

ESライブラリ & & ゲームプログラミング

3D編 - 第16回 モデルデータの読み込み

モデルデータの読み込み

- ・モデルデータは、xファイルの読み込み または シンプルシェイプの生成を行って取得する
- ・モデルデータを格納する型には、MODEL型とアニメーション対応のANIMATIONMODEL型がある
- ・MODEL型は、アニメーションできないが、レイとの交差判定が行える
- ・ANIMATIONMODEL型は、アニメーションデータを読み込めるが、レイとの交差判定が行えない

概要

3Dオブジェクトを定義するモデルデータは、ESライブラリではxファイルまたはシンプルシェイプをもとに利用できます。モデルデータを管理する型(クラス)には2種類有り、固定モデルを扱うのに適したMODEL型と、アニメーションデータに対応したANIMATIONMODEL型があります。アニメーションデータの有無にかかわらず、どちらの型に対してもモデルデータを読み込むことができますが、それぞれ専用の関数を使って生成する必要があります。

xファイル

xファイルは、MicrosoftがDirectX10まで使っていたモデル格納形式で、位置や法線情報などを持った頂点データ、マテリアル、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、ESライブラリでは、どちらも扱えます。

テキスト形式は、情報が文字で確認できるため、テキストエディタで開いて編集することができます。そのかわり、ファイルサイズが大きくなり、読み込みに時間が掛かります。バイナリ形式はその逆で、ファイルをエディタで編集できないもののファイルサイズが小さくなり、読み込み速度がかなり向上します。サイズの大きいテキスト形式で読み込めないデータでも、バイナリ形式にすると読み込める場合があります(逆に、バイナリ形式だと正常に描画できないデータがテキスト形式では正常に描画できるといったこともあります)。

MODEL型

MODEL型は、アニメーション不要の固定モデルを扱うのに適しています。MODEL型独自の機能として、マテリアルの設定とレイとの交差判定を行えます。また、システムメモリに読み込むこともできます。レイとの交差判定は、内部的にロック処理を行うため、システムメモリに読み込んだ方が負荷が少なくなります。

MODEL型は、GraphicsDevice.CreateModelFromX関数またはCreateModelFromSimpleShape関数で生成します。

ANIMATIONMODEL型

ANIMATIONMODEL型は、アニメーションデータを読み込むことができるので、キャラクターなどの動きのあるモデルを扱うのに適しています。ただしMODEL型では可能なマテリアルの変更とレイとの交差判定は行えません。ANIMATIONMODEL型はGraphicsDevice.CreateAnimationModelFromX関数で生成します。

ファイルに含まれるアニメーションは、データごとにトラックに読み込まれます。トラックの有効/無効を設定するには、SetTrackEnable関数を呼び出します。モーシヨンプレンドに対応しているため、複数のトラックを有効にすると、それらがブレンドされたモーシヨンになります。あるトラックの設定が他のトラックに影響を及ぼさないため、トラックごとに有効/無効を設定する必要があります。モーシヨンを徐々に変更できるよう、SetTrackWeight関数によってブレンド比率の設定もできます。そのほか、アニメーションのループタイムの設定(SetTrackLoopTime関数)やアニメーションモーシヨンの位置の設定(SetTrackLoopTime関数)、アニメーション経過時間の設定(AdvanceTime関数)もできます。

シンプルシェイプ(SimpleShape)

シンプルシェイプは、立方体、円柱、多角形、球、トーラス(輪環)、ティーポットといったさまざまなテストによく利用される基本的な立体をシステムに生成させる機能です。CreateModelFromSimpleShape関数で生成しますが、形状の情報を格納したSimpleShape構造体の適切なメンバに値を設定する必要があります。この情報が間違っていると生成に失敗します。生成する形状によって設定するメンバが以下のように異なります。

- ・立方体

Typeメンバに'Shape_Box'を指定し、Width, Height, Depthメンバに、それぞれ立方体の幅、高さ、奥行きを設定します。

- ・円柱(シリンダー)

シリンダーという名前ですが、円錐の生成もできます。Typeメンバに'Shape_Cylinder'を指定し、Radius1メンバに手前の円の半径、Radius2メンバに奥の円の半径、Lengthメンバに円柱の長さ、Slicesメンバに円を構成する辺の数、Stacksメンバに柱部分の構成数(ここで指定した数の柱が積み重なり、Lengthで指定した長さになります)を設定します。

- ・多角形

Typeメンバに'Shape_Polygon'を指定し、Lengthメンバに辺の長さ、Slicesメンバに多角形を構成する辺の数を設定します。

- ・球

Typeメンバに'Shape_Sphere'を指定し、Radiusメンバに球の半径、Slicesメンバに正面から見たときの円を構成する辺の数、Stacksメンバに上から見たときの円を構成する辺の数を設定します。

- ・トーラス

ドーナツ状の立体です。Typeメンバに'Shape_Torus'を指定し、InnerRadiusメンバに内側の半径、OuterRadiusメンバに外側の半径、Sidesメンバにチューブを構成する円の辺の数、Ringsメンバに環を構成する円の数(外側 = 内側の円を構成する辺の数)を設定します。

- ・ティーポット

Typeメンバに'Shape_Teapot'を指定するだけです。そのほかのメンバーは参照しないので、未初期化でも問題ありません。

いずれの立体もテクスチャ座標をもたないため、テクスチャマッピングは行えません。