



逻辑教育
Logic education

Hello CC

Metal 主题 I 3 I

视觉班—Metal Shading Language

课程研发:CC老师

课程授课:CC老师

转载需注明出处,不得用于商业用途.已申请版权保护



逻辑教育
Logic education

学习目的:

学习Metal 程序中的Metal 着色语言

课程研发:CC老师

课程授课:CC老师

转载需注明出处,不得用于商业用途.已申请版权保护



Metal Shading Language

Metal 语言介绍

Metal着色语言是一个用来编写3D图形渲染逻辑和并行计算核心逻辑的编程语言,编写**Metal**框架的APP需要使用**Metal** 着色语言程序。

Metal 着色语言 与 **Metal** 框架配合使用,**Metal** 框架管理**Metal**着色语言的运行和可选编译选项。**Metal** 着色器语言使用**Clang**和**LVM**,编译器对于在GPU上的代码执行效率有更好的控制。:

Metal 与 C++ 11.0

Metal 这门语言是基于c++ 11.0标准设计的.它在c++基础是行多了一些拓展和限制.下面我们可以简单介绍介绍**Metal**着色语言与c++11.0 相比之下的修改和限制。

课程研发:CC老师

课程授课:CC老师



Metal Shading Language

Metal Restrictions 限制

如下的C++11.0的特性在Metal 着色语言中是不支持的.

- Lambda表达式
- 递归函数调用
- 动态转换操作符
- 类型识别
- 对象创建(`new`)和释放(`delete`)操作符
- 操作符 `noexcept`
- `goto` 跳转
- 虚函数修饰符
- 派生类
- 异常处理

课程研发:CC老师

课程授课:CC老师



逻辑教育
Logic education

Metal Shading Language

C++的标准库不可以在Metal 着色语言中使用

Metal 着色语言是有对于指针的使用限制.

Metal图形和并行计算函数用到的入参如果是指针使用地址空间修饰符
(device ,threadgroup ,constant)

不支持函数指针

Metal函数名不能命名为Main函数

课程研发:CC老师

课程授课:CC老师



Metal 数据类型

Metal 语言支持的标量类型

类型	描述
bool	布尔类型,取值范围true , false; true可以拓展为整数常量 1, false 可以拓展为整数常量0
char	有符号8-bit整数
unsigned char uchar	无符号8-bit整数
short	有符号16-bit整数
unsigned short ushort	无符号16-bit整数
int	有符号32-bit整数
unsigned int uint	无符号32-bit整数
half	一个16-bit 浮点数
float	一个32-bit浮点数
size_t	64-bit无符号整数,表示sizeof操作符的结果
ptrdiff_t	64-bit 有符号整数,表示2个指针的差
void	表示一个空的值集合

课程研发:CC老师

课程授课:CC老师



逻辑教育
Logic education

Metal 向量和矩阵数据类型

向量支持如下类型:

booln

charn

shortn

intn

ucharn

ushortn

uintn

halfn

floatn

向量中的n,指的是维度

课程研发:CC老师

课程授课:CC老师



逻辑教育
Logic education

Metal 向量和矩阵数据类型

矩阵支持如下类型:

halfn \times m

floatn \times m

n \times m分别指的是矩阵的行数和列数

课程研发:CC老师

课程授课:CC老师



逻辑教育
Logic education

访问 向量与矩阵

课程研发:CC老师

课程授课:CC老师

转载需注明出处,不得用于商业用途.已申请版权保护



逻辑教育
Logic education

缓存buffer

在Metal 中实现缓存靠的是一个指针.它指向一个在Device 或者 constant 地址空间中的内建或是开发者自定义的数据块.缓存可以被定在程序域域中,或是当做函数的参数传递.

课程研发:CC老师

课程授课:CC老师

转载需注明出处,不得用于商业用途.已申请版权保护



纹理Textures

纹理数据类型是一个句柄,它是一个一维/二维/三维纹理数据.而纹理数据对应这个一个纹理的某个level的mipmap的全部或者一部分

```
enum class access {sample ,read ,write};  
texture1d<T, access a = access::sample>  
texture1d_array<T, access a = access::sample>  
texture2d<T, access a = access::sample>  
texture2d_array<T, access a = access::sample>  
texture3d<T, access a = access::sample>  
texturecube<T, access a = access::sample>  
texture2d_ms<T, access a = access::read>
```



采样器 Samplers

Metal 支持的采样器状态和默认值

枚举名称(Enum Name)	有效值(Valid Value)	描述(Description)
coord	normalized, pixel	从纹理中采样时,纹理坐标是否需要归一化
address	clamp_to_edge, clamp_to_zero , mirrored_repeat , repeat	设置所有的纹理坐标的寻址模式
s_address , t_address , r_address,	clamp_to_edge, clamp_to_zero , mirrored_repeat , repeat	设置某个纹理坐标的寻址模式
filter	nearest , linear	设置纹理采样的放大和缩小的过滤模式
mag_filter	nearest , linear	设置纹理采样的放大过滤模式
min_filter	nearest , linear	纹理采样的缩小过滤模式
mip_filter	none , nearest , linear	纹理采样的minmap滤模式,如果是none,那么只有一个层纹理能生效
compare_func	none , less , less_qual , greater , greater_equal , equal , not_equal	为使用r纹理坐标做shadow map 设置比较测试逻辑,这个状态值的设置只可以在Metal 着色语言程序中完成

老师

课程授课:CC老师



函数修饰符

Metal 着色器语言支持下列的函数修饰符:

kernel,表示该函数是一个**数据并行计算着色函数**,它将被分配在一个一维/二维/三维的线程组网格中执行.

vertex,表示该函数是一个**顶点着色函数**,它将为顶点数据流中的每个顶点数据执行一次然后为每个顶点生成数据输出到绘制管线中去.

fragment,表示该函数是一个**片元着色函数**,它将片元数据流中的每个片元和其关联的数据执行一次然后为每个片元生成数据输出到绘制管线中去.



变量与参数的地址空间修饰符

Meta1 着色语言使用"地址空间修饰符号",来表示一个函数变量或是参数被分于那块内存区域。下面这些修饰符描述了不相交叠地址空间:

device — 设备地址空间

threadgroup — 线程组地址空间

constant — 常量地址空间

thread — thread地址空间



函数参数和变量

图形绘制或是并行计算着色函数的输入/输出都需要通过参数传递(除了常量地址空间变量和程序域中定义的采样器以外)。参数可以是如下之一:

device buffer — 设备缓存

constant buffer — 常量缓存

texture object — 纹理对象

sample object — 采样器对象

threadgroup — 线程共享的缓存



逻辑教育
Logic education

用于寻址缓存,纹理,采样器属性修饰符

为每个参数,指定一个属性修饰符. 指定明确的缓冲,纹理位置.

device 和 constant buffers — [[buffer (index)]]

texture — [[texture (index)]]

sampler — [[sampler (index)]]

threadgroup buffer — [[threadgroup (index)]]

课程研发:CC老师

课程授课:CC老师