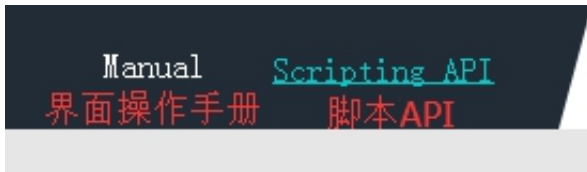


官方文档:英文版



官方文档:中文版

unity.cn

Input类

鼠标

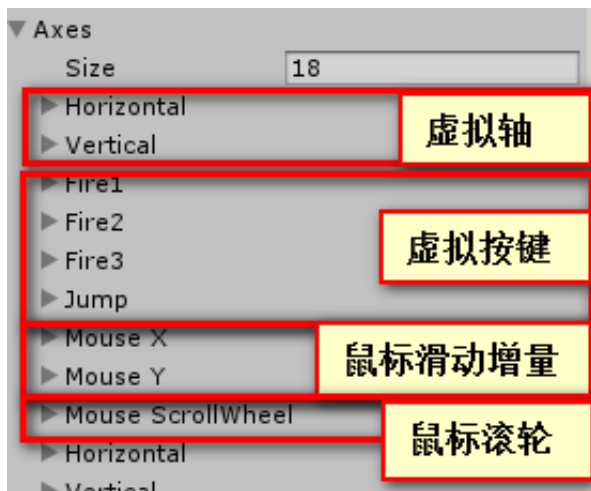
```
Input.GetMouseButtonDown(); 鼠标按下  参数:0左键 1右键 2中键  
Input.GetMouseButton(); 鼠标按住  
Input.GetMouseButtonUp(); 鼠标抬起  
Input.mousePosition 鼠标位置
```

按键

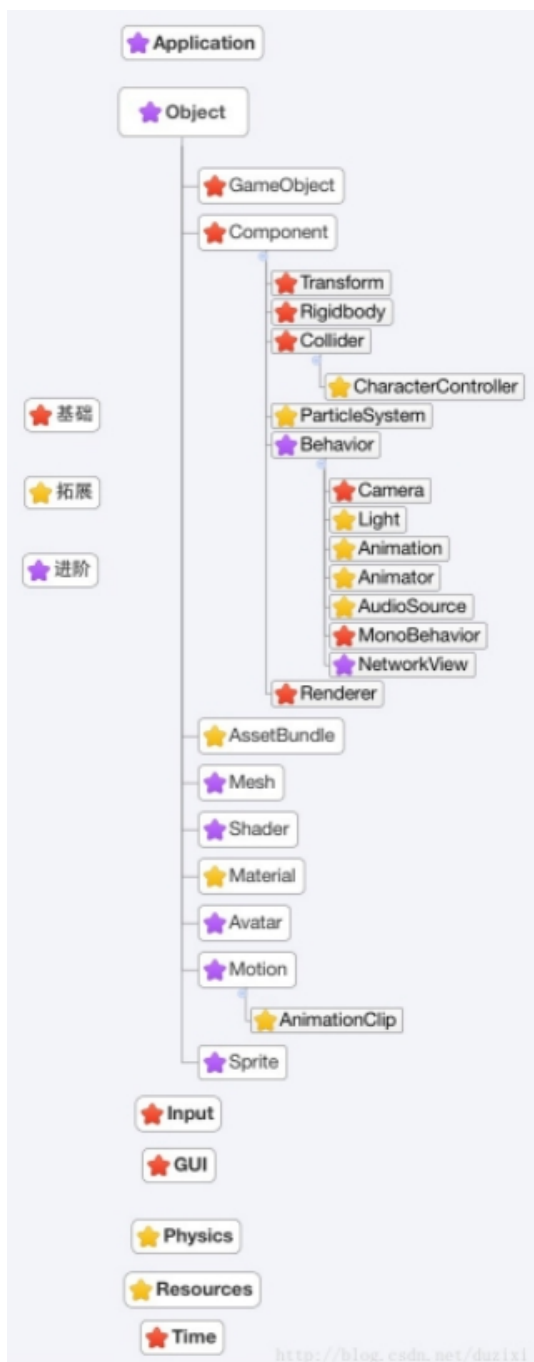
```
Input.GetKey(); 按键按住 参数是KeyCode枚举  
Input.GetKeyDown(); 按键按下  
Input.GetKeyUp(); 按键抬起
```

虚拟按键和虚拟轴

```
设置:Edit > ProjectSettings > Input  
Input.GetButton(); 获取虚拟按键  
Input.GetAxis(); 获取虚拟轴 (带缓动)  
Input.GetAxisRaw(); 获取虚拟轴 (不带缓动)
```



Unity继承关系图



GameObject (游戏物体)

游戏物体是存放组件的容器,不同的组件决定整个游戏物体的表现效果

静态方法

`GameObject.Instantiate();` 实例化 //此方法继承自`UnityEngine.Object`
`GameObject.Destroy();` 销毁////此方法继承自`UnityEngine.Object`
`GameObject.DontDestroyOnLoad();` 加载新场景时时不销毁
`GameObject.Find();` 查找游戏物体(根据名称) 无法查找被禁用的物体
`GameObject.FindGameObjectWithTag();` 标签查找游戏物体
`GameObject.FindGameObjectsWithTag();` 标签查找, 返回游戏物体数组
`GameObject.FindObjectOfType();` 通过类型查找组件
`GameObject.FindObjectsOfType();` 通过类型查找组件, 返回数组

方法

`GetComponent();` 获取组件
`AddComponent();` 添加组件
`SetActive();` 设置激活状态
`SendMessage();` 发送消息 不需要获取组件就可以调用方法,效率低,使用反射实现的
`SendMessageUpwards();` 向上(所有父物体)发送消息 (包括自己)
`BroadcastMessage();` 向下(所有子物体)发送消息,所有子物体(包括自己)

属性

`name;` 名称
`tag;` 标签
`transform;` tranform组件 特殊:不需要`GetComponent`就可以拿到
`activeSelf;` 激活状态(当前是否激活)
`activeHierarchy;`在层级面板中的激活状态

MonoBehaviour类

`print()`可以替代`Debug.Log()`

`Invodke()`可以延时调用方法

`MonoBehaviour`定义了Unity的脚本生命周期,

继承自`MonoBehaviour`的类不能写构造函数 不能通过`new`创建对象,只能通过

`AddComponent`添加的组件的方式创建对象

作业:

1. 方块初始为红色,点击方块,让方块的颜色 红>绿>蓝>黄>蓝>绿>红>

绿.....点一次变一个颜色

乒乓变化

2. 通过鼠标左键点击让游戏物体: 激活>禁用>激活.....