

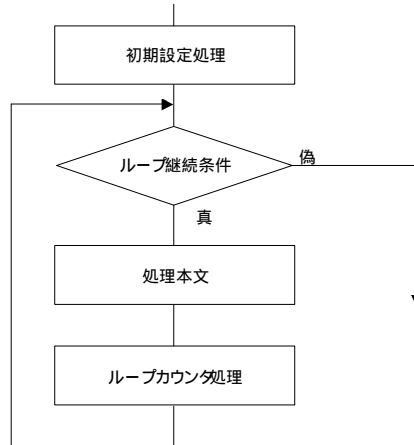
# ESライブラリ&& ゲームプログラミング

## 荷物勇者編 - 第4回 壁の描画

### 反復

反復は、プログラムの一部分を繰り返し実行することをいいます。1回ごとに反復条件を調べ、条件が成立している間は繰り返しを続けます。

反復には、while文、for文、do-while文があり、forやwhileがよく使われます。たとえば、for文は以下のように処理され、条件が成り立つ間、反復させることができます。



for文は、おもに回数を指定したループを行うときに使用します。回数を管理するために、ループカウンタという変数を使用します。for文の書式は以下ようになります。

```
for(初期設定処理; ループ継続条件; ループカウンタ処理) {  
    ...  
    処理本文  
    ...  
}
```

「初期設定処理」は、ループに入る前に一度だけ必ず実行されます。通常は、ループカウンタの宣言と初期化を行います。","で区切ることにより、複数の変数を宣言し、初期化することができます。

「ループ継続条件」は、ループを継続させるための条件を記述します。ループを終了させるための条件ではないことに注意してください。条件が真の間、ループが継続されます。この条件は、初期設定処理の後に判定が行われるので、条件によってはループに入らない場合があります。

「ループカウンタ処理」は、処理本文の後に実行され、ループカウンタの増減を行う処理を記述します。","(カンマ)で区切ることにより、複数のカウンタを処理することができます。

「初期設定処理」「ループ継続条件」「ループカウンタ処理」は、必要のないものは省略することができます。3つの処理すべてを省略してしまうこともできます。

以下のようにfor文を記述すると、iが0から9までの10回、処理本文が実行されます。

```
for(int i = 0; i < 10; i++) {  
    処理本文  
}
```

## 課 題

ステージの周りを壁で覆いましょう。ステージの大きさは  $10 \times 10$  とします。

(1) 壁の画像は  $32 \times 32$  ピクセルとします。以下のような画像を用意しましょう。  
(1パターンでかまいません)



(2) 壁のスプライトを管理するための変数の宣言をします。以下のプログラムをヘッダーファイル "GameMain.h" の "private:" の下に追加しましょう。

```
SPRITE wallSpr;
```

(3) 画像を読み込み、スプライトとして扱えるようにします。以下のプログラムをソースファイル "GameMain.cpp" の "LoadContent関数" の適切な場所に追加しましょう。

```
// 壁画像読み込み  
wallSpr = GraphicsDevice.CreateSpriteFromFile( TEXT("壁画像のファイル名") );
```

(4) 壁スプライトを座標(0.0, 0.0, 0.1)に表示します。以下のプログラムを完成させ、ソースファイル "GameMain.cpp" の "Draw関数" の適切な場所に追加しましょう。

```
SpriteBatch.Draw(*wallSpr, Vector3( ここは各自考えてください ));
```

(5) 壁スプライトをさらにもう1つ表示し、合計2つ表示します。座標(32.0, 0.0, 0.1)にも表示されるように、プログラムを作成しましょう。

(6) 壁スプライトが横一列に10個並ぶプログラムを作成しましょう。

(7) 壁スプライトをfor文を使った繰り返しで表示してみましょう。

壁スプライトの表示プログラムは以下のようにになっています。

```
SpriteBatch.Draw(*wallSpr, Vector3( 0.0f, 0.0f, 0.1f)); // 1個目  
SpriteBatch.Draw(*wallSpr, Vector3( 32.0f, 0.0f, 0.1f)); // 2個目  
SpriteBatch.Draw(*wallSpr, Vector3( 64.0f, 0.0f, 0.1f)); // 3個目  
...  
SpriteBatch.Draw(*wallSpr, Vector3(256.0f, 0.0f, 0.1f)); // 9個目  
SpriteBatch.Draw(*wallSpr, Vector3(288.0f, 0.0f, 0.1f)); // 10個目
```

このプログラムは、「x座標の数値」(下線部)以外の部分はまったく同じです。また、「x座標の数値」には、「32.0ずつ増えている」という法則性があります。このような場合は、for文などの繰り返しでまとめることができます。

「壁スプライトの表示」では、最初のx座標が0.0、最後は288.0、32.0ずつ増えるので、以下のように記述できます。

```
for(float x = 0.0f; x <= 288.0f; x += 32.0f) {  
    SpriteBatch.Draw( ここは各自考えましょう );  
}
```

xが0.0から始まり、288.0f以下の間、SpriteBatch.Draw関数が実行されます。ループが繰り返されるたび、xは32.0ずつ増えていくので、ちょうど1キャラクター分横にずれた位置に壁を表示できます。

「 ここは各自考えましょう 」を埋め、プログラムを完成させましょう。

(8)縦方向にも壁を表示します。(0.0, 32.0)から(0.0, 256.0)まで1キャラクター間隔で縦に壁が表示されるプログラムを作成しましょう。

ヒント:

```
for(float y = 32.0f; y <= 256.0f; y += 32.0f) {  
    SpriteBatch.Draw(*wallSpr, Vector3(0.0f,   ここは各自考えましょう   , 0.1f));  
}
```

(9)下と右側にも壁が表示されるプログラムを作成しましょう。

ヒント: 下は(0.0, 288.0)から、右は(288.0, 32.0)からスタートします

(10)上の壁を描画するfor文を書き換え、以下のようなループにしてみましょう。

```
for(int i = 0; i < 10; i++) {  
    SpriteBatch.Draw(*wallSpr, Vector3(   ここは各自考えましょう   , 0.0f, 0.1f));  
}
```

iが0から始まり、10になったら終了、つまり10回繰り返されるというfor文です。なれてくると、一目で10回繰り返される = 壁が10個表示されるプログラムであることがわかるようになります。

(7)のfor文は、「0.0から288.0まで、32.0間隔で壁を表示」ということが読み取れます。(7)は壁の描画範囲、(10)は壁の個数がわかりやすいという利点があります。好きな方を使ってかまわないのですが、どちらかという(10)が使われています。

「 ここは各自考えましょう 」を埋め、プログラムを完成させましょう。

ヒント: 0.0 0 \* 32.0  
 32.0 1 \* 32.0  
 64.0 2 \* 32.0  
 96.0 3 \* 32.0

応用問題1: ループを2つに減らしましょう

上下左右ごとに4つのループがあります。これを上下と左右の2つに減らすことができます。(for文の中は2行になります)

応用問題2: ループを1つにしてみましょう。

描画範囲が10×10と正方形、1キャラクターも32×32の正方形なので、角の壁が2重に描画されていますが、ループを1つにすることもできます(for文の中は4行になります)。