

CHAPTER 14

迷路ゲームに挑戦！

解説するテクニック： 大きなマップを表示する 上下左右がつながったマップを表示する 迷路の作成 軌跡の表示

見本

▶ はじめに

本章では「迷路」ゲームを製作します。まず、どのような「迷路」を作っていくかということについてですが、ただ「迷路」を自動作成するだけではありません。画面には「迷路」の一部を表示してプレイヤーの移動に従って画面がスクロールしたり、プレイヤーの移動の軌跡が画面に表示されたりと、内容盛りだくさんのプログラムにしていきます。ただし、ゲームの目的としては、「2, 4, 6, 8」キーを使ってマル君をゴールに導くという、ルールとしては簡単なものです。

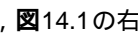
今回のゲームで使用するテクニックは、

- (1) 大きなマップを表示するテクニック
- (2) 上下左右がつながったマップを表示するテクニック
- (3) 迷路を作成するテクニック
- (4) 軌跡を表示するテクニック

です。

最初の大きなマップを表示するテクニックは、画面より大きなマップを作成し、画面をスクロールさせながら表示するテクニックです。一般的に、マップの上下と左右がつながっていないバージョンはRPGの町の場面処理で使用され、マップの上下左右がつながったバージョンはRPGのフィールドの処理に使用されています。

迷路を作成するテクニックはそのものです。

軌跡を表示するテクニックは、14.1の右上に表示されているように、マップ上の軌跡を表示するテクニックです。さまざまなゲームで見かけることができるこのテクニックは、レーダーを使用するゲームにも応用が可能です。

○ 図14.1 本章で製作する「迷路」ゲームの画面



▶ 大きなマップを表示するテクニック

それでは、最初のテクニックから考えていきましょう。画面に収まりきらない「大きな背景を表示するテクニック」として、マップからはみ出した部分が黒く表示されるバージョンを作成していきます。背景は、縦横の大きさも含めてテキストファイルで指定するので、応用範囲の広い方法です。

通常、背景はある大きさ、例えば32ドット×32ドットのパーツを組み合わせて作ります。もし、背景が256マス×256マスで構成されていたら、これは8192ドット×8192ドットの大きさになります。これを24ビットカラーのビットマップで保存しようとする、1ドットにつき3バイト必要なので192Mバイトのビットマップが必要になります。ここでは、背景をすべてビットマップ化せず、表示領域に関係したマスのみをビットマップに変換することでメモリの使用量を抑えています。

それでは、「大きなマップを表示するテクニック」を考えるために、「大きなマップの表示1」というプログラムを作成しましょう。デザイン画面を図14.2に示します。

まず、マップを表示するフィールドにあたる部分に、PictureBoxコントロール (pictureBox1) を配置します。次に、操作の説明文としてLabelコントロール「2,4,6,8キーで移動します」(label1) を配置します。次に、マップのビットマップを格納するImageListコントロール (imageList1) を配置し、ImageSizeプロパティを「32, 32」に設定します。

imageList1には、図14.3に示した背景を構成するパーツを設定します。図14.3の最後に黒い四角を設定していますが、これはマップからはみ出したときに表示するところのパーツです。そして、Timerコントロール (timer1) を配置し、Enabledプロパティを「True」に、Intervalプロパティを「50 (0.05秒)」に設定します。