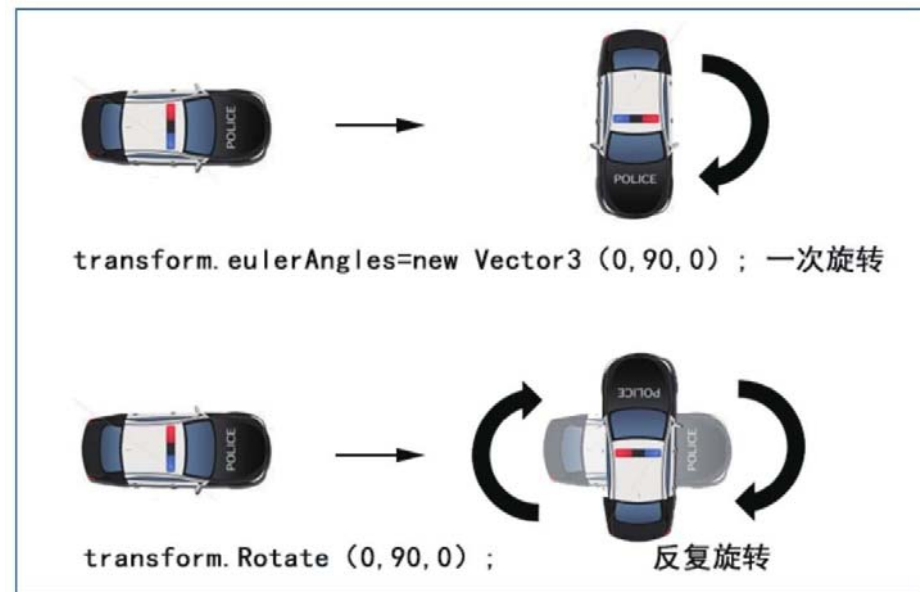
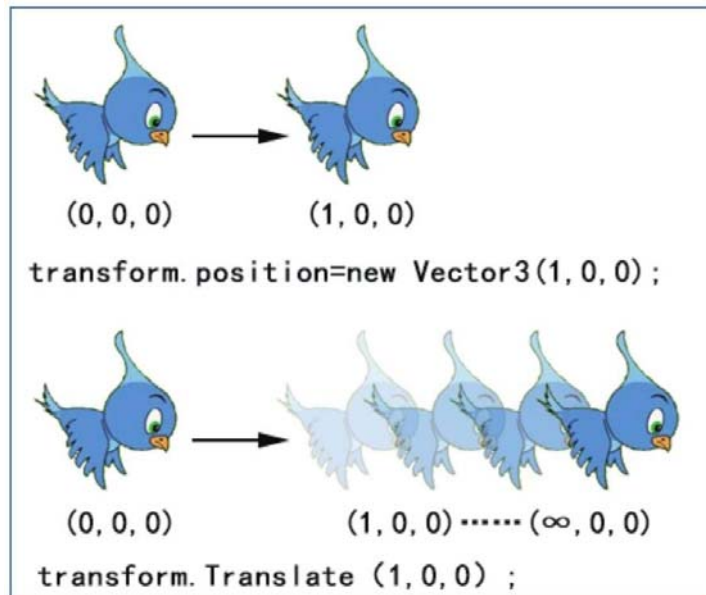
➤ **transform.eulerAngles**: 指物体的角度

- `transform.eulerAngles=new Vector3 (0,60,0)`

➤ **transform.localScale**: 指物体的缩放，注意的是缩放各个轴不能为0，否则会消失

- `transform.localScale= new Vector3 (2,3,5)`

3.1.2 Transform类-续



3.1.3 Time类

- Time类是在Unity中获取时间信息的接口类，可以用来计算时间的消耗，只有静态属性（单位：秒）

变量名	说明
time	游戏从开始到现在所经历的时间
deltaTime	从上一帧到当前帧消耗的时间
fixedTime	最近FixedUpdat的时间，从游戏开始计算
fixedDeltaTime	物理引擎和FixedUpdate的更新时间间隔
timeSinceecLevelLoad	从当前Scene开始到目前为止的时间
realtimeSinceStartup	从游戏启动到现在已经运行的时间
frameCount	已渲染的总帧数

代码演示

“Project 37:
TimeTest.cs”

$$\sum_{i=1}^{fps_rate}$$

$$\text{deltaTime}_i = 1s$$

$$\text{deltaTime} \times \text{fps_rate} = 1s!$$

3.1.4 Mathf类 (1/2)

■ **Mathf**是Unity提供的所有数学计算时需要用到的函数

➤ 常用的变量

变量名	说明	例子	结果
PI	圆周率 π	Mathf.PI;	3.141593
Deg2Rad	角度到弧度的转换系数	Mathf.Deg2Rad	0.01745329
Rad2Deg	弧度到角度的转换系数	Mathf.Rad2Deg	57.29578

