

# Logic\_视觉班课堂笔记[CC023]

---

- 日期: 2019年7月3日星期三
- 授课: CC老师
- 课程次数: 视觉班第23次课--共计(22次课)
- 主题: Metal 主题

## 课程内容

- Metal 初探

## 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

## 课程回顾

---

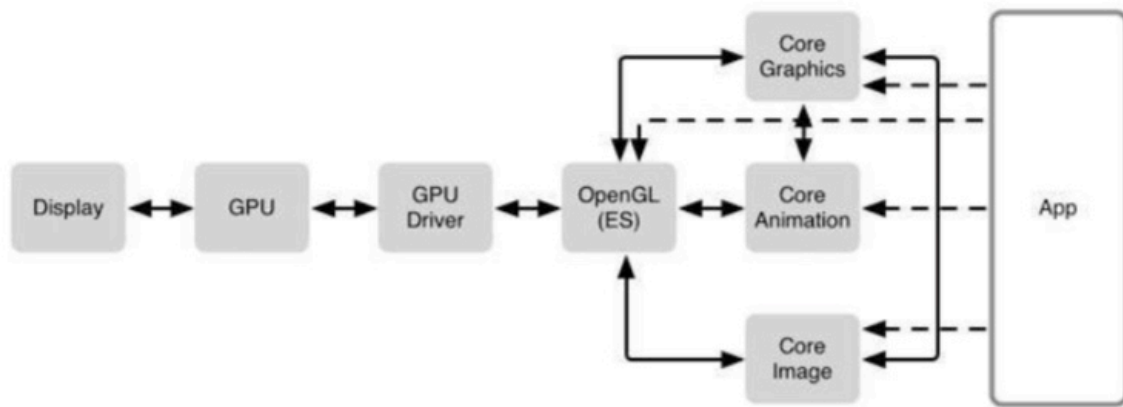
## 课程笔记

---

### 强调:

- OpenGL ES 是否可以模拟器. 可以,但是它借助CPU来完成GPU的计算.
- Metal 5S以上设备才可以真机执行! 至少A7处理器. 必须真机执行.不支持模拟器.

UIKit -> Core Animation -> Metal -> GPU Driver -> GPU



Metal Kit 基于 Metal 封装框架. 极少的代码完成Metal 需求.

核心类. MTKView

MTLDevice -> GPU

MTLCommandEncoder

- 图形渲染 MTLRenderCommandEncoder
- 计算. MTLComputeCommandEncoder
- 内存管道 MTLBlitCommandEncoder
- 并行编码多个图形显示任务: MTLParallelRenderCommandEncoder

Apple 建议: Separate Your Rendering Loop

```

_render = [[CCRenderer alloc] initWithMetalKitView:_view];
if (!_render) {
    NSLog(@"Renderer failed initialization");
    return;
}
_view.delegate = _render;

```

Apple 建议: Respond to View Events

响应视图事件:

- (void)mtkView:(nonnull MTKView \*)view drawableSizeWillChange:(CGSize)size;
- (void)drawInMTKView:(nonnull MTKView \*)view;
- 

MTKViewDelegate 有2个方法

### 1. `drawableSizeWillChange`

调用条件:窗口大小发生变化(Mac OS), 重新布局(设备方向更改)时

### 2. `drawInMTKView` ; 渲染循环 60FPS

Apple 建议: Metal Command Objects

*//MTLDevice 对象表示GPU.*

```
_view.device = MTLCreateSystemDefaultDevice();
```

*//所有应用程序需要与GPU交互的第一个对象是`MTLCommandQueue`对象*

```
_commandQueue = [_device newCommandQueue];
```

