ESライブラリ＆＆
ゲームプログラミング

○　３Ｄ編 - 第５回　ライト　○

ライト

・モデルにマテリアルを設定しただけでは色がつかない
・ライトでモデルを照らし、モデルが光を反射することによって色が見えるようになる
・ライトには「ディレクショナルライト」「ポイントライト」「スポットライト」がある
・「アンビエントライト」というすべてのオブジェクトに均一に光を当てる特殊なライトもある

概要

モデルにマテリアルを設定しただけでは色が付きません。モデルは実世界と同じように、光で照らしてそれを反射することにより色が付き、見えるようになります。

Direct3Dでは、モデルを照らす光を「ライト」という機能で表現します。ライトもマテリアル同様、ディフューズ色、アンビエント色、スペキュラ色があり、それぞれマテリアルの同名の反射に反応するようになっています。モデルの色は、ライトとマテリアルの同じ属性どうしが演算されたものになります。

ライトは、デバイスの上限まで、同時にいくつでも使用することができます。さまざまな光を再現できるように、それぞれのライトには位置や方向、減衰率といった属性と色が設定できます。色は、マテリアルと同じように、ディフューズ、アンビエント、スペキュラーがありますが、マテリアルと異なり、これらは互いの色に影響を与えることはありません。

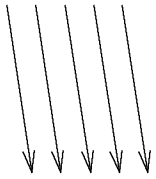
ライトの属性は、D3DLIGHT9構造体に設定します。この構造体のメンバは以下のようになっています。

```
D3DLIGHTTYPE    Type;         // ライトの種類
D3DCOLORVALUE   Diffuse;      // 光のディフューズ色
D3DCOLORVALUE   Specular;     // 光のスペキュラー色
D3DCOLORVALUE   Ambient;      // 光のアンビエント色
D3DVECTOR       Position;     // ライトの位置(ワールド座標)
D3DVECTOR       Direction;    // 光の方向ベクトル
float           Range;        // ライトの有効範囲(0.0～FLT_MAXの平方根)
float           Falloff;      // スポットライトのフォールオフ
float           Attenuation0; // 定常減衰係数
float           Attenuation1; // 線形減衰係数
float           Attenuation2; // 平方減衰係数
float           Theta;        // スポットライト内側の円錐の角度(０～πラジアン)
float           Phi;          // スポットライト外側の円錐の角度(０～πラジアン)
```

D3DLIGHT9構造体は、３つのライトに必要なメンバが定義されています。ライトの種類によって使用するメンバが異なります。

ディレクショナルライト(平行光源)

ディレクショナルライトは、地球を照らす太陽光のように、無限大の距離から一方向に平行な光を照射します。

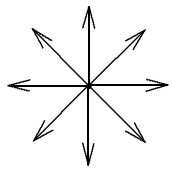


ディレクショナルライトは「光の色」と「光の方向」の属性を持ちます。D3DLIGHT9構造体は、Type, Diffuse, Specular, Ambient, Directionメンバを設定します。

```
D3DLIGHT9   light;
ZeroMemory(&light, sizeof(light));
light.Type      = D3DLIGHT_DIRECTIONAL;                 // ディレクショナルライト
light.Diffuse   = D3DXCOLOR(1.0f, 1.0f, 1.0f, 1.0f);    // ディフューズ色
light.Specular  = D3DXCOLOR(1.0f, 1.0f, 1.0f, 1.0f);    // スペキュラー色
light.Ambient   = D3DXCOLOR(1.0f, 1.0f, 1.0f, 1.0f);    // アンビエント色
light.Direction = D3DXVECTOR3(0.0f, -1.0f, 0.0f);       // ライトの方向ベクトル
```