3.1.4 Mathf类 (2/2)

➤ 常用的方法

变量名	说明	例子	结果
Abs	计算绝对值	Mathf.Abs(-6.5f)	6.5
Sqrt	计算平方根值	Mathf.Sqrt(1.69f)	1.3
Min	返回最小值	Mathf.Min(9,3)	3
Max	返回最大值	Mathf.Max(5,8)	8
Pow(a,n)	返回a的n次方	Mathf.Pow(3,5)	243
Log	计算对数	Mathf.Log(32,2)	5
Round	四舍五入取整	Mathf.Round(7.6f)	8
Clamp	将数值限制在min和max之间	Mathf.Clamp(10,1,3)	3
Sin	计算角度的正弦值	Mathf.Sin(2);	0.9092974
Cos	计算角度的余弦值	Mathf.Cos(2);	-0.4161468
Tan	计算角度的正切值	Mathf.Tan(2);	-2.18504

3.1.5 Random类

■ Random类是用于产生随机数

➤ 常用的成员变量和成员函数

变量名	说明
seed	随机生成器的种子
value	返回一个0~1之间的随机浮点数
rotation	返回一个随机旋转

函数名	说明
Range(min,max)	返回一个min~max之间的一个数

代码演示

“Project 37: SwingAndRandomTest.cs”

