

个人资料



BigoSprite

+ 关注

✉ 发私信

恒

访问：34770次

积分：1632

等级：BLOG > 4

排名：千里之外

原创：129篇

转载：5篇

译文：3篇

评论：7条

个人信息

邮箱：bigosprite@163.com

新浪微博：BIGOSPRITE

只有高手才能活下来.

文章分类

C++初学笔记

(21)

C++

(6)

深度探索C++对象模型

(9)

每天学点C++/C#

(7)

Boost

(3)

LeetCode算法(C++)

(11)

STL

(2)

基础知识扫盲

(4)

Unity3D初学笔记

(7)

数据结构和算法

(6)

边学边码的数据结构和算法

(1)

TCP/IP协议

(0)

Cocos2d-x 3.x开发之旅的笔记

(4)

桂老师C++/JAVA/设计模式的课堂记

(4)

Javascript学习笔记

(2)

设计模式

(5)

ASP.NET

(4)

C#

(3)

MSMQ

(1)



💡【评论送书】每周荐书：我的世界、架构师、OpenStack Python 创意编程活动 CSDN日报20170515 ——《聊聊我对 WannaCry 产生的感慨》

原 Unity 3D：访问另一个C#类中的属性或方法

2016-06-04 18:33

👁 879人阅读

💬 评论(0)

★ 收藏

⚡ 举报

≡ 分类：设计模式 (4) ▾

❗ 版权声明：本文为博主原创文章，未经博主允许不得转载。

在开发Unity 3D 游戏的过程中，经常遇到在某个C#类中访问另一个C#类中的方法或属性；比如：在Eenmy.cs中访问PlayAttack.cs脚本中的TakeDamage(int damage)方法(该方法控制主角受伤的情况)，以实现当敌人攻击主角时调用TakeDamage方法对主角进行受伤的处理。那么，如何实现呢？有下面常用的三种方法：

1. 设置PlayerAttack类为单例模式

```
[csharp]
01. // PlayerAttack脚本
02. public class PlayerAttack : MonoBehaviour {
03.     public static PlayerAttack _instance; // static关键字。 单例模式
04.
05.     void Awake(){
06.         _instance = this;// 确保单例模式在使用前已被初始化
07.     }
08.
09.     public void TakeDamage(int damage){
10.         // do something
11.     }
12. }
```

```
[csharp]
01. // Enemy脚本
02. public class Enemy : MonoBehaviour {
03.     //
04.     void Attack(){
05.         // 当敌人攻击主角时
06.         PlayerAttack._instance.TakeDamage(20);
07.     }
08. }
```

2. 把要访问的方法设置为静态方法

```
[csharp]
01. // PlayerAttack脚本
02. public class PlayerAttack : MonoBehaviour {
03.
04.     public static void TakeDamage(int damage){
05.         // do something
06.     }
07. }
```

```
[csharp]
01. // Enemy脚本
02. public class Enemy : MonoBehaviour {
03.     //
04.     void Attack(){
05.         // 当敌人攻击主角时
06.         PlayerAttack.TakeDamage(20);
07.     }
08. }
```

3. 通过SendMessage方法传递参数（SendMessage用法）

```
[csharp]
01. // PlayerAttack脚本
02. public class PlayerAttack : MonoBehaviour {
03.
04.     public void TakeDamage(int damage){
05.         // do something
06.     }
07. }
```

```
[csharp]
01. // Enemy脚本
02. public class Enemy : MonoBehaviour {
03.     //
04.     void Attack(){
05.         // 当敌人攻击主角时
06.         // 首先需要获取挂PlayerAttack的主角
07.         GameObject player = GameObject.FindGameObjectWithTag("PlayerBoy");// Unity中主角模型的标签为PlayerBoy
08.         player.SendMessage("TakeDamage", 20);
09.     }
10. }
```

关闭

