

# ESライブラリ&& ゲームプログラミング

## 2D編 - 第4回 キャラクターの移動

### キャラクターの移動

- ・ゲームの仕組みはセルアニメーションと同じ
- ・フレームごとに、キャラクターの表示位置をずらすと、移動しているように見える
- ・固定された数だとキャラクターの表示位置も固定になるので、変数を使って書き換える
- ・キャラクターの画像を変更すると、アニメーションしているように見える

#### 概要

ゲームはセルアニメーションと同じように、1フレームごとに背景やキャラクターを描画しています。このとき、任意のタイミングで表示位置を変更することによって、動いているように見せています。

プログラムによってキャラクターを動かすには、キャラクターの表示位置(座標)を変更しなければなりません。決められた数値(定数)を指定し続けると、同じ位置にしか描画できません。プログラム言語には、変数という仕組みがあります。また、関数には変数を渡すことができます。変数を渡した場合、変数が保持している内容(数値)が読み取られます。状況によって変数の内容を書き換えれば、関数の動作を変えることができます。ESライブラリのSpriteBatch.Draw関数も変数を渡すことができます。表示座標に変数を渡すことにより、フレーム毎に表示位置を自在に変更することができます。これによって、キャラクターが移動しているように見せることができます。

#### 変数

変数とは、プログラムで計算を行うときに使う数値を記憶するために、メモリに数値を格納するための箱のことをいいます。この箱(変数)を介してさまざまな演算を行うことができます。

```
int a = 3, b = 10, c = 3;  
a[3] b[10] c[3]
```

変数に付ける名前の変数名といいます。プログラムは変数名によって変数の区別を行います。上の例では、a, b, cという変数名がつけられ、それぞれ3, 10, 3が入っています。

また、変数には型という属性があります。型によって、記憶できる数値の種類と範囲が異なります。C言語では、基本の型として以下の4つの型があります。上の変数a, b, cはint型なので、約±21億の数値を記憶できます。

型名	説明	Windows上での値の範囲
char	文字を格納	-128 ~ 127
int	符号付き整数	-2147483648 ~ 2147483647
float	浮動小数点	±10の±38乗(有効桁数7桁)
double	倍精度浮動小数点	±10の±308乗(有効桁数15桁)

コンピュータは無限に大きな数字や、無限に大きな精度の数を記憶することができません。上の表のようにそれぞれの型には一定の範囲内の数しか格納することができないようになっています。

## 演算子

C言語の演算子は、加算と減算はそれぞれ+と-を用いますが、乗算はxではなく\*を用います。また、キーボードに÷のキーがないので、除算は/(スラッシュ)で代用します。すなわち、 $a \times b$ は $a * b$ と記述し、 $a \div b$ は $a / b$ と記述します。括弧があれば、その中が先に計算されます。代表的な演算子を紹介します。

算術演算子	Cでの記号	意味	備考
+	+	加算	
-	-	減算	
x	*	乗算	
÷	/	除算	
余り	%	余り	整数のみ使用可

## 条件分岐

C / C++は、上から順番にプログラムが実行されますが、プログラムの流れを何らかの条件によって分岐させることができます。分岐は、if文やswitch文で行います。

if文は、

もし~ならば「処理」を実行し、  
そうでなければ(「処理」を実行しないで)次の処理を実行する

という分岐をさせることができます。  
if文の書式は以下ようになります。

```
if(条件) {  
    条件を満たしたときに実行する処理  
}
```

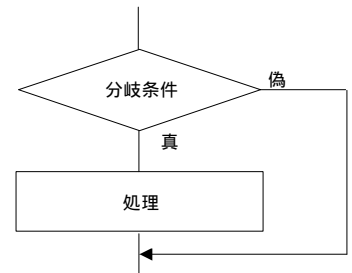
「条件」にはどのような条件でも記述できるわけではなく、基本的に2つの値を比較することしかできません。比較に使用できる演算子も決められており、以下のものしか使用できません。

==    !=    <    >    <=    >=

「==」は等しいかどうかを調べます。「==」の左側と右側が等しいときは真(成り立つ)、等しくないときは偽(成り立たない)となります。C / C++では、「=」が代入に割り当てられているので、比較は「==」を用いるようになっています。

「!=」は「==」の逆で、「!=」の左側と右側が異なるときは真(成り立つ)、等しいときは偽(成り立たない)となります。キーボードに `<=` がいないため、このような記述になっています。

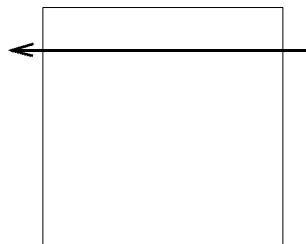
「<」「>」は大小比較に用います。「<=」「>=」は以上と以下(キーボードに `<=` と `>=` がいないためにこのように記述します)を調べる場合に用います。



