

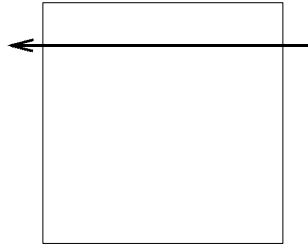
# BREW & EZアプリ ゲームプログラミング

## 第9回 キャラクターの移動

### 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. キャラクターの座標を管理する変数を宣言します。

キャラクターの座標は、内部処理(プロジェクト名\_Proc関数)と描画処理(プロジェクト名\_Draw関数)など、いくつかの関数で参照しなければなりません。

BREWでは、グローバル変数とstatic変数を使用できません。これは、必要に応じて動的にダウンロードされるアプリケーションがそれらの変数进行处理できないという規格上の制限のためです。また、イベントドリブンが採用されているため、イベントハンドラを中心に処理が進められます。しかし、イベントハンドラに対し、自由にデータを渡すことはできません。グローバル変数を使用できず、関数にも自由に引数を渡せない、ということになります。

ここで、イベントハンドラの引数に注目してみると、アプレット構造体が最初にあります。アプレット構造体は、アプリケーションと同じ名前を持つ構造体で、BREWアプリケーションが使用する変数をまとめた構造体です。あらかじめIDisplayインタフェースやIShellインタフェースなど、さまざまな関数で参照しなければならない変数が定義されています。この構造体は、プログラマが自由にメンバを追加することができます。

BREWでは、上記の制限がある以上、グローバルで参照する変数をアプレット構造体のメンバに定義するしかありません。これまでの課題で作成した関数の引数が、アプレット構造体なのもそのためです。よって、ゲームで必要となる変数も、アプレット構造体に宣言します。

キャラクターの座標を管理する以下の変数をアプレット構造体の中に宣言しましょう。

```
int chX; // キャラクターx座標
int chY; // キャラクターy座標
```

2. 変数を初期化します。以下のプログラムをプロジェクト名\_InitGameData関数の適切な場所に追加しましょう。

```
// キャラクター初期化
pMe->chX = 240;
pMe->chY = 10;
```

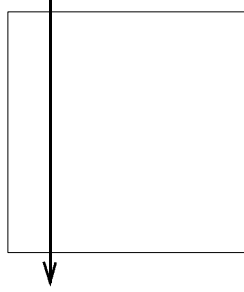
3. キャラクターを描画する部分を、以下のように変数を用いたものに変更しましょう。

```
// キャラクター描画(変数chX, chYの座標に描画する)
IDISPLAY_BitBit(pMe->pIDisplay, pMe->chX, pMe->chY, 60, 60,
pMe->pBmpChara, 0, 0, AEE_RO_TRANSPARENT);
```

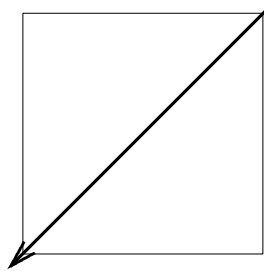
4. 変数を増減させ、キャラクターを移動させます。この処理は、内部処理で行います。以下のプログラムを適切な場所に追加しましょう。

```
// キャラクター移動
pMe->chX--;
```

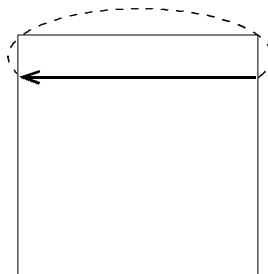
(2)キャラクターが、以下のようなルートを移動するプログラムを作成しましょう。



(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。

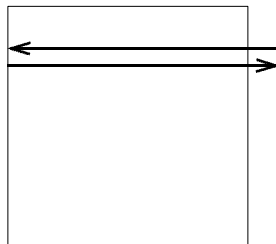


(4)キャラクターが、以下のようなルートを移動するプログラムを作成しましょう。



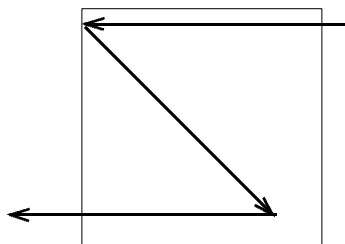
左端まで到達したら、  
右端に戻る

(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。

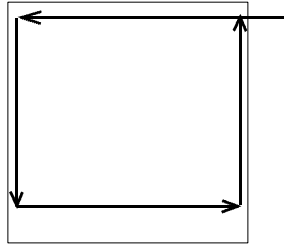


左端まで到達したら右端へ向かう

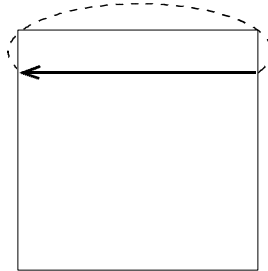
(6)キャラクターが、以下のようなルートを移動するプログラムを作成しましょう。



(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。

#### マクロの定義

アプレット構造体のメンバは、「pMe->」で参照しなければなりません。これが面倒な場合は、以下のようなマクロを定義しておく、簡単に参照できるようになります。

```
// マクロ定義
#define IDISPLAY pMe->pIDisplay
#define ISHELL pMe->pIShell
#define IGRAPHICS pMe->pIGraphics

#define CHARA_X pMe->chX
#define CHARA_Y pMe->chY
```