

XNA Game Studio

ゲームプログラミング

3D編 - 第6回 マテリアル・ライト・フォグ

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
- ・設定できる反射の種類は「ディフューズ」「スペキュラー」「エミッシブ」「透明度」

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」「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。

スペキュラー反射

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

XNAでは、マテリアルの設定をエフェクトクラスで行います。エフェクトクラスを継承して作られているBasicEffectクラスもマテリアルをサポートしています。

ディフューズ反射はDiffuseColorメンバ、スペキュラ反射はSpecularColorメンバとSpecularPowerメンバ、エミッシブはEmissiveColorメンバで設定します。また、モデルの透明度もマテリアル扱いとなっています。透明度はAlphaメンバで、赤・緑・青ごとに設定できるようになっています。

マテリアルを各メンバに設定すると、以後、BasicEffect.Begin~Endメソッドに囲まれたすべてのプリミティブの描画時に、設定した反射が適用されます。

```
// マテリアルの設定(effectは、生成済みのBasicEffect型の変数)
// 色はVector3で設定します(左から赤、緑、青。0.0~1.0で設定)
effect.DiffuseColor = new Vector3(0.5f, 0.0f, 0.0f); // ディフューズ反射
effect.SpecularColor = new Vector3(1.0f, 1.0f, 1.0f); // スペキュラー反射
effect.SpecularPower = 5.0f; // スペキュラー反射の鮮明度
effect.EmissiveColor = new Vector3(0.2f, 0.2f, 0.2f); // エミッシブ
effect.Alpha = new Vector3(1.0f, 1.0f, 1.0f); // 透明度
```

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、光を反射し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」と「アンビエントライト」がある

概要

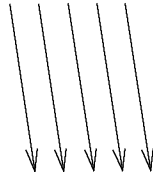
モデルにマテリアルを設定しただけでは色が付きません。モデルは実世界と同じように、光で照らしてそれを反射することにより色が付き、見えるようになります。

XNA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、ディフューズ色と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。

BasicEffectには、「ディレクショナルライト」と「アンビエントライト」という2種類のライトが用意されており、ディレクショナルライトは、同時に3つ使用できるようになっています。

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。BasicEffectクラスでは、ディレクショナルライトが同時に3つ使用できるようになっており、DirectionalLight0~2のメンバで定義されています。それぞれに、ディフューズ色のDiffuseColorメンバ、スペキュラ色のSpecularColorメンバ、光の方向を示すDirectionメンバがあります。

```
// BasicEffectのディレクショナルライト0を設定(effectは、生成済みのBasicEffect型の変数)
// ライトの方向ベクトル
effect.DirectionalLight0.Direction = new Vector3(0.0f, -1.0f, 0.0f); // 真下
// ディフューズ色(色もVector3で設定します(左から赤、緑、青。0.0~1.0で設定)
effect.DirectionalLight0.DiffuseColor = new Vector3(1.0f, 1.0f, 1.0f);
// スペキュラ色
effect.DirectionalLight0.SpecularColor = new Vector3(1.0f, 1.0f, 1.0f);
```

アンビエントライト(環境光)

BasicEffectクラスには、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、モデルのすべての面に対し、(法線の)方向に関係なく均一に当たり、一定の明るさを提供します。

アンビエントライトは、AmbientLightColorメンバに設定します。

```
// アンビエントライトの設定(effectは、生成済みのBasicEffect型の変数)
// 色はVector3で設定します(左から赤、緑、青。0.0~1.0で設定)
effect.AmbientLightColor = new Vector3(0.5f, 0.5f, 0.5f);
```

ライトの有効化

ライトの設定が終わったら、LightingEnabledメンバにtrueを設定し、ライトを有効化します。また、ライトを無効化したい場合は、このメンバにfalseを設定します。

また、読み込んだモデルに含まれるメッシュのエフェクトには、デフォルトライトが設定されているので、EnableDefaultLightingメソッドを呼び出せば、設定済みのライトが使用できます。

```
protected override void Draw(GameTime gameTime)
{
    graphics.GraphicsDevice.Clear(Color.CornflowerBlue);
    :
    // モデル描画
    foreach (ModelMesh mesh in model.Meshes)           // メッシュのループ
    {
        // エフェクトの設定
        foreach (BasicEffect effect in mesh.Effects)    // エフェクトのループ
        {
            // マテリアルや変換行列の設定を行う
            effect.EnableDefaultLighting();             // デフォルトライトの有効化

            effect.World = modelWorld;                   // ワールド変換行列の設定
            effect.View = viewMatrix;                    // ビュー変換行列の設定
            effect.Projection = projMatrix;              // 射影変換行列の設定
        }

        // メッシュ描画
        mesh.Draw();
    }
    :
}
```

ライトの計算は頂点ごとに行われていますが、これをピクセル単位で行うように変更することもできます。BasicEffectクラスのPreferPerPixelLightingメンバをtrueにすればピクセル単位のライティングになり、falseにすると頂点ごとのライティングになります。

フォグ

- ・フォグとは、頂点との距離に応じて霧がかかったようにぼやけさせる処理
- ・遠くから徐々にフェードインしながら現れるような効果が得られる

概要

フォグとは、ポリゴンを視点との距離に応じて霧がかかったようにぼやけて見えるようにする効果のことです。フォグをかけることで、オブジェクトを霧に包まれたようにしたり、徐々に現れたり徐々に消えるような効果を得ることができます。

BasicEffectのフォグは、「開始点」「終了点」「フォグの色」が設定できるようになっています。フォグの強さ(かかり具合)は、視点から頂点の距離(z値)をもとに、以下の式で計算されます。

$$\text{フォグの強さ} = \frac{\text{end} - d}{\text{end} - \text{start}}$$

start = フォグ効果が始まる距離(開始点)

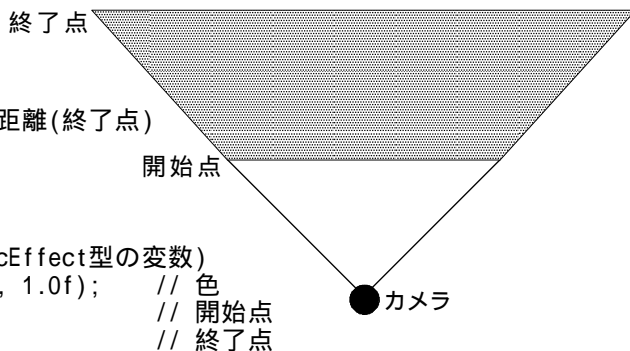
end = フォグ効果がこれ以上増加しなくなる距離(終了点)

d = 視点から頂点までの距離

フォグは、以下のように設定します。

```
// フォグの設定(effectは、生成済みのBasicEffect型の変数)
effect.FogColor = new Vector3(0.0f, 0.0f, 1.0f); // 色
effect.FogStart = 1.0f;                          // 開始点
effect.FogEnd   = 100.0f;                        // 終了点
```

フォグの設定が終わったら、FogEnabledメンバにtrueを設定し、フォグを有効化します。また、フォグを無効化したい場合は、このメンバにfalseを設定します。



課題

- (1) デフォルトライトが適用されるようにプログラムを変更し、モデルが正しい色で表示されるようにしましょう。
- (2) マテリアルやライトの色を変更してみましょう。