

# XNA Game Studio

## ゲームプログラミング

### 3D編 - 第5回 モデルの表示

#### モデルの表示

- ・モデルデータはContentManagerクラスのLoadメソッドで読み込む
- ・対応している形式はx形式とFBX形式
- ・モデルの表示は、メッシュ単位でエフェクトを設定してから行う

#### モデルデータの読み込み

モデルは、コンテンツに登録しておけば、ContentManagerクラスのLoadメソッドで読み込むことができます。読み込むと、モデルを管理するModelクラス型の変数が取得できます。

Modelクラスには、「頂点バッファ」「エフェクト」「テクスチャ」など、モデルの表示に必要な情報がすべて含まれています。

なお、モデルのテクスチャは、モデルファイルをコンテンツに登録したとき(ビルド時)に処理されるようになっています。適切なフォルダ(相対パス)においておけば、モデルとともにXNB形式へ変換されます。よって、テクスチャの画像ファイルをコンテンツに登録する必要はありません。

#### モデルの表示

モデルの表示は、メッシュ単位で行います。モデルには複数のメッシュが含まれており、すべてのメッシュにエフェクトを設定する必要があります。また、メッシュには複数のエフェクトが含まれている場合があるので、すべてのエフェクトを使って描画します。

よって、1つのモデルを表示するプログラムは、メッシュのループと、さらにその中でエフェクトのループという2重ループ構造になります。

#### コーディング例

```
// フィールド変数へ追加
Matrix viewMatrix, projMatrix; // ビュー変換行列、射影変換行列
Model model; // モデル管理
:

protected override void Initialize()
{
    :
    // ビュー変換行列設定
    viewMatrix = Matrix.CreateLookAt(new Vector3(0.0f, 0.0f, 1.0f),
                                     Vector3.Zero, Vector3.Up);

    // 射影変換行列設定
    projMatrix = Matrix.CreatePerspectiveFieldOfView(
        MathHelper.ToRadians(45.0f),
        (float)GraphicsDevice.Viewport.Width /
        (float)GraphicsDevice.Viewport.Height,
        1.0f, 100.0f);
    :
}

protected override void LoadContent()
{
    :
    // TODO: use this.Content to load your game content here
    model = Content.Load<Model>("Player1");
    :
}
```

```
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.View = viewMatrix;
            effect.Projection = projMatrix;
        }

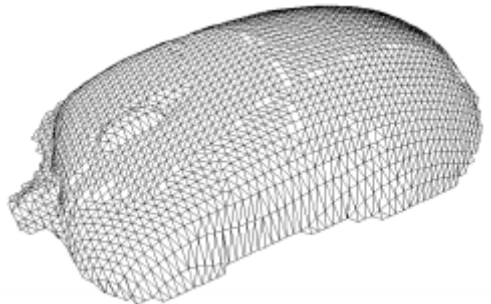
        // メッシュ描画
        mesh.Draw();
    }
    :
}
```

## メッシュ

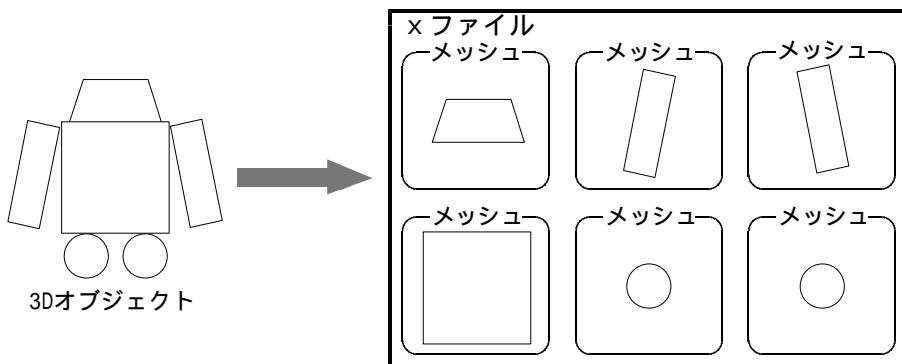
三角形を複数定義すると、網のようになります。XNAでは、三角形ポリゴン(面)の集合体のことをメッシュ(Mesh)と呼びます。モデルは面の集合体なので、メッシュで形成されているといえます。

メッシュといっても頂点の集まりなので、頂点バッファに格納します。しかし、メッシュには親子や兄弟といった関係を持たせ、他のメッシュに関連づけられているものもあるので、頂点バッファだけでは複雑なプログラムになってしまいます。

XNAでは、複雑な関係からなるメッシュを持つモデルも簡潔に扱えるように、Modelクラスを提供しています。



Modelクラスには、モデルの形状データがメッシュ単位で格納されています。1つのモデルの中に、複数のメッシュが格納されています。メッシュごとにマテリアルなどのエフェクトを設定したり、他のメッシュと親子、兄弟関係を持ったり、メッシュ単位で動きを制御したりすることができます。



## xファイル

xファイルは、XNAが対応するモデル格納形式のひとつで、位置や法線情報などを持った頂点データ、テクスチャの参照パス、フレーム情報、アニメーションデータなどをまとめたものです。

xファイルにはテキスト形式とバイナリ形式がありますが、XNAではテキスト形式のみ対応となっています。

## FBXファイル

FBXファイルは、XNAが対応するモデル格納形式のひとつで、Autodesk(オートデスク)社が提唱しています。このフォーマットもxファイル同様、ContentManager.Loadメソッドで読み込むことができます。

FBXフォーマットは、モデルデータを異なるアプリケーション間でスムーズに受け渡しができるように設計されており、頂点、モーションキー、モーフィング、マテリアル、テクスチャ、階層情報、ライト、カメラといったデータを、アプリケーションの垣根を越えて扱うことができるようになっています。

## 課題

モデルを読み込み、画面に表示しましょう。

モデルの色は、第6回「マテリアル・ライト・フォグ」でつけます