επιστημηの

オブジェクト指向的夜話

★C#から見た
★.NET Framework

第2回

ModelとViewを分離する

επιστημη
えぴすてーめー

Technology Tools

- ❑ Visual Basic .NET
- ☑ Visual C# .NET
- ❑ SQL Server 2000
- ❑ Oracle 9i
- ❑ Access 2002
- ❑ ASP.NET
- ❑ Internet Information Services
- ☑ Other:
  - Visual C++ .NET
  - NUnit v2.1

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¥YAWAディレクトリに収録しています。

¥GATE：サンプルアプリケーション
¥GATE_CONSOLE：コンソール版
¥GATE_GUI：GUI版
¥GATE_GUI_VB：
VB.NETによるGUI版
¥GATE_MODEL：
ModelとViewを分離
¥GATE_MODEL_TEST：
NUnitを使ったModelサンプル

GUIなんてへっちゃら

.NET FrameworkのおかげでGUIアプリケーションの構築がとっても楽になりました。MFC＆C++によるGUIアプリケーション開発環境としてのVisual Studioは「いったいこれのドコがVisualなわけ？」と叫びたくなるほどにアプリケーションの“見てくれ”作りに苦労します。いや、苦労というと語弊がありますか……。面倒なんですよね。テキストボックス貼り付けてそいつのフォントや色を変えたいとなれば、ちまちまとコードを書くしかありませんし、期待通りの見てくれができたのかは実行してみないと画面に現われません。対して.NET Frameworkが提供するFormsクラス群とVisual Studio .NETの連携プレイで見てくれの変更のほとんどがプロパティの設定だけで済んでしまいますからホントに楽です。これはGUIアプリケーションをバリバリ書きまくるプログラマにはもちろんのこと、僕のようなGUIよりは内部のロジックをこねくり回すプログラマにも大きなメリットがあります。ロジックを組み立てる過程ではときとして簡単なUIをこしらえてお試しにちょいちょいと動かしてみたくなることがあります。そんなときにさほどのコードを書かずにサクサクGUIが構築できれば、それだけロジックに専念できるわけですからね。GUIの苦手な僕には誠に有難い。

ゲートのシミュレーション

オブジェクト指向の教材として、少し前に簡単なコードをC++/Javaで書きました。お題は“ゲート”です。近所の大きな病院の駐車場にあるもので、見舞い客の車がこのゲートを通過します。病院の受付の横にコインの自動販売機があっ

図1：状態と事象と動作の関係

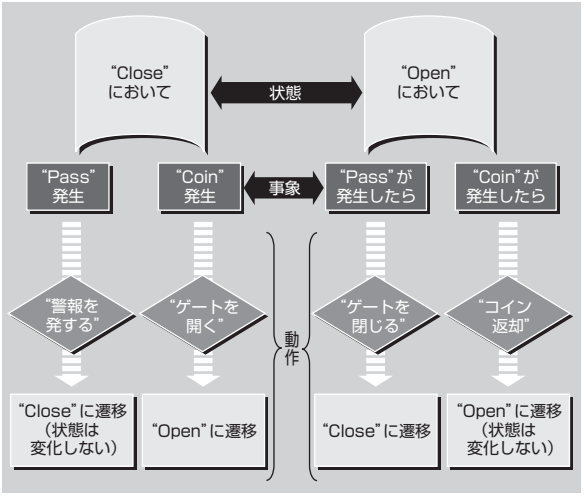
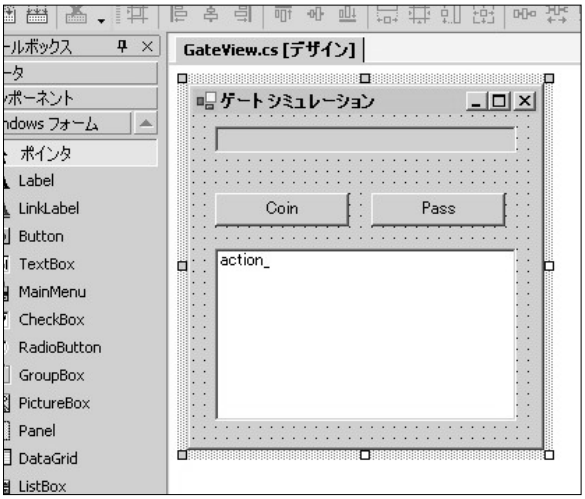


図2：“ゲート”のインターフェイス



て、見舞い客はこのコインを買い、駐車場を出るときにこのコインをスロットに投入すればゲートが開き、車が通過するとゲートが閉じるからくりになっています（実際にはもっと複雑なのでしょうが）。ゲートには2つの状態（state：ステート）、

・Open：開いている
・Close：閉じている

があり、ゲートに対して発生する事象（event：イベント）は、

・Coin：コインが投入される
・Pass：車が通過する

があります。ゲートの動きはこの状態と事象の組み合わせで表現できます。ある状態（s）において事象（e）が発生したとき、状態と事象の組み合わせに対応した動作（action）を行なって新たな状態に遷移します（図1）。

これを素直にコードに落とせば二重のswitch文になるでしょう。状態／事象の組み合わせで定まる（2×2＝4種の）挙動では動作と次の状態を定めることになります。実装してみましょうか。Visual Studio .NETを立ち上げ、C#でWindowsアプリケーションプロジェクト「gate」を作成します。生成された雛形フォームに状態を表示するTextBox、挙動を表示するListBox、そしてCoin/Passそれぞれの事象を発生させる2個のButtonを配置しました（図2）。“キモ”を抜粋したコードがリスト1です。メソッド「drive」が二重のswitch文になっているのが見て取れることと思います。

RADの功罪

……あっけないほどに簡単でした（図3）。見てくれができてしまえば、そこにコードを埋め込んでゲートシミュレーションの体裁を整えるのにものの30分もかかっていないんじゃないかしら。この程度ならMFC/C++でもさほどの時間はかからなかったでしょうが、さらに凝った見てくれに仕上げるとなるとコード量が増えてうんざりするのでしょう。さすがはRAD（Rapid Application Development）環境です。手軽にコードを書いて遊べるその手軽さはビギナーには特にウケがいいでしょう。Visual Basicの人気もうなずけます。

図3：“ゲート”実行中

RAD環境の手軽さ、それは誠に結構なことなのですが、同時に僕は