

Unity 游戏框架搭建

2017（三） MonoBehaviour 单例的模板

上一篇文章讲述了如何设计 C# 单例的模板。也随之抛出了问题:

- 如何设计接收 MonoBehaviour 生命周期的单例的模板?

如何设计?

先分析下需求:

- 约束脚本实例对象的个数。
- 约束GameObject的个数。
- 接收MonoBehaviour生命周期。
- 销毁单例和对应的GameObject。

首先, 第一点,约束脚本实例对象的个数,这个在上一篇中已经实现了。

但是第二点,约束 GameObject 的个数,这个需求,还没有思路,只好在游戏运行时判断有多少个 GameObject 已经挂上了该脚本,然后如果个数大于1抛出错误即可。

第三点,通过继承MonoBehaviour实现,只要覆写相应的回调方法即可。 第四点,在脚本销毁时,把静态实例置空。完整的代码就如下所示:

```
using UnityEngine;
```

```
/// <summary>
```

```

/// 需要使用Unity生命周期的单例模式
/// </summary>
namespace QFramework
{
    public abstract class QMonoSingleton<T> : MonoBehaviour where
T : QMonoSingleton<T>
    {
        protected static T instance = null;

        public static T Instance()
        {
            if (instance == null)
            {
                instance = FindObjectOfType<T>();

                if (FindObjectsOfType<T>().Length > 1)
                {
                    QPrint.FrameworkError ("More than 1!");
                    return instance;
                }

                if (instance == null)
                {
                    string instanceName = typeof(T).Name;
                    QPrint.FrameworkLog ("Instance Name: " +
instanceName);

                    GameObject instanceGO =
GameObject.Find(instanceName);

                    if (instanceGO == null)
                        instanceGO = new
GameObject(instanceName);

                    instance = instanceGO.AddComponent<T>();
                    DontDestroyOnLoad(instanceGO); //保证实例不会
被释放

                    QPrint.FrameworkLog ("Add New Singleton " +
instance.name + " in Game!");
                }
                else
                {
                    QPrint.FrameworkLog ("Already exist: " +
instance.name);

```

```

        }
    }

    return instance;
}

protected virtual void OnDestroy()
{
    instance = null;
}
}
}

```

转载请注明地址：凉鞋的笔记：liangxiegame.com

更多内容

- QFramework 地址：<https://github.com/liangxiegame/QFramework>
- QQ 交流群：[623597263](https://jq.qq.com/?_w=1027&u=623597263)
- **Unity 进阶小班：**
 - 主要训练内容：
 - 框架搭建训练（第一年）
 - 跟着案例学 Shader（第一年）
 - 副业的孵化（第二年、第三年）
 - 权益、授课形式等具体详情请查看[《小班产品手册》](https://liangxiegame.com/master/intro)：<https://liangxiegame.com/master/intro>
- 关注公众号：liangxiegame 获取第一时间更新通知及更多的免费内容。

