

001--视觉班第1次课程[OpenGL专题]



一. 视觉班课程安排:

- 课程日期: 2020 年 7 月 1 日 周三 第 1 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
- 课程内容:
 - OpenGL Mac 端搭建固定着色器环境需要注意问题
 - 了解图形API主要解决什么问题?
 - 必备的OpenGL 开发过程的专业名词解析
 - 了解OpenGL 下的坐标系
 - 理解图片/图形从文件渲染到屏幕的过程解析
 - 案例: 001--使用OpenGL 固定管线下的着色器渲染一个三角形;
 - 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
- 课程作业:

- 将OpenGL 常见的专业名词,用自己的理解记录在自己的博客上;
- 完成案例001与002 的理解(绘制出这2个案例的执行流程图)

二. 课程内容安排

2.1 课程笔记(上课ing)

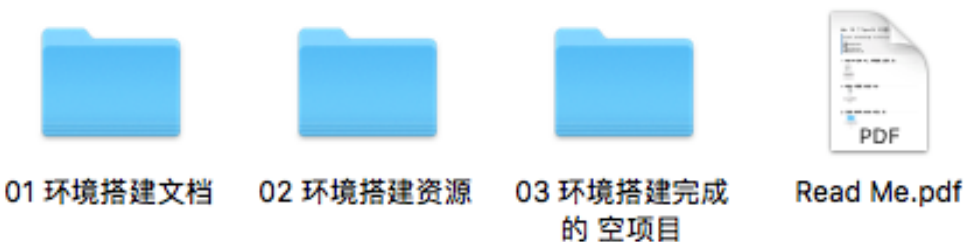
2.1.1 关于OpenGL 基础课程学习要求

不需要对OpenGL 固定着色器下的方法,死记硬背. 在OpenGL ES的背景下会有新的方法出现~
主要理解渲染, 绘制手段,图元,以及纹理坐标等;

为何要先讲固定着色器. 因为在这个基础上,同学可以暂时不用着眼与GLSL 语言以及相关的可编程下渲染着色器的问题; 先把基础掌握之后.再进行学习OpenGL ES 就会更加简单;

学习过程, 前面10节,会有比较重的畏难心理.比较正常.但是得坚持听. "不断重复,坚持"

2.1.2 关于环境搭建问题(资源)

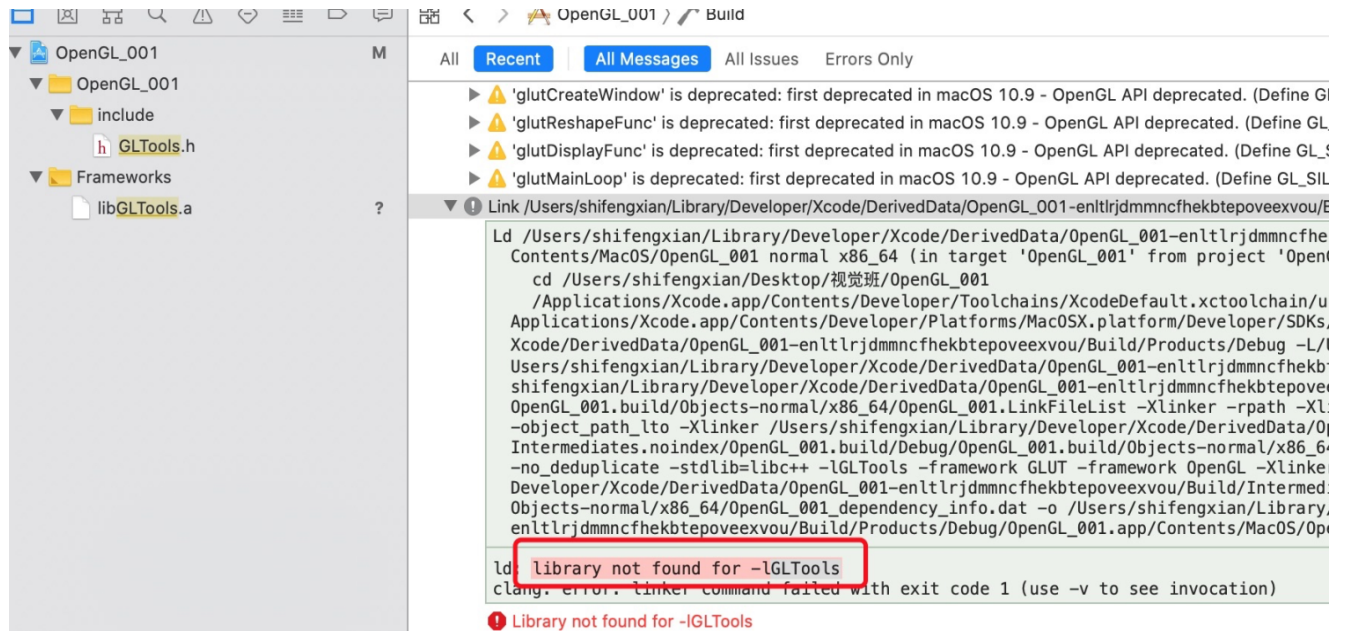


✨ 参考优秀学员博客:

