如何理解物理时间周期库 Cadencer

裸代码解决时间周期问题是个复杂的工作,例如,我们必须先记录上一次时间,再用本次周期的时间减掉上一次,这样才能得到本次周期.另一方面,对物理时间的转换会使用浮点和整数,这样非统一化编写物理时间处理流程会很容易发生低级错误并且十分烧脑,长出白发和秃头.

Cadencer 库被广泛运用于全部基础库,并且统一使用浮点来处理物理时间周期,并且度量统一化:0.001=1ms,1.0=1 秒

例如在 Z.Net 中,DelayFreeObj(2.0, MyObj),这样的 API 表示在 2 秒以后,释放 MyObj,而 2 秒的时间计数则是交给 Cadencer 在干,而触发 DeltaFreeObj 则是交给 Z.Notify 库来干.使用 DelayFreeObj 来延迟释放,许多 MM 管理器在多线程释放对象的虚拟机制表时会时而抛出异常,在许多流程都是直接 DelayFreeObj 把 Obj 扔给主线程去延迟释放,从而解决晦涩的 Obj 地址表在多线程释放的问题.尤其某些继承于 Interface 的 Obj,Destroy 谜之异常的几率会到 1/10W,在 HPC 服务器程序模型中,多线程和并行是被大量使用的机制.DelayFreeObj 虽然暴力,但作用很明显.

在 Z.Net 中所有的主循环均含有对 Cadencer 库的调用,其作用就是给每次主循环的 Progress 传递一个周期时间,这样可以触发一些需要延迟处理的事件.

在 Z.DrawEngine 中,fps 计数器,运动库的周期时间,同样也是使用 Cadencer.

在 FFMPEG 中,视频的图像换帧周期,同样也是使用 Cadencer 来进行换算,而音频则让 Bass 内部自己计算音频数据帧的时间周期,应为二者都使用物理所以才能做到画音同步.

在 Z.AI 中,记录 GPU 在 DNN-Thread 中每秒的 PSP,Max-PSP,估算性能,算力自洽调节都是按物理时间来干,这也使用到了 Cadencer.

2024-1

By.qq600585