

XNA Game Studio

ゲームプログラミング

ゲーム制作編 - 第2回 プレイヤーの表示

プレイヤーの表示

- ・プレイヤーのモデルを用意(x形式またはfbx形式)
- ・プレイヤーは、ワールド座標(0.0, 0.0, 0.0)に配置
- ・モデルを表示するには、カメラが必要
- ・カメラは、ワールド座標(0.0, 20.0, 0.0)に配置、視点を(0.0, 0.0, 6.0)に設定する
- ・2Dと3Dが混在する場合、レンダリングステートと描画順に注意

課題

プレイヤーのモデルを表示しましょう。

(1)プレイヤーを制御するための変数を宣言します。

ヒント1：プレイヤーのモデルを表示する必要がある

ヒント2：プレイヤーは移動する

ヒント3：上記のため、最低2つの変数を宣言。座標は3DなのでVector3型が適切

(2)プレイヤーのモデルを用意し、コンテンツに登録しましょう。

(3)プレイヤーを制御するための2つの変数を初期化しましょう。

ヒント1：モデルの読み込みは、LoadContentメソッド

ヒント2：プレイヤーの初期座標は(0.0f, 0.0f, 0.0f)、初期化はInitializeメソッド

(4)カメラの設定をします(プレイヤーをワールド座標に配置しただけでは表示できません)。

手順1：以下のビュー行列、プロジェクション行列を格納する変数を宣言します。

固定カメラなので、カメラの座標などの変数は不用です。変換行列で管理します。

```
// カメラ
```

```
Matrix viewMatrix, projMatrix;
```

手順2：ビュー行列をviewMatrixに設定します。

ヒント1：視点(0.0f, 20.0f, 0.0f)、注視点(0.0f, 0.0f, 6.0f)

ヒント2：ビュー行列の生成はMatrix.CreateLookAtメソッド

手順3：プロジェクション行列をprojMatrixに設定します。

仕様：視野角45.0度、アスペクト比はウィンドウ(ビューポート)と同じ横縦比、

前方クリップ面1.0、後方クリップ面100.0

ヒント1：プロジェクション行列の生成はMatrix.CreatePerspectiveFieldOfViewメソッド

ヒント2：角度を引数にする場合、ラジアン単位で指定。弧度法からラジアン単位への変換は

MathHelper.ToRadiansメソッド

(4)3Dモデルが適切に描画できるように、Drawメソッドで3Dモデル描画用のレンダリングステートを2つ設定しましょう。

ヒント1：1.深度バッファを有効にする 2.深度バッファへの書き込みができるようにする

ヒント2：3D編 - 第7回

(5)プレイヤーのモデルを描画するための以下のプログラムを完成させ、適切な場所に追加しましょう。

```
// プレイヤー描画
foreach (ModelMesh mesh in playerModel.Meshes)
{
    Matrix world = Matrix.????????????????( プレイヤーの座標 );
    // エフェクトの設定
    foreach (BasicEffect effect in mesh.Effects)
    {
        effect.EnableDefaultLighting();           // デフォルトライトの適用
        effect.World = world;
        effect.View =   ここは各自考えましょう;
        effect.Projection =   ここは各自考えましょう;
    }

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

- ヒント1：ワールド行列は、今回は回転・スケーリングを行わないため、移動行列がそのまま使用可
モデルによっては、回転させないと画面奥を向かないので、回転行列×移動行列になります
- ヒント2：描画を行うのは、レンダリングステート設定後
- ヒント3：得点(2D描画)との画面での前後関係を考慮し、どちらを先に描画するか決定する
- ヒント4：描画処理のメソッドで行う