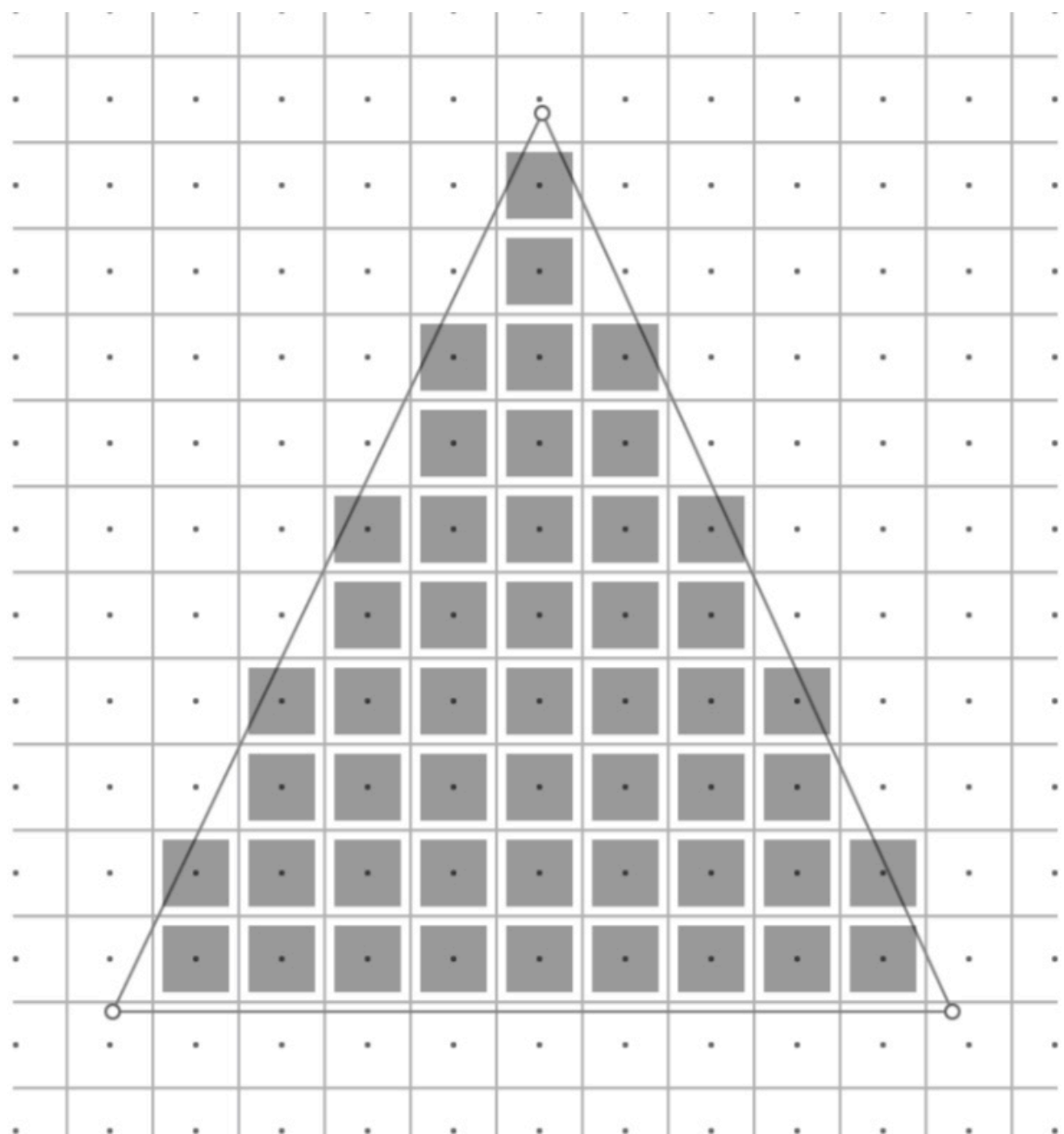
*//使用MTLCommandQueue 创建对象并且加入到MTCommandBuffer对象中去.*

