

# 18363857132

HOME

CONTACT

GALLERY

## Unity3d 基础概念

2016-10-19 10:23 by 刘杰0913, 48 阅读, 0 评论, 收藏, 编辑

坐标系：

世界坐标系（word）与本地坐标系（local）：所有无敌的世界坐标系都是相同的，不会改变：每个物体都有一个本地坐标系，当物体旋转时，本地坐标系也会跟着物体一起旋转。

例如：吧Cubede Rotation改为（45.0.0），这样世界坐标和本地坐标就不一样了。

常用的必然事件：

有一些特定的函数，这些函数会在满足某些条件时有Unity自动调用，他们被称为必然事件（Certain Events）

Awake():唤醒脚本，此方法为系统执行的第一个方法，用于脚本的初始化，在脚本的生命周期中只执行一次。

Update():正常更新，用于更新逻辑。每创建一个脚本，脚本中都会默认添加这个方法。此方法每帧都会由系统自动调用一次。

Start():此方法在Awake()方法之后，Update()方法之前执行，并且只执行一次。

LateUpdate():此方法在Update()方法执行后调用，同样每一帧都调用。

FixedUpdate():固定更新，更新频率为0.02s

OnGUI() OnGUI()在每一帧都会由系统自动调用一次，渲染和处理GUI事件时调用。

输出：

```
void star(){
    print ("hello");
    Debug.Log("hello");
    Denbug.LogWarning("hello");
    Debug.logError("hello");
}
```

MonoBehanviour类：

所有的脚本类均需要从它之间或者间地继承，脚本必然事件就是从它继承而来。除了必然事件它还定义了各种特定事件的响应函数。如下图：

监视变量：

```
public int id ; //在监视面板中看得到。
private string name; //在监视面板中看不到。
protected string password ; //在 监视面板中看不到。
int gae; //在监视面板中看不到。
```

Transform:

游戏对象的Transform:

gameObject 的 ransform属性可以进行位置.旋转。大小的设置：

位置：position

旋转：rotate

大小：localScale

如果要操作脚本当前使用的ganmeObject,则可以省略不写要操作的gameObject。

### About

昵称：[刘杰0913](#)  
园龄：[4个月](#)  
粉丝：[0](#)  
关注：[0](#)  
[+加关注](#)

SEARCH

### 最新评论

日历							随笔档案						
< 2017年3月 >							2016年10月(5)						
日	一	二	三	四	五	六							
26	27	28	1	2	3	4							
5	6	7	8	9	10	11							
12	13	14	15	16	17	18							
19	20	21	22	23	24	25							
26	27	28	29	30	31	1							
2	3	4	5	6	7	8							
随笔分类													
推荐排行榜													
阅读排行榜													
1. U3d中C#脚本常用方法(151)													
2. Unity3d 基础概念(49)													
3. U3d知识点的总汇(11)													
4. C#中的委托和事件(6)													

Vector3向量介绍：

Vector3是Unity中定义的一个含有x,y,z三个字段的类，可以表示位置点，也可以表示一个向量。Vector3乘以数值表示对向量的三个值乘以相应的值，v1-v2则表示从v2指向v1的向量。

transform.Transform(Vector3)让对象沿着vector3向量的方向移动。

Vector3类中内置了 up, down, left, right, forward, back, zero几个简化的静态成员。

```
transform.Translate (Vector3.forward);
```

```
transform.Translate (new Vector3(10,0,0));
```

```
transform.Rotate (new Vector3(0,10,0));
```

```
transform.localScale += new Vector3 (0.1F, 0, 0);
```

RotateAround(v1.v2.angle).元素围绕着世界坐标的V1点采用V2向量选装angle角度。

```
transform.RotateAround(transform.position,Vector3.up,1)
```

键盘判断：

GetKey 当通过名称指定的按键被用户按住时返回true

GetKeyDown 当用户按下指定名称的按键时的那一帧返回true。

GetKeyUp 在用户释放给定名字的按键的那一帧返回true。

GetAxis(“Horizontal”)和GetAxis(“Verical”)

用方向键或WASD键来模拟-1到1的平滑输入

```
if (Input.GetKey(“up”)) {
```

```
print("up arrow key is held down");
```

```
}
```

```
if (Input.GetKey(“down”)) {
```

```
print("down arrow key is held down");
```

```
}
```

```
If(Input.GetKeyDown(KeyCode.Escape)){
```

```
print(“按下Esc键”);
```

```
}
```

```
If(Input.GetKeyUp(KeyCode.Escape)){
```

```
print(“松开Esc键”);
```

```
}
```

```
If(Input.GetKeyDown(KeyCode.A)){
```

```
print(“按下A键”);
```

```
}
```

```
If(Input.GetKeyUp(KeyCode.D)){
```

```
print(“松开D键”);
```

```
}
```

```
If(Input.GetAxis(“Horizontal”)){
```

```
print(“按下水平键”);
```

```
}
```

```
If(Input.GetAxis(“Verical”)){
```

```
print(“按下垂直键”);
```

```
}
```

鼠标判断：

GetButton 根据按钮名称返回true当对应的虚拟按钮被按住时。

GetButtonDown 在给定名称的虚拟按钮被按下那一帧返回true。

GetButtonUp 在用户释放指定名称的虚拟按钮时返回true。

按下鼠标左键

```
if(Input.GetButton("Fire1")){
    print("按下鼠标左键");
}
if (Input.GetMouseButton(0)) {
    Debug.Log("按下鼠标左键");
}
if (Input.GetMouseButton(1)) {
    Debug.Log("按下鼠标右键");
}
if (Input.GetMouseButton(2)) {
    Debug.Log("按下鼠标中键");
}
```

给物体加入刚体：

```
ransform.rigidbody.AddForce (0,0,1000);
```



**刘杰0913**  
关注 - 0  
粉丝 - 0

0

0

[+加关注](#)

« 上一篇：[U3d知识点的总汇](#)  
» 下一篇：[U3d中C#脚本常用方法](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库  
【推荐】Google+ GitHub联手打造前端工程师课程  
【推荐】票选最美云上大数据暨大数据技术峰会



最新IT新闻：

- 苹果以32GB iPhone 6作为入门机是否冒险？
  - 美团点评的酒旅业务想叫板携程 还得跨过这些坎
  - 揭秘顺丰背后资本推手 元禾控股：官办PE的市场化探索
  - 没有人知道，我们是否已经步入一个新时代
  - 朱啸虎：创业不需要追风口，但一定要追求时机
- » [更多新闻...](#)



**最新知识库文章:**

- 垃圾回收原来是这么回事
- 「代码家」的学习过程和学习经验分享
- 写给未来的程序媛
- 高质量的工程代码为什么难写
- 循序渐进地代码重构
- » 更多知识库文章...