ポイントライト

ポイントライトは、ライトの位置から全方向に一様に照らす電球のようなライトです。

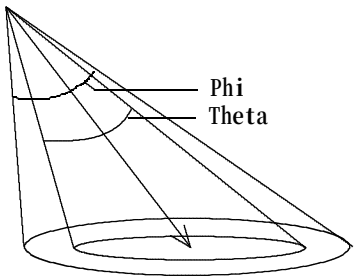


ポイントライトは「光の色」「光源の位置」「光の減衰」「光の有効範囲」の属性を持ちます。D3DLIGHT9構造体は、Type, Diffuse, Specular, Ambient, Position, Range, Attenuation0～2メンバに値を設定します。

```
light.Type         = D3DLIGHT_POINT;                    // ポイントライト
light.Diffuse      = D3DXCOLOR(1.0f, 1.0f, 1.0f, 1.0f); // ディフューズ色
light.Specular     = D3DXCOLOR(1.0f, 1.0f, 1.0f, 1.0f); // スペキュラー色
light.Ambient      = D3DXCOLOR(1.0f, 1.0f, 1.0f, 1.0f); // アンビエント色
light.Position     = D3DXVECTOR3(10.0f, 15.0f, 0.0f);   // ライトの位置(左からxyz)
light.Range        = 20.0f;                             // ライトの有効範囲
light.Attenuation0 =  1.0f;                             // 減衰定数
```

スポットライト

スポットライトはその名のとおり、ライトの位置から光が一定方向に照射され、その照射範囲は距離が遠くなるにつれ広がります。



光の照射範囲はライトの位置を頂点とする円錐状になり、この円錐はコーンとも呼ばれます。また、スポットライトには「フォールオフ」と呼ばれる減衰があります。フォールオフとは、円錐の内側から外側に向かって起こる光の減衰のことです。スポットライトは光の減衰によって「内側の明るい部分」と「外側の暗い部分」に分かれます。

スポットライトは「光の色」「光の方向」「光源の位置」「光の減衰」の属性を持ちます。D3DLIGHT9構造体はすべてのメンバを使用します。ライトの位置を設定するPositionメンバとともに、ライトの照射方向を決めるDirectionメンバの設定が必要になります。また、ライトの強度が最も高いコーンの角度を指定するThetaメンバ、コーン全体の広がりを決定するPhiメンバなどの設定も重要です。Falloffメンバは、ThetaからPhiまでのライト強度の減衰量を決定します。

```
light.Type         = D3DLIGHT_SPOT;                     // スポットライト
light.Diffuse      = D3DXCOLOR(1.0f, 1.0f, 1.0f, 1.0f); // ディフューズ色
light.Specular     = D3DXCOLOR(0.5f, 0.5f, 0.5f, 1.0f); // スペキュラー色
light.Ambient      = D3DXCOLOR(0.2f, 0.2f, 0.2f, 1.0f); // アンビエント色
```

```
light.Position     = D3DXVECTOR3(1.0f, 15.0f, 2.0f);     // ライトの位置
light.Direction    = D3DXVECTOR3(0.0f, -1.0f, 0.0f);     // ライトの方向ベクトル
light.Range        = 12.0f;                              // ライトの有効範囲
light.Falloff      =  1.0f;                              // フォールオフ
light.Attenuation0 =  1.0f;                              // 定常減衰係数
light.Theta        = D3DXToRadian(10.0f);                // 内側のコーンの角度
light.Phi          = D3DXToRadian(30.0f);                // 外側のコーンの角度
```

ライトの設定

D3DLIGHT9構造体に設定したライトは、IDirect3DDevice9::SetLightメソッドを呼び出すことにより、デバイスに割り当てられますが、SetLightメソッドはライトを割り当てるだけなので、このままでは使用されません。IDirect3DDevice9::LightEnableメソッドでライトを有効にすることで、オブジェクトを照らすようになります。

```
pD3DDevice->SetLight(0, &light);    // ライトをインデックス値０に割り当てる
pD3DDevice->LightEnable(0, TRUE);   // インデックス値０のライトを有効にする
```

照明演算はデフォルトで有効になっています。独自の照明演算を行う場合は、明示的に無効にする必要があります。照明演算の無効化は、IDirect3DDevice9::SetRenderStateメソッドの１つ目の引数に「D3DRS_LIGHTING」、２つ目の引数に「FALSE」を指定することにより行います。

```
// 照明演算を無効に設定
pD3DDevice->SetRenderState(D3DRS_LIGHTING, FALSE);
```

照明演算を無効にすると、LightEnableメソッドで設定した状態にかかわらず、すべてのライトが使用されなくなります(以下のアンビエントライトは除く)。

ESライブラリでは、ライトの割り当てと有効化をCDXGraphics9クラスのSetLight関数で行うことができます。

アンビエントライト(環境光)

