XNA Game Studio
ゲームプログラミング

○　共通編 - 第１回　キーボード・マウス　○

キーボード

・キーボードの状態はKeyboardクラスのGetStateメソッドで取得
・Keyboard.GetStateメソッドは、KeyboardState構造体にキーボードの状態を返す
・キーが「押されているか」はKeyboardState構造体のIsKeyDownメソッド
・キーが「放されているか」はKeyboardState構造体のIsKeyUpメソッド

概要

キーボードの状態は、KeyboardクラスのGetStateメソッドで取得できます。このメソッドを呼び出すと、キーの状態がKeyboardState構造体に格納されます。

KeyboardState構造体には、いくつかのメソッドも定義されており、IsKeyDownメソッドは、引数で指定したキーが押されているかを調べることができます。また、IsKeyUpメソッドは、引数で指定したキーが放されているかを調べることができます。

特定のキーの状態を調べるには、Keys列挙体で定義されている定数を使用します。たとえば'Enter'キーは"Keys.Enter"、'A'キーは"Keys.A"となります。このように、Keys列挙体は、キーの名称そのままの名前となっています。

```
// キーボード状態の取得
KeyboardState keyState = Keyboard.GetState();

// キャラクタ移動
if (keyState.IsKeyDown(Keys.Left))
    // '←'が押されているときの処理
if (keyState.IsKeyDown(Keys.Right))
    // '→'が押されているときの処理
```

キーを押されたか放されたかを判定する

取得できるキーの状態は、「押されている」「押されていない」の２つだけです。押し下げられた瞬間や押し上げられた瞬間という情報を取得することはできません。これらの情報が欲しい場合は、前フレームのキーの状態と現フレームのキーの状態を使って判定します。

前フレームのキーの状態を準備するには、内部処理(Updateメソッド)で現在のキーの状態を変数にコピーしておきます。次のフレームでは、それが前フレームのキーの状態になるわけです。次フレームのキーの状態と、コピーしておいたキーの状態を比較し、状態が変わっていればキーが「押された」もしくは「放された」が起こったといえます。

キーの状態が前フレームで「押されていない」、現フレームで「押されている」場合、押し下げられた瞬間を表しています。逆に、前フレームでは「押されている」、現フレームでは「押されていない」場合、押し上げられた(放された)瞬間を表しています。

```
// フィールド変数
KeyboardState oldKeyboarState;       // 前フレームのキーボードの状態

protected override void Update(GameTime gameTime)
{
    // Allows the game to exit
                ⋮
    KeyboardState keyState = Keyboard.GetState();
```

```
        // スペースキーが「押された」かどうか調べる
        if (keyState.IsKeyDown(Keys.Space) && oldKeyboardState.IsKeyUp(Keys.Space))
            // 現フレームでスペースキーが「押された」

        // 'Z'キーが「放された」かどうか調べる
        if (keyState.IsKeyUp(Keys.Z) && oldKeyboardState.IsKeyDown(Keys.Z))
            // 現フレームで'Z'キーが「放された」

                              ⋮
        // キーボードの状態の保存
        oldKeyboarState = keyState;

        base.Update(gameTime);
    }
```

マウス

・マウスの状態はMouseクラスのGetStateメソッドで取得
・Mouse.GetStateメソッドは、MouseState構造体にカーソルの座標やボタンの状態を返す
・カーソル座標はMouseState構造体のXメンバとYメンバ
・ボタンが「押されているか」はMouseState構造体のメンバとButtonState.Pressed を比較する
・ボタンが「放されているか」はMouseState構造体のメンバとButtonState.Releasedを比較する
・ホイールはMouseState構造体のScrollWheelValueメンバに累積値が設定される
・マウスはXbox 360に未対応

概要

マウスの状態は、MouseクラスのGetStateメソッドで取得できます。このメソッドを呼び出すと、MouseState構造体にカーソルの座標とボタンの状態を格納します。

```
// マウス状態の取得
MouseState mouseState = Mouse.GetState();
```

カーソルは、デフォルトでは非表示ですが、Gameクラス(基底クラス)のプロパティIsMouseVisibleにtrueを設定すると表示されます。falseを設定すると非表示になります。

```
// マウスカーソルの表示
this.IsMouseVisible = true;
```

マウスカーソルの座標は、MouseState構造体のXとYメンバで取得できます。非表示でも移動処理は行われています。また、MouseクラスのSetPositionメソッド(Mouse.SetPosition)では、カーソルを特定の座標に設定することができます。

```
Vector2 cursolPos = new Vector2((float)mouseState.X, (float)mouseState.Y);
```

ボタンの状態は、ButtonState列挙体で定義される定数との比較で行います。ボタンが押されているかは"ButtonState.Pressed"、放されているかは"ButtonState.Released"と比較します。
ボタンを表すメンバは、"LeftButton", "RightButton", "MiddleButton", "XButton1", "XButton2"があります。XButtonは、ボタンが多数あるマウスで、たいていブラウザの「戻る」や「進む」に割り当てられているボタンです。

```
// マウスボタン判定
if (mouseState.LeftButton  == ButtonState.Pressed)
    // マウスの左ボタンが押されているときの処理
if (mouseState.RightButton == ButtonState.Released)
    // マウスの右ボタンが放されているときの処理
```

ホイールの状態は、MouseState構造体のScrollWheelValueメンバで取得できますが、ホイールの移動量ではなく、アプリケーション起動時からの累積量となります。ホイールの移動量は、前フレームの値と現フレームの値を比較するなどして判定します。

課　　題

(１)キーボードでキャラクタが移動できるようにしましょう。

(２)キーボードでカメラを移動させてみましょう。

(３)キーボードでモデルを移動させてみましょう。