

16 节点之外的对象——RefCounted

一、RefCounted 概述

①RefCounted 的地位

- RefCounted 继承自 Object，它以及它的子类同样具有 Object 的所有功能。
- 它拥有众多的子类。RefCounted 内置了一个独特的引用计数器，要了解掌握这些子类，就必须先了解掌握 RefCounted 的引用计数器机制。

②RefCounted 对象的生成

- 类名.new() (类名由 class_name 进行自定义)
- load(“路径”).new() (也可以 preload(“路径”).new())

③RefCounted 的特征及作用

- 特征 1：内置引用计数器，当引用计数器归 0 时，此对象自动被程序删除。
- 特征 2：在 GDS 中不包含任何内置属性，仅有四个内置函数，也不必像节点一样要频繁的受场景树的控制，参与服务器的计算工作，相比于节点更加轻量。
- 作用：RefCounted 不但比 Node 占用更少的内存节约更多的电脑的算力、还可以进行内存的自动管理。非常适合用于制作游戏中数目庞大的数据对象。如仙侠游戏中的成百上千的随机 NPC 数据对象。

二、RefCounted 的重要子类

最重要：

- Resource——游戏文件与游戏程序的中转站，所有的图片、音频、视频文件都必须先转为 Resource，然后才可以被节点等对象使用。

了解即可，没有编程经验的学习者可以先不用了解：

- Astar2D/Astar3D 等可以用于构建游戏中的寻路数据。
- Thread/Mutex/SemaPhore 可以用于构建多线程。
- FileAccess/DirectoryAccess 可以用于加载读取计算机内的文件。
- 还可以使用某些子类进行 UDP、TCP 或 Websocket 的网络通信。
- StreamPeer 可以以字节单位对网络传输中收发的数据进行处理。
- 此外，它的子类还涉及了正则、JSON、游戏物理、自定义引擎编辑器等领域。