

002--视觉班第2次课程[OpenGL补充]



一. 视觉班课程安排:

- 课程日期: 2020 年 7 月 3 日 周五 第 2 次课程 (共 21 次课程)
- 授课老师: CC 老师 (QQ: 1323177506)
- 研发老师: CC 老师
- 班主任老师:
 - 大大老师 (QQ: 188706023)
 - 朵朵老师 (QQ: 1550934962)
 - 婷婷老师 (QQ: 3470520842)
- 课程时长: 2小时
- 课程时间安排:
 - 上课: 20:00 – 21:00
 - 休息: 21:00 – 21:10
 - 上课: 21:10 – 22:00
- 课程主题: iOS场景下的渲染[视觉班补充内容]
- 课程内容:
 - 案例: 002--使用OpenGL 固定管线下的着色器渲染一个正方形,并完成使用键盘对其移动控制
 - CPU 与 GPU 芯片区别;
 - 计算机渲染原理
 - 屏幕成像过程与卡顿情况;
 - iOS 下的渲染框架
 - CoreAnimation 在渲染角色00

- 离屏渲染从视觉课程的角度解读;
- 课程作业:
 - 在博客中,将案例002的执行过程,使用简图绘制出来; 理解代码执行过程;
 - 将此次补充课程内容中, CoreAnimation 在渲染充当的角色以及从视觉课程的角度来理解离屏渲染 结合自己的理解总结在博客(面试高频题目)

二. 课程内容安排

2.1 上节课复盘

2.1.1 优秀作业点评;

2.1.2 上节课知识点快速回顾;

2.2 课程笔记

2.2.1 CPU 与 GPU 角色;

CPU -> 运算核心/控制

GPU->绘图运算的微处理器

CPU-> 逻辑复杂,数据

GPU-> GLSL ->

CPU -> 依赖性非常高. 并发,时间片切换

GPU -> 计算单元->高并发 ->依赖性非常低

媳妇->怀孩子(10个月) -> 拆解10个人->1个月 CPU

GPU-> 找女朋友(10个人)~~GPU

2.2.2 计算机显示方式演变; 最初形态->最终形态+

随机扫描显示;

光栅扫描显示;

图像->像素阵列组成, 显示一个图像时间,显示整个光栅所需的时,和图像复杂度无关

显示过程->不断刷新. 感受不到~

1秒多少帧~连贯(16帧)

人眼视觉暂留特性解读:

视觉暂留

[编辑](#)[讨论](#)[上传视频](#)[+](#) [★ 收藏](#) [👍 1559](#) [🗨️ 128](#)

视觉暂留现象即视觉暂停现象 (Persistence of vision, Visual staying phenomenon, duration of vision) 又称“余晖效应”，1824年由英国伦敦大学教授皮特·马克·罗葛特在他的研究报告《移动物体的视觉暂留现象》中最先提出。

人眼在观察景物时，光信号传入大脑神经，需经过一段短暂的时间，光的作用结束后，视觉形象并不立即消失，这种残留的视觉称“后像”，视觉的这一现象则被称为“视觉暂留”。

中文名	视觉暂留现象	别 称	余晖效应
外文名	Persistence of vision, Visual staying phenomenon, duration of vision	出 自	《移动物体的视觉暂留现象》
		发现人	伦敦大学教授皮特·马克·罗葛特
		时 间	1824年

显示器 显示内容→帧缓存区

视频控制器：控制刷新部件 帧缓存区与显示器的对应关系~ 进行显示

帧缓存区：颜色值(黑白~) 帧缓存, 显存~

内存：连续的计算机存储器，主要存储刷新图像信息；

位图：60 * 60 = 3600 * 4(RGBA) = 14400

GPU进行渲染→帧缓存区里 →视频控制器→读取帧缓存区信息(位图) → 数模转化(数字信号处→模拟型号) →(逐行扫描)显示；

完美情况：每扫描一张图→不断显示/不断刷新

//60Fps

帧缓存区(旧的数据) → 显示屏幕 → 帧缓存区(下一帧数据) → 显示屏幕
及时更新

苹果：垂直同步Vsync + 双缓存区 DoubleBuffering

iOS一直使用策略 垂直同步Vsync + 双缓存区 DoubleBuffering

垂直同步Vsync：帧缓存区加锁 防止出现撕裂情况

撕裂：CPU/GPU 计算时间 等等它；双缓存区（撕裂的方式）

掉帧：启用垂直同步Vsync + 双缓存区 DoubleBuffering 解决(屏幕撕裂问题) →新的问题

接收Vsync ,cpu/gpu图片数据(速度问题) → 拿不到FrameBuffer → 掉帧(重复渲染同一帧数据)