- <https://www.jianshu.com/p/8a01ed478611>
- <https://www.jianshu.com/p/e4be61ba5fd5>
- <https://www.jianshu.com/p/43ae1587f45b>

✨ 预习课程(1-5节) 优秀学员博客: (非常优秀~ 加入了自己的理解. 推荐阅读)

<https://www.jianshu.com/nb/43003165> (成都-收纳箱)



将 libGLTools.a 不用拖到Framworks下面，拖到工程就可以了~

上海 - 养生失败猿

编译，将文件<>系统引入，改为“”普通引入

command+ ' 跳下一个issue

ctrl+option+command+F 全修复

很快就修完了 🤔

上海 - 养生失败猿

AppDelegate.h 、 AppDelegate.m 、 main.m 、 ViewController.h 、
ViewController.m ; 创建 main.cpp文件

建command line tools就行，然后把main.m改成main.cpp就好，不需要删文件了 (强行偷懒)

OpenGL/OpenGL ES

OpenGL ->PC 图形图像渲染处理(MAC,windows)

OpenGL ES -> 移动端(iOS,安卓)

OpenGL / OpenCV

- a. 渲染(显示->位图)
- b. 识别(人脸识别/身份识别/物体->OpenCV face++) -> 与人工智能结合;

OpenGL ES/Metal

- a. 2018 渲染部分-> CoreAnimation ->基于OpenGL ES 封装上层框架; 2018 -> 底层渲染迁移 Metal
- b. OpenGL ES 第三方~ 契合你项目定制化需求,迭代
- c. 很多项目->OpenGL ES (跨平台-> 兼容性 GLSL ->iOS/安卓)
- d. 替代原因: 1. 不可能一直把非常重要的渲染核心框架使用第三方; 2. 调用GPU来进行运行(GPU 并发运算,编码解码,识别,大量并发运算. 2018->入口,让你调用GPU来自定义事情. Metal 入口.借助 Metal 调度GPU来给你做运算.)

图形API 解决什么问题: -->渲染问题

- a. 系统针对按钮.图片,视图,图层渲染问题
- b. 游戏引擎->人物/场景渲染(图形API)
- c. 视频播放框架->ijkplayer,kxmovie 视频解码->渲染(图形API)
- d. 核心动画->动画操作(旋转,缩放,移动, 图层特效)
 - i. 视觉班 -> 80% (直播上课率+作业80%) 兑换一门视频课程免费学习(核心动画/音视频短训班)
- e. 视频/图片->特效(图形API)
- f. 离屏渲染 一知半解 (后续课程增加离屏渲染内容)

招聘-> 音视频项目(OpenGL ES 加分项)

上下文: 深度测试(功能) 记录

渲染: 图片/按钮/视频->显示绘制到屏幕过程 解码图片->渲染

顶点数组: 顶点数据->内存中

顶点缓存区: 顶点数据-> GPU 显存

位图: $120 * 120 = 14400$ -> RGBA -> $14400 * 4 =$ 位图 (纹理)

映射: 对应关系

管线: 流水线;

//工厂: 肥皂 -> 肥皂水->模具(10)->肥皂块

固定管线: 对应非常多模具 (固定着色器) ->结果 (OpenGL -> 固定)

可编程管线: 肥皂 -> 肥皂水->自定义编程模具(GLSL语法来驱使GPU)->肥皂块 (OpenGL ES GLSL)

//10分钟

着色器: 函数/方法(代码段) -> CPU来使用; shader(代码段)->GPU

固定着色器(存储着色器) 苹果提供API(代码段)->调用(参数) OpenGL 来提供 ;

自定义着色器: 进行自定义(自己基于GLSL语法来进行编写代码段)

顶点着色器: 用来处理顶点相关代码 1.确定位置 2.缩放/平移/旋转位置换算 3.手机端显示3D 手机屏幕实际2D, 3D图形数据->2D(投影换算) ->OpenGL ES

片元着色器: 片元(像素点)着色器. 处理一个一个像素点; 120 * 120 像素点. 14400次 GPU 并行运算;

拓展一下: 图片进行饱和度调整->片元着色器进行一个个像素点修改~ --> OpenGL ES

GLSL: 自定义着色器-> 代码(语言) OpenGL 标准调用GPU 来做计算;

光栅化: 2个过程(不可以编程) 第一个过程: 确定图形在像素范围 第二个过程: 颜色附着上去

纹理: 压缩->位图 OpenGL .tga 纹理文件 OpenGL ES 移动端显示一张 png->位图

混合: Layer 2个图层 -> 粉红色0.5 ,蓝色 0.2 = 叠加在一起. = 颜色混合行为 -> 离屏渲染 组透明度 ->OpenGL

变换矩阵: 见课件

投影矩阵: 见课件 camera 不是相机 (观察者视角)

投影方式: 2种: 1: 正投影(平行投影) 图片绘制不管远近 1:1 进行绘制 ->显示 2D 效果

1. 透视投影 远小近大的效果. 3D图形

物体坐标: 物体本身的坐标系

viewController

并不是所有方法都要用~

7次课->给到提前学习~

不只讲1次. 案例

UIImage SDWebImage -> 图片解码 CoreGraphics

OpenGL ES -> 解压位图-> 类似的(png -> 位图)

必须品-> 逐行讲解