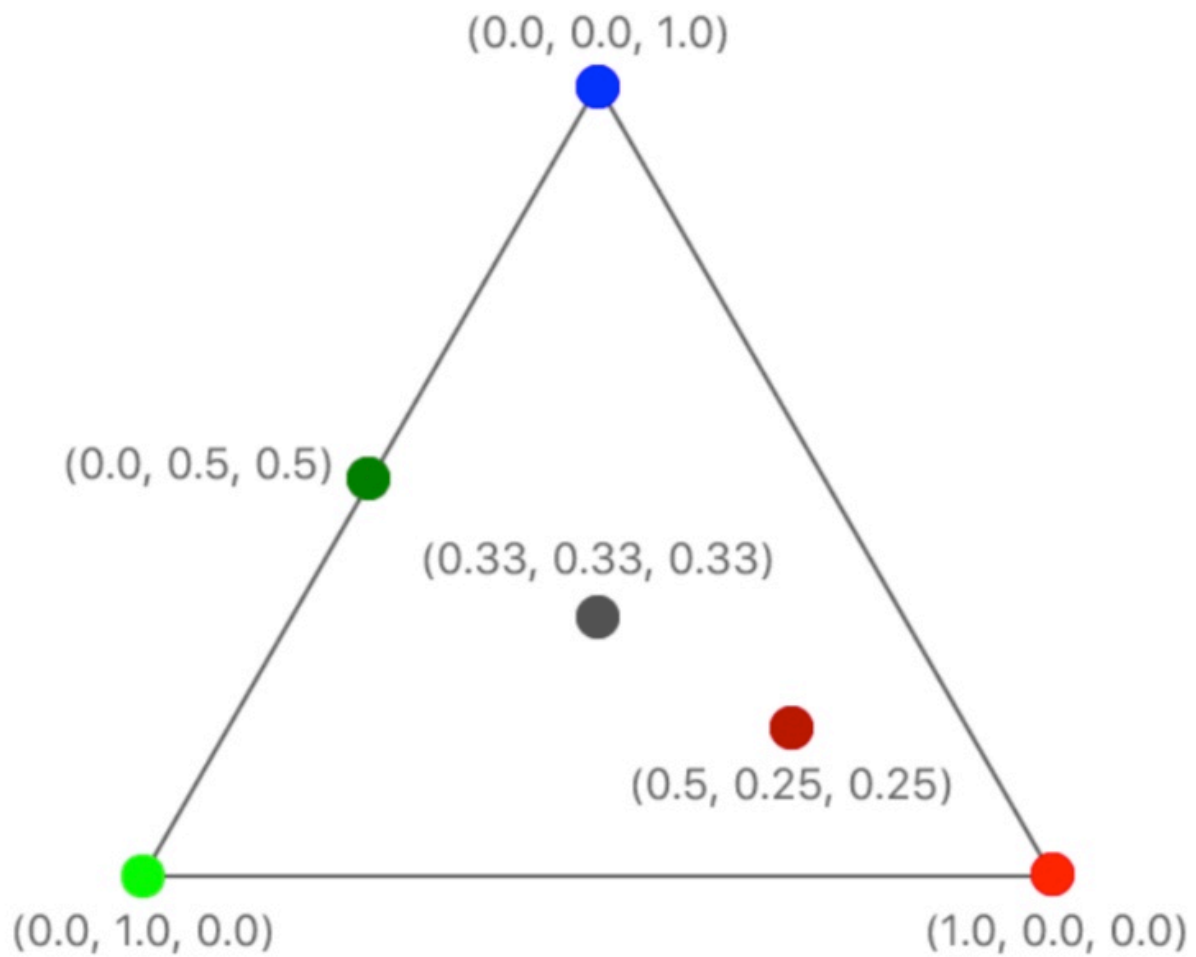
*//为当前渲染的每个渲染传递创建一个新的命令缓冲区*

```
id<MTLCommandBuffer> commandBuffer = [_commandQueue commandBuffer];  
commandBuffer.label = @"MyCommand";
```

管线3大阶段:

- 顶点函数
- 光栅化阶段/栅格化阶段
- 片元函数





## 课程总结

---

## 课程答疑

---