三缓存区(CPU /GPU 闲置时间, A显示屏幕, B也渲染好.)三缓存也有可能出现掉帧->比双缓存区

屏幕卡顿原因:

1. CPU/GPU 渲染流水线耗时过长->掉帧
2. 垂直同步Vsync + 双缓存区 DoubleBuffering 以掉帧作为代价=>屏幕撕裂
3. 三缓存区: 合理使用CPU/GPU 减少掉帧次数;

+ ★ 收藏 | 7569 | 264

内存

编辑

本词条由“科普中国”科学百科词条编写与应用工作项目 审核。

内存是计算机中重要的部件之一，它是外存与CPU进行沟通的桥梁。计算机中所有程序的运行都是在内存中进行的，因此内存的性能对计算机的影响非常大。内存(Memory)也被称为**内存存储器**和**主存储器**，其作用是用于暂时存放CPU中的运算数据，以及与**硬盘**等**外部存储器**交换的数据。只要计算机在运行中，操作系统就会把需要运算的数据从内存调到CPU中进行运算，当运算完成后CPU再将结果传送出来，内存的运行也决定了计算机的稳定运行。**内存条**是由**内存芯片**、电路板、**金手指**等部分组成的。

[1]

★ 收藏 | 1346 | 49

显卡内存

锁定

同义词 显存一般指显卡内存

显存，也被叫做帧缓存，它的作用是用来存储显卡**芯片**处理过或者即将提取的**渲染**数据。如同计算机的内存一样，显存是用来**存储**要处理的图形信息的部件。

作用

如同**计算机**的内存一样，显存是用来存储要处理的**图形信息**的部件。我们在显示屏上看到的画面是由一个个的像素点构成的，而每个像素点都以4至32甚至64位的**数据**来控制它的亮度和色彩，这些数据必须通过显存来保存，再交由**显示芯片**和CPU**调配**，最后把运算结果转化为**图形输出**到**显示器**上。显存和**主板**内存一样，执行存贮的功能，但它存贮的对像是显卡输出到**显示器**上的每个像素的**信息**。显存是显卡非常重要的组成部分，**显示芯片**处理完**数据**后会将数据保存到显存中，然后由RAMDAC（数模转换器）从显存中读取数据并将**数字信号**转换为**模拟信号**，最后由屏幕显示出来。在高级的**图形加速卡**中，显存不仅用来存储图形**数据**，而且还被**显示芯片**用来进行3D函数运算。在nVIDIA等高级**显示芯片**中，已发展出和CPU平行的“GPU”（**图形处理单元**）。“T&L”（变形和照明）等高密度运算由GPU在显卡上完成，由此更加重了对显存的依赖。由于显存在显卡上所起的作用，显然显存的速度和带宽直接影响到显卡的整体速度。显存作为**存储**器也和**主板**内存一样经历了多个发展阶段，甚至可以说显存的发展比主板内存更为活跃，并有着更多的品种和类型。被广泛使用的**显存类型**是SDRAM和SGRAM，性能更加优异的DDR内存首先被应用到显卡上，促进了显卡整体性能的提高。DDR以在显卡上的成功为先导，全面发展到了**主板**系统，一个DDR“独领风骚三两年”的时代即将呈现在世人面前。



光栅显示系统组成的演变:

帧缓存区的大小计算:

光栅显示系统结构的演变过程:

屏幕成像与卡顿的问题:

屏幕撕裂的原因 ScreenTearing:

垂直同步信号 Vsync + 双缓存区 Double Buffering

UIView

绘制和动画;

布局和子view的管理

点击事件处理'

CALayer 渲染/动画

手机APP/ MAC -> 布局是一样? 2 APPKit / UIKit

layer ->迎合任何布局

1. 职责分离

2. 两个系统交换规则是不一样 ,APPKit / UIKit

HandleEvents : 事件处理

Commit Transaction: 图片

Render Server >解码(CPU)

-> CoreAnimation -> 提交OpenGL -> GPU ->渲染流程(顶点数据->顶点着色器->片元着色器->runloop ->显示)