XNA Game Studio

# ゲームプログラミング

○　３Ｄ編 - 第３回　ワールド変換行列　○

## ワールド変換行列

・3Dグラフィクスは、モデルを3D空間に配置し、仮想世界を構築する
・「3D空間にモデルの配置を行う」には、モデルの座標をワールド座標に変換しなければならない
・ワールド変換行列は、モデルの座標変換を行い、「3D空間にモデルの配置を行う」ための行列
・頂点座標の変換には４行４列の行列が使われ、XNAではMatrix構造体で定義される

### ワールド変換行列

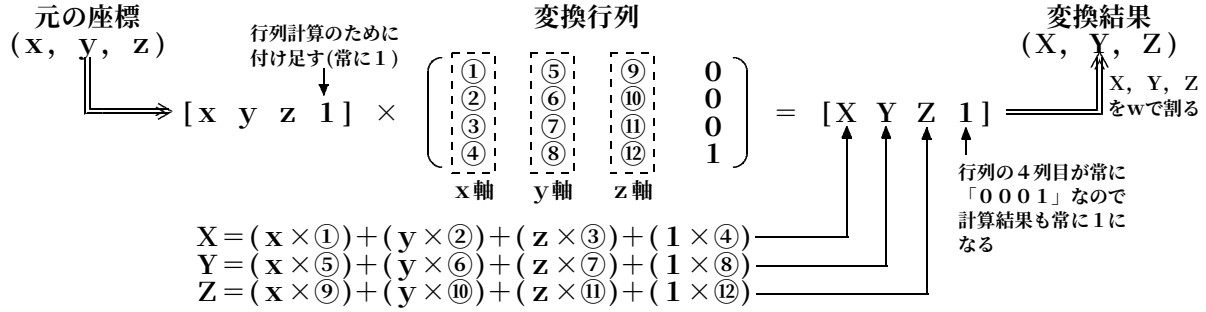
3Dグラフィクスでは、複数のモデルを3D空間に配置し仮想世界を構築しますが、各モデルを中心とした独自の座標(モデル座標)を共通の3D空間の座標(ワールド座標)に変換する必要があります。
モデルをワールド座標系に配置する場合、ワールド座標の値を直接指定することはできず、どこに、どのように配置するかを格納した「行列」を指定し、座標変換をして配置します。

座標変換は、変換行列によって行われます。モデル座標系をワールド座標系に変換する行列のことをワールド変換行列(ワールドトランスフォーム行列)と呼びます。ワールド変換行列には、回転、スケーリング、平行移動などが用いられます。回転しながら移動するといった場合は、回転のための行列と移動のための行列が必要となります。複数の変換行列を用いる場合は、行列を乗算し、合成します。

座標変換では、頂点を示すベクトルに変換行列が掛けられ、新しい点に変換されます。3Dで座標変換に使用する行列は、同次座標を用いるため４×４の行列でなければなりません。3Dベクトルと４×４の行列を掛けることはできないので、3Dベクトルに４つ目の成分wを１としたものを加え、4Dベクトルに変換してから処理が行われます。

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

◀座標変換で使用される４×４の行列




XNAでは、４×４の行列をMatrix構造体として定義しています。

### 座標変換行列

座標変換に用いる代表的な行列を以下に示します。ある点をｘ，ｙ，ｚのそれぞれについて$t_x$，$t_y$，$t_z$だけ移動する変換行列は、以下のようになります。

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & t_z & 1 \end{pmatrix}$$

オブジェクトを拡大縮小させる操作をスケーリングといいます。x，y，zのそれぞれのスケール(倍率)を$s_x$，$s_y$，$s_z$とすると、スケーリング行列は以下のようになります。

$$\begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

回転では、軸ごとに行列を使います。x軸回りの回転は、yz平面に対して影響を与えるので、以下のような行列になります。

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & \sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

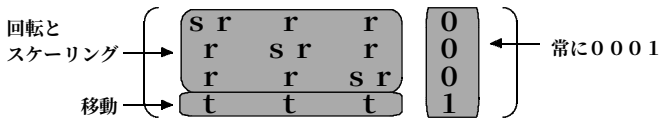
y軸回りの回転は、zx平面に対して影響を与えるので、以下のような行列になります。

$$\begin{pmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

z軸回りの回転は、xy平面に対して影響を与えるので、以下のような行列になります。

$$\begin{pmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

行列の各成分は、以下のように移動、スケーリング、回転と対応しています。




## 行列を作成するメソッド

Matrix構造体には、行列を作成するためのメソッドが多数用意されています。これらは、必要な情報を引数に指定するだけで目的の行列を作成することができるので、ベクトルや行列の知識が無くても扱うことができます。

