

XNA Game Studio ゲームプログラミング

ゲーム制作編 - 第3回 プレイヤーの移動

プレイヤーの移動

- ・ でプレイヤーが左に移動し、 でプレイヤーが右に移動する

課題

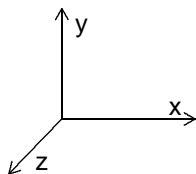
キー入力により、プレイヤーを移動させましょう。

(1) 内部処理のメソッドに、キー入力によってプレイヤーが移動する処理を作成しましょう。

仕様：左右のキーが押されるたびに、 $0.2f$ ずつ移動。移動範囲は $-12.0f \sim +12.0f$ まで

ヒント1：共通編 - 第1回

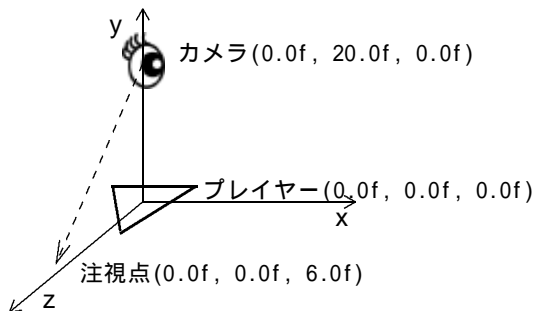
ヒント2：



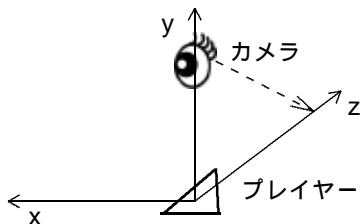
正しくプログラムを記述すると、 を押すと画面右に、 を押すと画面左に移動します。

(2) を押したら画面左に、 を押したら画面右に移動するようにします。

現在のプレイヤー、カメラの位置関係は、以下のようになっています。



XNAでは、右手系が採用されているため、 z 軸は画面手前に向かっていています。上の図を画面に投影すると、カメラが基点となって描画されますが、カメラ = 視点の方向は、画面奥方向となるため、以下のように、 y 軸を基点に180度回転したものとなります。



そのため、実際は x 軸について正しく移動しているにもかかわらず、見かけ上は左右が反対になっているように見えます。

3Dでは、モデルの向きや視点の設定の仕方により、向きが逆になっているように見える場合があります。そのため、上下左右のキー入力を素直に反映させる場合、向きや移動方向を逆にするなどの工夫が必要です。

手順1：モデルを画面奥方向(z軸マイナス方向)に向けます。モデルの回転を制御するための以下の変数を適切な場所に追加しましょう。

```
Vector3 playerRot;
```

手順2：初期化メソッドでplayerRot変数を初期化します。z軸のマイナス方向に向くように、数値を設定してください(角度の単位は弧度法でかまいません)。

手順3：プレイヤーのモデルを描画するときの世界行列を「回転行列×移動行列」に変更します。

ヒント1：回転行列を生成するのはMatrix.CreateFromYawPitchRollメソッド

ヒント2：playerRot変数が弧度法の場合、MathHelper.ToRadiansメソッドでラジアン単位に変換

ヒント3：MathHelper.ToRadians(playerRot.X)

手順4：カメラの視点方向も反転させます。注視点を正反対の座標にすれば、向きが逆になります。

ヒント1：注視点を(0.0f, 0.0f, 6.0f)の逆方向に変更

(3)左右のキーが押されたら、プレイヤーが移動方向に30度傾くようにしましょう。

ヒント1：傾きの回転軸と傾く方向に注意

ヒント2：どちらも押されていないときは傾かないようにする