**Chapter 1**

**シェーダープログラミングとは**

* 1. **固定機能**

シェーダーが生まれる前、DirectX9までは固定機能パイプラインというものが存在していました。この固定機能パイプラインはDirectX10で削除され、それ以降固定機能パイプラインは用意されなくなっています。これはOpenGL、OpenGLES、Sonyや任天堂などが提供する専用SDK(PS4、PS3、WiiUなどで使用できるDirectXのようなもの)でも同じで固定機能はグラフィックプログラミングの世界では過去のものとなっています。では固定機能と呼ばれるものがどのようなものか見ていきましょう。下記は固定機能を使って3Dポリゴンを表示しているコードです。

//-----------------------------------------------------------------------------

// ワールド\*ビュー\*プロジェクション行列を設定。

//-----------------------------------------------------------------------------

void SetupMatrices()

{

// ワールド行列を設定。

D3DXMATRIXA16 matWorld;

D3DXMatrixIdentity( &matWorld );

D3DXMatrixRotationX( &matWorld, timeGetTime() / 500.0f );

g\_pd3dDevice->SetTransform( D3DTS\_WORLD, &matWorld );

// ビュー行列を設定。

D3DXVECTOR3 vEyePt( 0.0f, 3.0f,-5.0f );

D3DXVECTOR3 vLookatPt( 0.0f, 0.0f, 0.0f );

D3DXVECTOR3 vUpVec( 0.0f, 1.0f, 0.0f );

D3DXMATRIXA16 matView;

D3DXMatrixLookAtLH( &matView, &vEyePt, &vLookatPt, &vUpVec );

g\_pd3dDevice->SetTransform( D3DTS\_VIEW, &matView );

// プロジェクション行列を設定。

D3DXMATRIXA16 matProj;

D3DXMatrixPerspectiveFovLH( &matProj, D3DX\_PI / 4, 1.0f, 1.0f, 100.0f );

g\_pd3dDevice->SetTransform( D3DTS\_PROJECTION, &matProj );

}

//-----------------------------------------------------------------------------

// ライトを設定。

//-----------------------------------------------------------------------------

void SetupLights()

{

//マテリアルを設定。

D3DMATERIAL9 mtrl;

ZeroMemory( &mtrl, sizeof( D3DMATERIAL9 ) );

mtrl.Diffuse.r = mtrl.Ambient.r = 1.0f;

mtrl.Diffuse.g = mtrl.Ambient.g = 1.0f;

mtrl.Diffuse.b = mtrl.Ambient.b = 0.0f;

mtrl.Diffuse.a = mtrl.Ambient.a = 1.0f;

g\_pd3dDevice->SetMaterial( &mtrl );

// ディフューズライトの向きとカラーを設定。

D3DXVECTOR3 vecDir;

D3DLIGHT9 light;

ZeroMemory( &light, sizeof( D3DLIGHT9 ) );

light.Type = D3DLIGHT\_DIRECTIONAL;

light.Diffuse.r = 1.0f;

light.Diffuse.g = 1.0f;

light.Diffuse.b = 1.0f;

vecDir = D3DXVECTOR3( cosf( timeGetTime() / 350.0f ),

1.0f,

sinf( timeGetTime() / 350.0f ) );

D3DXVec3Normalize( ( D3DXVECTOR3\* )&light.Direction, &vecDir );

light.Range = 1000.0f;

g\_pd3dDevice->SetLight( 0, &light );

g\_pd3dDevice->LightEnable( 0, TRUE );

g\_pd3dDevice->SetRenderState( D3DRS\_LIGHTING, TRUE );

// アンビエントライトを設定。

g\_pd3dDevice->SetRenderState( D3DRS\_AMBIENT, 0x00202020 );

}

//-----------------------------------------------------------------------------

// 描画

//-----------------------------------------------------------------------------

VOID Render()

{

// バックバッファとZバッファをクリア

g\_pd3dDevice->Clear( 0, NULL, D3DCLEAR\_TARGET | D3DCLEAR\_ZBUFFER,

D3DCOLOR\_XRGB( 0, 0, 255 ), 1.0f, 0 );

// 描画開始。

if( SUCCEEDED( g\_pd3dDevice->BeginScene() ) )

{

// ライトとマテリアルを設定。

SetupLights();

// ワールドビュープロジェクション行列を設定。

SetupMatrices();

// 頂点バッファを設定。

g\_pd3dDevice->SetStreamSource( 0, g\_pVB, 0, sizeof( CUSTOMVERTEX ) );

// 頂点のフォーマットを指定。

g\_pd3dDevice->SetFVF( D3DFVF\_CUSTOMVERTEX );

// 描画。

g\_pd3dDevice->DrawPrimitive( D3DPT\_TRIANGLESTRIP, 0, 2 \* 50 - 2 );

// 描画終了。

g\_pd3dDevice->EndScene();

}

// バックバッファの内容を表示。

g\_pd3dDevice->Present( NULL, NULL, NULL, NULL );

}

さて、上記のコードでどの部分が固定機能と呼ばれるものか分かりますか？実は、ワールドビュープロジェクション行列の設定、上のコードですと、ワールド行列、ビュー行列、プロジェクション行列、マテリアル、ライトの設定が固定機能になります。

ではGPUで実行される処理を考えながら、固定機能とはなんなのかを考えていきましょう。

**1.1.1グラフィックスパイプライン**

CPUからDraw命令送られてくると、下記のようにセットされた頂点バッファから頂点をフェッチ(取り出して)して、その頂点に陰影処理を行いスクリーン座標系に変換して対応するピクセルの色を決定します。

**メモリ**

**プロジェクション行列**

**ビュー行列**

**ワールド行列**

**GPU**

**確定したピクセルカラーをフレームバッファに書き込む**

**ピクセル処理(ピクセルの色を決める)**

**頂点をスクリーン座標系に変換**

**頂点フェッチ(頂点取り出し)**

**頂点バッファ**

**バックバッファ(解像度が1280\*720なら、RGBA32bitの1280\*720の２次元配列)**