| | |
|---|---|
| Matrix.Identity | 単位行列を作成します |
| Matrix.Invert | 逆行列を作成します |
| Matrix.Transpose | 行と列を入れ替えた転置行列を作成します |
| Matrix.CreateRotationX | x軸を回転軸とした回転行列を作成します |
| Matrix.CreateRotationY | y軸を回転軸とした回転行列を作成します |
| Matrix.CreateRotationZ | z軸を回転軸とした回転行列を作成します |
| Matrix.CreateFromYawPitchRoll | y，x，z軸を回転軸とした回転行列を作成します |
| Matrix.CreateFromAxisAngle | 任意のベクトルを回転軸とした回転行列を作成します |
| Matrix.CreateScale | スケーリング行列を作成します |
| Matrix.CreateTranslation | 平行移動行列を作成します |
| Matrix.CreateShadow | 頂点を平面に射影する行列を作成します |
| Matrix.CreateReflection | 平面の座標系を反映した行列を作成します |
| Matrix.CreateFromQuaternion | クォータニオンから回転行列を作成します |

## 行列の合成

複数の行列を適用したい場合は、行列を乗算して合成します。このとき、掛けた順に適用されます。回転行列に移動行列を掛けるのと、移動行列に回転行列を掛けるのとでは、行列の性質上、行列そのものが異なるものになります。

ワールド座標の原点をもとに計算が行われるので、(実世界と同じように)回転してから移動するのと、移動してから回転するのとでは、まったく別の状態となります。通常は回転行列、スケーリング行列、平行移動行列、またはスケーリング行列、回転行列、平行移動行列の順に掛けます。
このとき問題になるのが、回転行列を掛ける順番です。回転行列は、ｘ，ｙ，ｚ軸で個別の行列を用います。ｘ，ｙ，ｚ軸の順で掛けた行列と、ｚ，ｙ，ｘの順で掛けた行列は、別の行列になります。この順番は、そのときの状況により正解が変わるので、状況にあった順に掛けます。

```
// ワールド変換行列の作成
// ｘ軸回転(０度回転)
Matrix   rot_x = Matrix.CreateRotationX(MathHelper.ToRadians(0.0f));

// ｙ軸回転(４５度回転)
Matrix   rot_y = Matrix.CreateRotationY(MathHelper.ToRadians(45.0f));

// ｚ軸回転(０度回転)
Matrix   rot_z = Matrix.CreateRotationZ(MathHelper.ToRadians(0.0f));

// スケーリング(ｘ，ｙ，ｚ方向それぞれ1.5倍に拡大)
Matrix   scale = Matrix.CreateScale(new Vector3(1.5f, 1.5f, 1.5f));

// 平行移動(ワールド座標(0.0, 0.0, 3.0)に移動)
Matrix   trans = Matrix.CreateTranslation(new Vector3(0.0f, 0.0f, 3.0f));

// 各行列を合成する
Matrix   _world = rot_y * rot_x * rot_z * scale * trans;

// BasicEffectにワールド変換行列を設定(basicEffectは生成済みのBasicEffect型変数)
basicEffect.World = _world;
                    ⋮
(以降、すべてのモデルが「ｙ軸に45度回転して1.5倍に拡大、座標(0，0，3)」に描画されます)
```

## ワールド変換行列の設定

ワールド変換行列が作成できたら、BasicEffectのWorldメンバに設定します。以後、BasicEffect.Begin～Endメソッドに囲まれたすべてのオブジェクトの描画時に、設定したワールド変換行列が適用されます。しかし、モデルごとに配置座標などは変わりますので、それにあわせてワールド変換行列も変えるのが一般的です。よってほとんどの場合、モデルの描画前にWorldメンバの設定が必要となります。

```
// ワールド変換行列設定(ｚ軸時計回りに45度回転させ、ワールド座標(500.0, 0.0, 0.0)に配置)
BasicEffect basicEffect = new BasicEffect(GraphicsDevice, null);
basicEffect.World = Matrix.CreateRotationZ(MathHelper.ToRadians(10.0f)) *
                    Matrix.CreateTranslation(new Vector3(500.0f, 0.0f, 0.0f));
```

## モデル座標系(ローカル座標系)

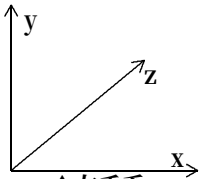
モデル座標系(ローカル座標系)は、モデル(3Dオブジェクト)が独自に持っている座標空間です。モデルは、頂点を指定して表現されています。これらの点はモデル座標系、いわゆるモデル独自の空間に定義されています。
この座標系の原点はオブジェクトの中心になります。中心は、必ずしも重心などの幾何学上の中心とは限りませんが、回転したりスケーリングしたりする場合の基準となる点となります。

