

Logic_视觉班课堂笔记[CC010]

- 日期: 2019年5月31日星期五
- 授课: CC老师
- 课程次数: 视觉班第10次课--共计(22次课)
- 主题: OpenGL ES 主题

课程内容(3D数学 & OpenGL ES)

- 了解GLKit
- 案例(1). 初探GLKit
- 案例(2). 使用OpenGL ES GLKit 加载图片

课程安排

- 08:00 - 09:00 第一节课
- 09:00 - 09:10 课间休息
- 09:10 - 10:00 第二节课
- 10:00 - 10:10 课程总结
- 10:10 - 10:30 课后答疑

课后作业:

1. 请在个人博客上更新一篇博文,将今晚所学总结!
 - OpenGL ES GLKit 常用API解析
- 要求:
 - i. 将课程内容加上自己的理解
 - ii. 更新的博客地址通过QQ私发给我.互相学习

一.上节课回顾

- OpenGL ES 渲染流程
- 顶点着色器

- 片元着色器
- 光栅化/图元装配 过程

节选

1.4.9 片元着色

最后一个可以通过编程控制屏幕上显示颜色的阶段，叫做片元着色阶段。在这个阶段中，我们使用着色器来计算片元的最终颜色（尽管在下一个阶段（逐片元的操作）时可能还会改变颜色一次）和它的深度值。片元着色器非常强大，在这里我们会使用纹理映射的方式，对顶点处理阶段所计算的颜色值进行补充。如果我们觉得不应该继续绘制某个片元，在片元着色器中还可以终止这个片元的处理，这一步叫做片元的丢弃（discard）。

如果我们需要更好地理解处理顶点的着色器和片元着色器之间的区别，可以用这种方法来记忆：顶点着色（包括细分和几何着色）决定了一个图元应该位于屏幕的什么位置，而片元着色使用这些信息来决定某个片元的颜色应该是什么。

- ❑ 将输入图元的数学描述转换为与屏幕位置对应的像素片元（fragment）。这一步也称作光栅化（rasterization）。
- ❑ 最后，针对光栅化过程产生的每个片元，执行片元着色器（fragment shader），从而决定这个片元的最终颜色和位置。
- ❑ 如果有必要，还需要对每个片元执行一些额外的操作，例如判断片元对应的对象是否可见，或者将片元的颜色与当前屏幕位置的颜色进行融合。

二. 上课笔记

OpenGL ES GLKit 图片加载(CC老师)

viewDidLoad 函数

1. OpenGL ES 相关初始化 setUpConfig 函数

- A. 初始化上下文&设置当前上下文
- B. 获取GLKView & 设置context
- C. 配置视图创建的渲染缓存区.
- D. 设置背景颜色

2. 加载图片顶点坐标&纹理坐标 setUpVertexData 函数

- A. 设置顶点数组(顶点坐标,纹理坐标)
- B. 开辟顶点缓存区
 - 创建顶点缓存区标识符ID
 - 绑定顶点缓存区.(明确作用)
 - 将顶点数组的数据copy到顶点缓存区中(GPU显存中)
- C. 打开读取通道.
 - 顶点坐标数据
 - 纹理坐标数据

3. 加载纹理数据(GLBaseEffect) setUpTexture 函数

- A. 获取纹理图片路径
- B. 设置纹理参数
- C. 使用苹果GLKit 提供GLKBaseEffect 完成着色器工作

GLKViewDelegate drawInRect 函数

- 1. 清除颜色缓存区
- 2. 准备绘制
- 3. 开始绘制

三. 课程总结

- GLKit 基本流程

四. 课后答疑

- 请视频 参考009--课程答疑