3d Game Programming with DirectX 11

ianaesthetic

October 14, 2017

1 向量（Vector）

1.1 正交化（Orthogonalization）

对于一组向量 $\{\mathbf{v}_0, \mathbf{v}_1 \cdots \mathbf{v}_n\}$，正交化的过程为：

$$\mathbf{w}_0 = \mathbf{v}_0$$

$$\mathbf{w}_i = \mathbf{v}_i - \sum_{j=0}^{i} \mathrm{proj}_{\mathbf{w}_j}(\mathbf{v}_i)$$

$$\mathbf{w}_i = \frac{\mathbf{w}_i}{\| \mathbf{w}_i \|}$$

1.2 数学库 DirectXMath

数学库已经从原来的 XNAMath 转变为集成到 Windows SDk 的 DirectXMath 。相较于使用 <xnamath.h> ，现在是使用 <DirectXMath.h> 并且在所有的函数都将在命名空间 DirectX 下。

使用 SIMD 的向量 XMVECTOR 用于计算，而相对应的储存由专门的其他类型来储存，以 3D 为例 XMFLOAT3 。整形向量使用数组来定义。之间的转换定义为：

```
XMVECTOR XMLoadFloat3(const XMFLOAT3 *source);
void XMStoreFloat3(XMFLOAT3* destination, FXMVECTOR v);
```

单独获取一个分量或者修改一个分量的定义为（分量替换为 X，Y，Z，W）：

```
FLOAT XMVectorSetX(FXMVECTOR v, FLOAT x);
FLOAT XMVectorGetX(FXMVECTOR v);
```

在定义函数的时候，参数需要使用 FXMVECTOR 或者 CXMVECTOR 以更好的利用 SIMD 。函数的前三个 XMVECTOR 参数为 FXMVECTOR，后面的一律使用 CXMVECTOR 。引用参数依然为 XMVECTOR 。常量 XMVECTOR 定义为

```
XMVECTORF32 v = {1.0f, 1.0f, 1.0f, 1.0f};
```

一些很常用的 3D 函数列表（注意有一些返回标量的函数为了保持 SIMD 选择依然返回一个 XMVECTOR ，并且将所有的分量都设置为结果标量）：

```
XMVECTOR XMVectorSet(
    FLOAT x, FLOAT y, FLOAT z, FLOAT w
);
XMVECTOR XMVector3Length(FXMVECTOR v);
XMVECTOR XMVector3LengthSq(FXMVECTOR v);
XMVECTOR XMVector3Dot(FXMVECTOR v, FXMVECTOR u);
XMVECTOR XMVector3Cross(FXMVECTOR v, FXMVECTOR u);
XMVECTOR XMVector3Normalize(FXMVECTOR v);
BOOL XMVectorEqual(FXMVECTOR v, FXMVECTOR u);
```