5.1 基本数据类型

Cg 支持 7 种基本的数据类型：

1. float，32 位浮点数据，一个符号位。浮点数据类型被所有的 profile 支持（但是 DirectX8 pixel profiles 在一些操作中降低了浮点数的精度和范围）； 2. half，16 位浮点数据； 3. int，32 位整形数据，有些 profile 会将 int 类型作为 float 类型使用； 4. fixed，12 位定点数，被所有的 fragment profiles 所支持； 5. bool,布尔数据，通常用于 if 和条件操作符（?:），布尔数据类型被所有的 profiles 支持； 6. sampler*,纹理对象的句柄（the handle to a texture object），分为 6 类：sampler, sampler1D, sampler2D, sampler3D, samplerCUBE,和 samplerRECT。DirectX profiles 不支持 samplerRECT 类型，除此之外这些类型被所有的 pixel profiles 和 NV40 vertex program profile 所支持（CgUsersManual 30 页）。 7. string，字符类型，该类型不被当前存在的 profile 所支持，实际上也没有必要在 Cg 程序中用到字符类型，但是你可以通过 Cg runtime API 声明该类型变量，并赋值；因此，该类型变量可以保存 Cg 文件的信息。

Cg还提供了内置的向量数据类型(built-in vector data types)，内置的向量数据类型基于基础数据类型。例如：float4，表示 float 类型的 4 元向量；bool4，表示 bool类型 4 元向量

注意：向量最长不能超过 4 元，即在 Cg 程序中可以声明 float1、float2、float3、float4 类型的数组变量，但是不能声明超过 4 元的向量，例如： float5 array;//编译报错

向量初始化方式一般为： float4 array = float4(1.0, 2.0, 3.0, 4.0);

较长的向量还可以通过较短的向量进行构建： float2 a = float2(1.0, 1.0); float4 b = float4(a, 0.0, 0.0);

矩阵数据类型：最大的维数不能超过 4*4 阶。例如： float1x1 matrix1;//等价于 float matirx1; x 是字符，并不是乘号！ float2x3 matrix2;// 表示 2*3 阶矩阵，包含 6 个 float 类型数据 float4x2 matrix3;// 表示 4*2 阶矩阵，包含 8 个 float 类型数据 float4x4 matrix4;//表示 4*4 阶矩阵，这是最大的维数 矩阵的初始化方式为： float2x3 matrix5 = {1.0, 2.0, 3.0, 4.0, 5.0, 6.0}

5.2 数组类型

数组数据类型在 Cg 程序中的作用是：作为函数的形参，用于大量数据的传递。

Cg 中声明数组变量的方式和 C 语言类似：例如： float a[10];//声明了一个数组，包含 10 个 float 类型数据 float4 b[10];//声明了一个数组，包含 10 个 float4 类型向量数据 对数组进行初始化的方式为： float a[4] = {1.0, 2.0, 3.0, 4.0}; //初始化一个数组 要获取数组长度，可以调用“.length”，例如： float a[10]; //声明一个数组 int length = a.length;//获取数组长度

声明多维数组以及初始化的方式如下所示： float b[2][3] = {{0.0, 0.0, 0.0},{1.0, 1.0, 1.0}}; 对多维数组取长度的方式为： int length1 = b.length; // length1 值为 2 int length2 = b[0].length; // length2 值为 3

5.3 结构类型

结构体的声明以关键字 struct 开始，然后紧跟结构体的名字，接下来是一个 大括号，并以分号结尾（不要忘了分号）。大括号中是结构体的定义，分为两大 类：成员变量和成员函数。例如，定义一个名为 myAdd 的结构体，包含一个成 员变量，和一个执行相加功能的成员函数，然后声明一个该结构体类型的变量， 代码为

struct myAdd { float val; float add(float x) { return val + x; } }; myAdd s;

使用符号“• ”引用结构体中的成员变量和成员函数。例如： float a = s.value; float b = s.add(a);

5.4 接口（Interfaces）类型

Cg 语言提供接口类型，实际上，我本人很少见到使用接口类型的着色程序，原因在于，Cg 语言中的接口类型还不完善，不能被扩展和包含。此外，目前的GPU 编程大多只是针对独立的算法进行编码，规模较小，使用接口类型没有太大的优势。

5.5 类型转换

Cg 中的类型转换和 C 语言中的类型转换很类似。C 语言中类型转换可以是强制类型转换，也可以是隐式转换，如果是后者，则数据类型从低精度向高精度转换。在 Cg 语言中也是如此。

Cg 语言中对于常量数据可以加上类型后缀，表示该数据的类型，例如： 59 float a = 1.0; float b = a + 2.0h; //2.0h 为 half 类型常量数据，运算是需要做类型转换

常量的类型后缀（type suffix）有 3 种： z f ： 表示 float; z h: 表示 half; z x: 表示 fixed