## 課題

プログラムによってキャラクターを移動させましょう。

(1)キャラクターが、以下のようなルートを移動するプログラムを作成しましょう。



- 1 . 背景とキャラクターの画像を管理する変数をヘッダーファイル"GameMain.h"に宣言します。

```
SPRITE  BG;          // 背景
SPRITE  Chara;       // キャラクター
```

- 2 . LoadContent関数で画像を読み込みます。

```
// 画像読み込み
BG      = GraphicsDevice.CreateSpriteFromFile( TEXT("画像ファイル名") );
Chara   = GraphicsDevice.CreateSpriteFromFile( TEXT("画像ファイル名"), Color(0, 0, 0) );
```

- 3 . Draw関数で描画します。

```
void CGameMain::Draw()
{
    GraphicsDevice.Clear(Color_CornflowerBlue);

    // TODO: Add your drawing code here
    GraphicsDevice.BeginScene();

    // スプライト描画
    SpriteBatch.Begin();

    SpriteBatch.Draw(BG, Vector3(0.0f, 0.0f, 1.0f)); // 背景
    SpriteBatch.Draw(Chara, Vector3(0.0f, 0.0f, 0.0f)); // キャラクター

    SpriteBatch.End();

    GraphicsDevice.EndScene();
}
```

- 4 . キャラクターの座標を管理する変数を宣言します。

このままでは、キャラクターを移動させることはできません。なぜかという、キャラクターの描画座標(3 . の下線部)が0で固定されているからです。ここを増減させることにより、描画座標を自由に変更することができます。しかし、定数の場合は変更することができません。そこで、ここを変数にして自由に書き換えられるようにします。

キャラクターの座標を管理する以下の変数をヘッダーファイル"GameMain.h"で宣言しましょう。

```
float  charaX, charaY; // キャラクター座標
```

- 5 . 変数を初期化します。以下のプログラムをInitialize関数の適切な場所に追加しましょう。

```
charaX = 1280.0f;
charaY = 310.3f;
```

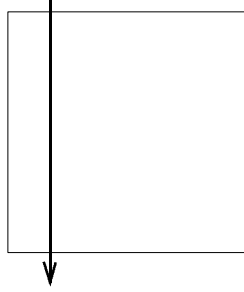
- 6 . キャラクターを描画する部分を以下のように、変数を用いたものに変更しましょう。

```
// キャラクターを変数charaX, charaYの座標に描画する
SpriteBatch.Draw(CharaSprite, Vector3(charaX, charaY, 0.0f));
```

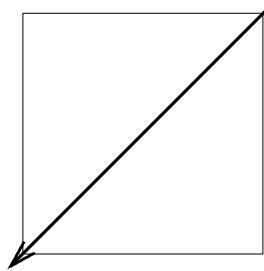
- 7 . 変数を増減させ、キャラクターを移動させます。この処理は、内部処理Update関数で行います。以下のプログラムをUpdate関数に追加しましょう。

```
charaX -= 1.0f;
```

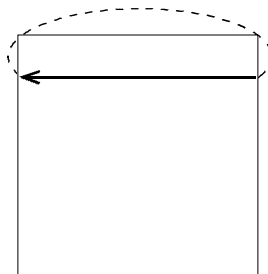
(2)キャラクターが、以下のようなルートを移動するプログラムを作成しましょう。



(3)キャラクターが、以下のようなルートを移動するプログラムを作成しましょう。

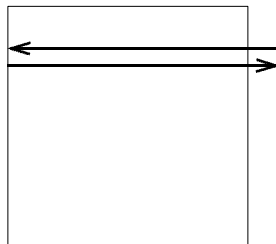


(4)キャラクターが、以下のようなルートを移動するプログラムを作成しましょう。



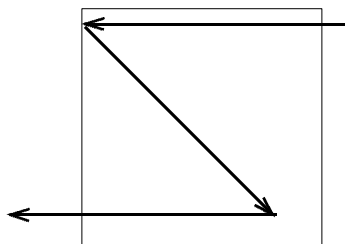
左端まで到達したら、  
右端に戻る

(5)キャラクターが、以下のようなルートを移動するプログラムを作成しましょう。

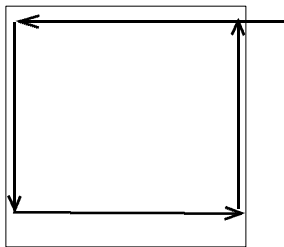


左端まで到達したら右端へ向かう

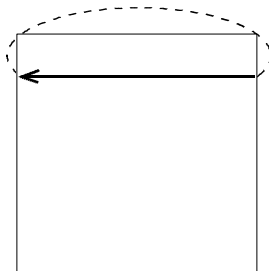
(6)キャラクターが、以下のようなルートを移動するプログラムを作成しましょう。



(7)キャラクターが、以下のようなルートを移動するプログラムを作成しましょう。



(8)キャラクターが、以下のようなルートを移動するプログラムを作成しましょう。



左端まで到達したら、  
右端に戻る。  
ただし、左端で消えた部分が  
右端から出てくる

応用問題 複数のキャラクターを移動させてみましょう。また、放物線や円運動を行うように移動させましょう。