

ESライブラリ&& ゲームプログラミング

3D編 - 第4回 マテリアル

マテリアル

- ・マテリアルは、モデルの「光の反射」を設定する
- ・モデルの色は、光の反射によって決定される
- ・設定できる反射は「ディフューズ」「アンビエント」「スペキュラー」「エミッシブ」
- ・ただし、エミッシブ以外の属性は、現実世界のようにライトで照らさないと色がつかない

概要

マテリアルは、表面属性ともいわれるもので、モデル表面の質感を表現する機能です。マテリアルは、光の反射と放射という属性を持ちます。光の反射には「ディフューズ反射」「アンビエント反射」「スペキュラー反射」があります。モデルの色は、これらの反射によって決定されます。また、光の放射には「エミッシブ」というモデル自体を発光させる属性があります。

ディフューズ反射(拡散反射)

ディフューズ反射は、光を一定の方向に反射し、オブジェクトに陰影を与えます。ライトからの角度で反射が決まり、ライトに向いているところほど明るくなります。オブジェクトの陰の色は、この反射の影響を大きく受けます。

アンビエント反射

アンビエント反射は、光の反射方向がない反射で、オブジェクト全体に均等に色を与えます。ライトからの角度に依存しないで反射します。

スペキュラー反射(鏡面反射)

スペキュラー反射は、オブジェクトの表面上での乱反射を再現し、オブジェクトに質感を与えます。物体の光沢を表す反射で、ライトの向きに影響されます。乱反射が小さいほど光沢が増し、大きいほど光沢が減ります。

スペキュラー反射では、光沢を出す光の「反射率」と「鮮明度」を設定します。反射率と鮮明度が大きいほど、オブジェクトの輝きが増します。

もっとも明るい部分(ハイライト)の色は、この反射の影響を大きく受けます。

エミッシブ

エミッシブは、オブジェクト自体から光を放射する機能です。エミッシブはライトの影響を受けないので、ライトが無くても物体自体が蛍光塗料を塗られたように光るような効果が得られます。ただし、別のオブジェクトを照らすことはできないので、光源の代わりにはなりません。

マテリアルの設定

オブジェクトにマテリアルを適用するには、D3DMATERIAL9構造体にマテリアルの属性を設定し、IDirect3DDevice9::SetMaterialメソッドに渡すことを行います。以後、すべてのオブジェクトのレンダリング時に、設定したマテリアルが適用されます。

```
// マテリアル設定
```

```
D3DMATERIAL9 material;  
ZeroMemory(&material, sizeof(material));  
material.Diffuse = D3DXCOLOR(0.5f, 0.0f, 0.0f, 1.0f); // ディフューズ反射  
material.Ambient = D3DXCOLOR(0.0f, 1.0f, 0.0f, 1.0f); // アンビエント反射  
material.Specular = D3DXCOLOR(1.0f, 1.0f, 1.0f, 1.0f); // スペキュラー反射  
material.Emissive = D3DXCOLOR(0.2f, 0.2f, 0.2f, 1.0f); // エミッシブ  
material.Power = 5.0f; // スペキュラー反射の鮮明度
```

```
// マテリアルの適用(pD3DDeviceは初期化済みのDirect3DDevice9オブジェクト)  
pD3DDevice->SetMaterial(&material);
```

マテリアルにスペキュラーを設定しても、デフォルトでは無効になっているため効果が現れません。スペキュラーを有効にするには、IDirect3DDevice9::SetRenderStateメソッドの1つ目の引数に「D3DRS_SPECULARENABLE」、2つ目の引数に「TRUE」を指定し、レンダリングステートの変更を行います。

```
// スペキュラー演算を有効にする
pD3DDevice->SetRenderState(D3DRS_SPECULARENABLE, TRUE);
```

ESライブラリでは、モデルごとにマテリアルの設定ができるようになっています。モデルは、描画前にマテリアルをデバイスに適用してから描画するようになっています。("Model.cpp" Draw関数)

課 題

モデルにマテリアルを設定しましょう。

(1)以下のプログラムを適切な場所に追加しましょう。

```
// マテリアル設定
Material mat;
mat.Diffuse = Color(0.5f, 0.5f, 0.5f);
mat.Ambient = Color(0.2f, 0.2f, 0.2f);
mat.Specular = Color(0.3f, 0.3f, 0.3f);
mat.Emissive = Color(0.0f, 0.0f, 0.0f);
mat.Power = 1.0f;
m_pShape->SetMaterial(mat);
```