3.2.1 Input

■ Mouse

➤ **Input.mousePosition**

■ button

➤ **Input.GetMouseButton (0)**

■ touch

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

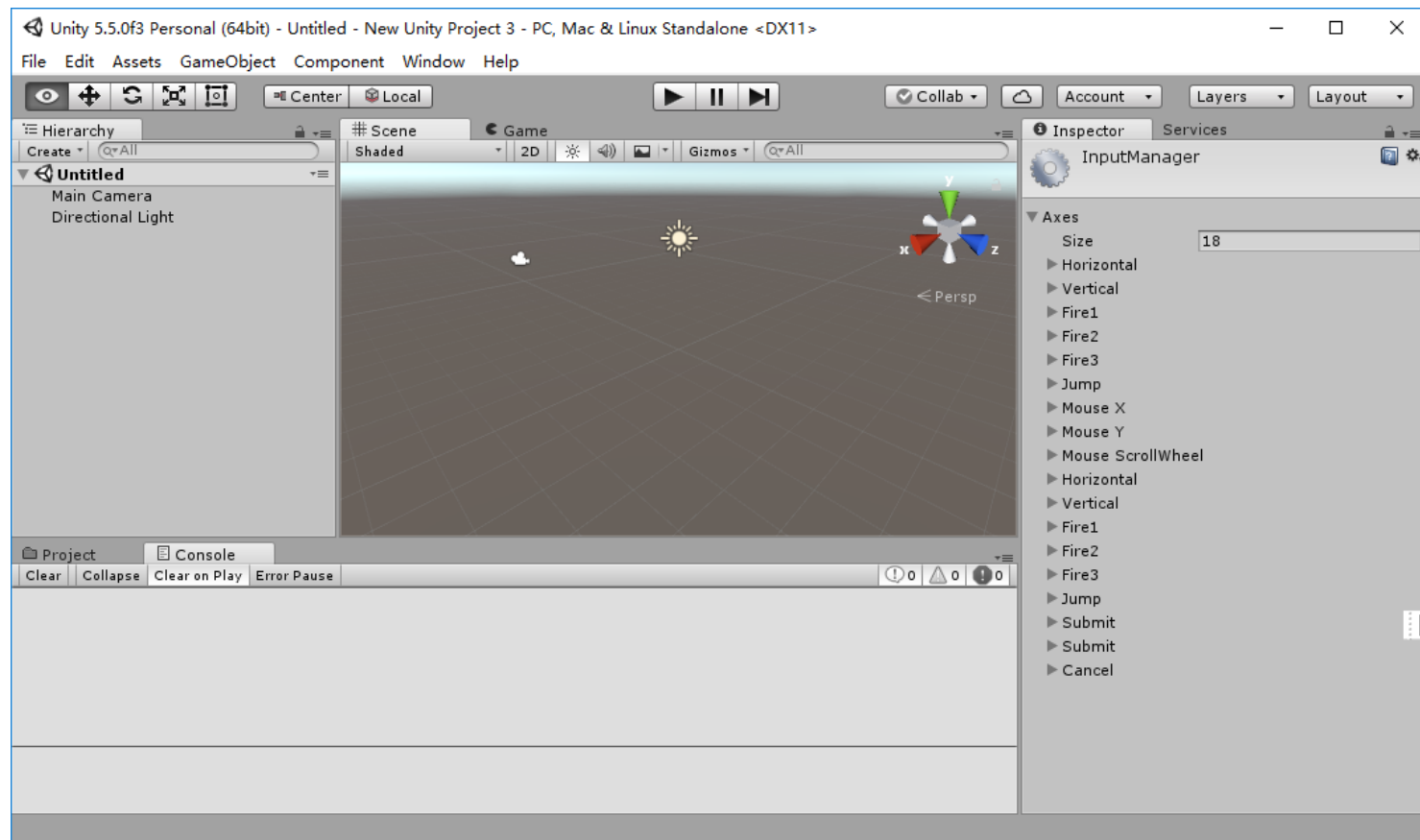
3.2.2 InputManager

■ Jump（测试运行游戏“LIGHT.rar”）

■ Fire1

■ ADWS

“Edit”→“Project Settings”→“Input”



代码演示

“Project 37: InputTest.cs”

3.2.3 键盘输入

■ Input.GetKey()

key本身的操作	Input Manager配置
Input.GetKey	Input.GetButton("Jump")
key = Input.GetKey(KeyCode.Space)	Input.GetButtonUp("Jump")
key = Input.GetKey("Space")	Input.GetButtonDown("Jump")

个性化、自定义快捷键

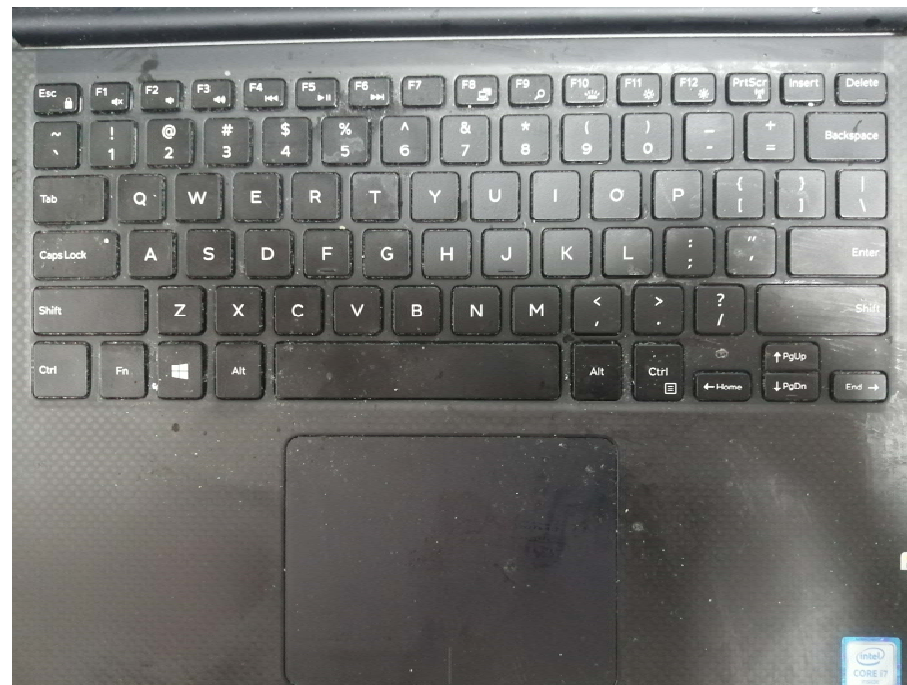
方向键对不同物体的
运动控制；
记录精确的键值

代码演示

“Project 37:

GetButtonTest.cs

MotionControl.cs”



作业5 Unity3D-脚本的响应事件与组件访问

❖ 编写Unity3D工程

- (1) 在C#脚本中使用Input类的touch方法（支持手机、平板触摸屏）编写响应函数对物体的运动控制；
 - ◆ 可参考链接：
<https://blog.csdn.net/u013108312/article/details/52575750>
 - ◆ 根据自己的设备类型将工程发布到Andriod或iOS平台，在手机或平板上测试
- (2) 在C#脚本中使用Input.GetKey()方法对从网站上下载的1段英文文本进行字母（不区分大小写）的频率统计
 - ◆ 列出下载链接和英文文本全文
 - ◆ Debug.Log()或print输出26个字母的出现次数和频率（百分比）

❖ 要求：

- 图文并茂记录整个实现过程，粘贴完整的代码。
- 通过网络教学综合平台（<http://jxpt.cuc.edu.cn>）提交作业，截止时间：**2020-05-04、24时前**。

参考文献

- ❖ Unity公司 主编 史明 刘杨 编著. Unity 5.X/2017标准教材. 北京: 人民邮电出版社, 2018.