## ワールド座標系

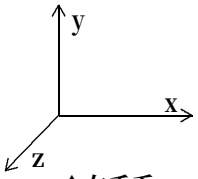
ワールド座標系は、デバイス内に表現される仮想的な世界が持つ座標系です。この世界は、モデルを配置するための空間です。ゲームなどの独自の世界は、モデル座標系で定義したオブジェクトをワールド座標系、つまりモデルを仮想世界に配置して独自の世界感を作成しています。このとき、モデル座標系からワールド座標系へ座標変換が行われます。この座標変換をワールドトランスフォーム(ワールド変換)と呼びます。
ワールド座標系は、3D空間における絶対座標系です。モデル座標系がモデルごとにあるのに対し、ワールド座標系は、デバイスに対し１つだけです。

## 右手座標

3D空間では、ｘ軸、ｙ軸、ｚ軸によって構成される座標系が用いられます。このとき、軸の方向付けによって２つの座標系が考えられます。ｘ軸とｙ軸で作られる平面が目の前にあると仮定したとき、平面からｚ軸が向こうを指している場合と、平面の手前を指している場合の２種類です。それらは、親指、人差し指、中指をそれぞれｙ軸、ｚ軸、ｘ軸に見立てたとき、左手で表されるか右手で表されるかによって、左手系あるいは右手系と呼ばれます。



▲左手系



▲右手系

XNAでは右手系が使われています。これは一般的なモデリングツールと同様ですが、DirectX9では、左手系が採用されています。

## 頂点とベクトル

3D空間中のモデルは、頂点によって構成されます。この頂点は、位置ベクトルによって空間のある位置を表します。その位置ベクトルは、通常は原点からの相対的な位置を表すものです。ベクトルは、どのような次元も考えられますが、ゲームでは２次元か３次元が使われます。それらの要素には、慣例としてｘ，ｙ，ｚという名前が使われます(ゲームでは単なる座標で問題ないですが、慣例として位置ベクトルという扱いになっています)。

XNAでは、行ベクトルが採用され、ベクトルをVector2，3，4構造体で表現します。OpenGLなど他の3Dライブラリでは、列ベクトルが採用されているものもあります。その場合には、変換行列の並びが変わります。

## 同次座標

ベクトルは平行移動をサポートしていません。そもそもベクトルは始点を持たない概念です。大きさと方向だけを持っていて、その始点は原点に固定されています。そのため、方向を保ったまま平行移動することができません。つまり、あるベクトルに、平行移動を表す別のベクトルを加えると、方向も変わってしまいます。これを回避するために、同次座標を導入する必要があります。

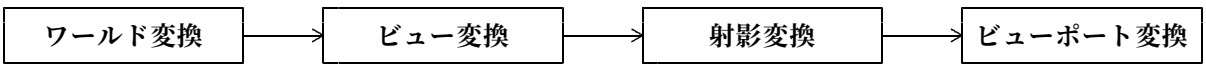
3Dベクトルの平行移動は、3D空間では不可能なので、いったん4D空間に持っていきます。4D空間内で平行移動を行い、その4D空間の断面を3D空間に変換するのです。

4D空間での点は、ｘ，ｙ，ｚに加えｗという仮想的な第４軸を基準とする第４の要素を持ちます。4Dベクトルの各要素をｗで割ったものが、3Dでの１点に対応します。このような4Dの座標系を同次座標系といいます。これは、ｗ＝１となる点群を4D空間の断面として、3D空間で認識しているということになり、このことを２つの空間で同じ同次頂点を指していると表現します。なお、ｗが０のときは除算することができないので、3D空間の無限遠にある点となります。

## ジオメトリ処理

XNAでは、モデルを描画するためのメソッドが呼び出されると、モデル座標で定義されるオブジェクトを「視点から捉えた」座標に変換するために、一連の座標変換が実行されます。一連の座標変換にはBasicEffectクラスではジオメトリパイプラインが用いられます(独自のEffectクラスを定義すれば、バーテックスシェーダを用いることができます)。

ジオメトリパイプラインとは、頂点データに対して、座標変換を行い、実際に描画できるデータに変換を行う処理のことです。頂点データは、ワールド、ビュー、射影、ビューポートの４つの座標変換を経てピクセルに変換されます。各座標変換には変換行列を用います。



▲ジオメトリパイプライン