Direct3Dには、３つのライト以外に、アンビエントライトという特殊なライトがあります。このライトは、すべてのオブジェクトに対し、方向に関係なく均一に当たります。すべてのオブジェクトのマテリアルのアンビエント反射に影響し、一定の明るさを提供します。
アンビエントライトは、IDirect3DDevice9::SetRenderStateメソッドで設定します。このメソッドの１つ目の引数に「D3DRS_AMBIENT」、２つ目の引数に色を指定します。

```
// アンビエントライトの設定
pD3DDevice->SetRenderState(D3DRS_AMBIENT, D3DXCOLOR(0.5f, 0.5f, 0.5f, 1.0f));
```

課　　題

モデルをライトで照らし、マテリアルが反映されるようにしましょう。

(１)モデルを上からの平行光源で照らし、マテリアルが反映されるようにします。以下のプログラムを適切な場所追加しましょう。

```
// ライト設定
Light   light;
light.Type      = Light_Directional;            // ディレクショナルライト
light.Diffuse   = Color(1.0f,  1.0f, 1.0f);     // ディフューズ色
light.Ambient   = Color(1.0f,  1.0f, 1.0f);     // アンビエント色
light.Specular  = Color(1.0f,  1.0f, 1.0f);     // スペキュラー色
light.Direction = Vector3_Down;                 // ライトの方向(光は上から下へ進む)
GraphicsDevice.SetLight(light);
```

(２)マテリアルやライトの色を変更してみましょう。

(３)エミッシブ、アンビエント、ディフューズ、スペキュラーがどのように適用されるか確認してみましょう。

１．最初にライトの色を白にし、マテリアルがそのままの色で反映されるようにします。ライトの設定を(１)の状態にしましょう。

２．ライトを無効にします。以下のプログラムを適切な場所に追加しましょう。

```
// ０番のライトを無効にする
GraphicsDevice.EnableLight(0, FALSE);
```

３．マテリアルにエミッシブのみ設定します。マテリアルを以下のように設定し、どのような色になるか確認しましょう。

```
mat.Diffuse  = Color(0.0f, 0.0f, 0.0f);
mat.Ambient  = Color(0.0f, 0.0f, 0.0f);
mat.Specular = Color(0.0f, 0.0f, 0.0f);
mat.Emissive = Color(0.0f, 0.0f, 0.5f);
```

４．ライトを有効にし、ます。２．を以下のように変更しましょう。また、エミッシブに影響があるかも確認しましょう。

```
// ０番のライトを有効にする
GraphicsDevice.EnableLight(0, TRUE);
```

５．マテリアルにアンビエントのみ設定します。マテリアルを以下のように設定し、どのような色になるか確認しましょう。

```
mat.Diffuse  = Color(0.0f, 0.0f, 0.0f);
mat.Ambient  = Color(0.5f, 0.0f, 0.0f);
mat.Specular = Color(0.0f, 0.0f, 0.0f);
mat.Emissive = Color(0.0f, 0.0f, 0.0f);
```

６．マテリアルにディフューズのみ設定します。マテリアルを以下のように設定し、どのような色になるか確認しましょう。

```
mat.Diffuse  = Color(0.0f, 0.5f, 0.0f);
mat.Ambient  = Color(0.0f, 0.0f, 0.0f);
mat.Specular = Color(0.0f, 0.0f, 0.0f);
mat.Emissive = Color(0.0f, 0.0f, 0.0f);
```

７．マテリアルにスペキュラーのみ設定します。マテリアルを以下のように設定し、どのような色になるか確認しましょう。

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

// スペキュラー演算を有効にする
GraphicsDevice.SetRenderState(SpecularEnable, TRUE);
```

８．「mat.Power」の数値を変更し、スペキュラーがどのように変更されるか確認しましょう。

９．マテリアルを以下のように設定し、どのように色が合成されているか確認しましょう。

```
mat.Diffuse  = Color(0.0f, 0.5f, 0.0f);
mat.Ambient  = Color(0.5f, 0.0f, 0.0f);
mat.Specular = Color(0.5f, 0.5f, 0.5f);
mat.Emissive = Color(0.0f, 0.0f, 0.5f);
mat.Power    = 1.0f;
```