



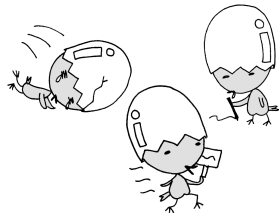
Visual Basic .NET  
TAMAHIO CLUB  
作って学ぶ  
VISUAL BASIC .NET

瀬戸 遥

SETO, Haruka

<http://www.big.or.jp/~seto/>

<http://hp.vector.co.jp/authors/VA006682/>



## お手軽シューティングゲームを作る-その2-

### Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:

### Level



### Samples

・この記事で取り上げたソースコードおよびサンプルプログラムは、付録CD-ROMの¥DOTNET¥TAMAHIOディレクトリに収録しています。

¥SHOOT2  
今回のサンプル



### ゲームを仕上げる

今回は、ゲームを開始する [開始] ボタンのコードまで作成しました。今回は的と弾の各アイコンのアニメーションといった、残りの処理を組み込みます。



### 的アイコンを動かす

[開始] ボタンが押されると、的アイコンが動き始めます。まずは、この的アイコンのアニメーション処理について解説しましょう。

的アイコンのアニメーションは、Timerコントロール「Timer1」のTickイベントプロシージャに記述します。フォームデザイナーで、Timer1をダブルクリックすると、コードエディタにTickイベントプロシージャが作成されます。

```
Private Sub Timer1_Tick(略) Handles _  
Timer1.Tick
```

```
End Sub
```

TimerコントロールのTickイベントは、Intervalプロパティに設定した時間間隔で発生し続けます。Timer1はIntervalプロパティを「20」に設定したので、20ミリ秒(0.02秒)ごとにTickイベントが発生し、Tickイベントプロシージャが実行されます。

そこで、PictureBoxコントロールのLocationプロパティの値を「2」ずつ減らしながら設定してゆけば、的アイコンが左方向に少しずつズレてゆくこととなります。これを20ミリ秒ごとに行なえば、ユーザーの目にはアイコンが左に移動しているかのように見えます(図1)。

### ●アニメーション処理

ではさっそく、的アイコンのアニメーション処理を組み込んでみま

しょう。

まず、変数xの値から「2」を減らします。変数xは、プログラムの初期化処理のところで、モジュールレベル変数として宣言したときに、値「680」を代入しました。ここから「2」だけ減らします。

この計算式には、**-=演算子**を使用します。**-=演算子**は、演算子の左辺から、右辺の値を差し引きます。したがって、変数xから2を減らすには、

```
x -= 2
```

と記述します。

そして、この値を使ってPoint構造体を作成し、PictureBoxコントロールのLocationプロパティに代入します。

```
Me.PictureBox1.Location = New Point(x, 10)
```

これで、的アイコンを表示するPictureBoxコントロール「PictureBox1」は、ほんの少し左にズレます。

そして、この計算は、0.02秒後に再度このイベントプロシージャが呼び出されたときにもう一度行なわれます。こうして、Tickイベントが発生するたびに、変数xの値は2ずつ減り、その値がLocationプロパティに設定されるので、PictureBox1は20ミリ秒ごとに左に移動してゆきます。

ところで、的アイコンがフォームの一番左にいつてしまったときはどうしたらよいでしょうか。その場合は、もう一度右からの的アイコンを表示しなければなりません。そこで、この処理を組み込んでみましょう。

まず、変数xの値を調べ、Ifステートメントにセットします。変数xの値が「-40」未満になったら、xの値を「680」、すなわちアニメーションのスタート位置に戻します。そして、この値でPoint構造体を作成し、Locationプロパティに代入します。

```
If x < -40 Then
    x = 680
End If
```

図1：的アイコンの動作範囲

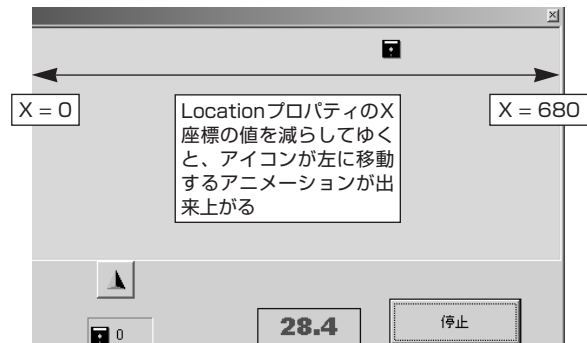
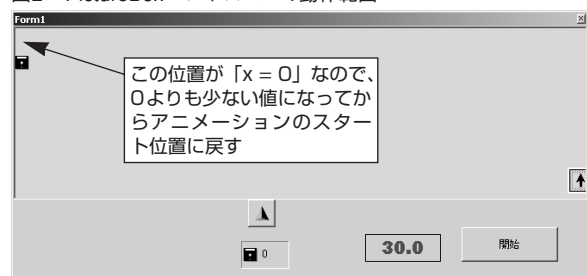


図2：PictureBoxコントロールの動作範囲



リスト1：的アイコンのアニメーション

```
Private Sub Timer1_Tick(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles Timer1.Tick
    x -= 2
    If x < -40 Then
        x = 680
    End If
    Me.PictureBox1.Location = New Point(x, 10)
End Sub
```

これで、「アイコンがフォームの一番左に消えてしまったら、右から再度表示する」という繰り返し処理が出来上がります。

なぜ、変数xを「0」でなく「-40」未満と比較しているのかというと、PictureBoxコントロールがすべてフォームから見えなくなるのは、「0」からPictureBoxコントロールの幅以上の値を引いた位置になるからです(図2)。

この的アイコンがフォームの一番左にいつてしまったかどうかの判定は、「x -= 2」の式のすぐ後で行ないます。

ここまでのコードをまとめると、リスト1のようになります。