

# オブジェクト指向と ゲームプログラミング

## § アルゴリズム編 - 第5回 仮想画面による通路移動 §

### 通路の移動

2次元のゲームでは、平面のフィールドの中で、上下左右にキャラクタを操作して移動させます。このタイプのゲームには、

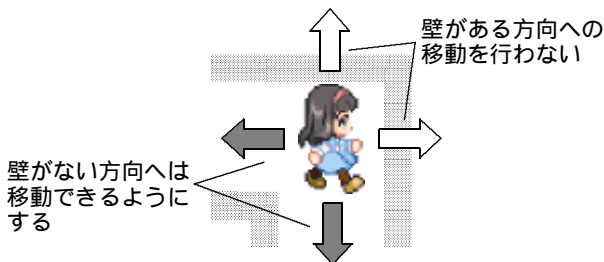
通路が存在して、移動がかなり限定されているもの  
障害物はとくに設定せず、自由に移動できるもの

という2種類があります。今回は、このうち「通路が存在して移動範囲が限定されている」パターンで進めていきます。

このパターンのゲームの代表格といえば、パックマンや平安京エイリアンです。とくに前者は世界的な人気があり、現在でもリメイクされて発売されています。

これらのゲームでは、登場するキャラクタは、壁によって構成された通路上を移動します。このような動作を行わせるには、キャラクタの移動処理をするときに、壁の存在する方向へ移動しないようにする必要があります。

たとえば、以下の図を見てください。このように、キャラクタの上と右に壁が存在する場合、上や右に移動しないように判断する必要があります。



壁の存在の判断方法にはいろいろな方法がありますが、今回は"仮想画面"を用いた方法を使ってみます。この方式は、壁とキャラクタの大きさが同じ場合に有効で、プログラムのにも簡単になります。

### 仮想画面とは

仮想画面による通路の判断では、キャラクタの大きさを1単位とした仮想画面を用意する必要があります。

この仮想画面とは、見た目は複雑なグラフィックになっているキャラクタを単純な数値に置き換えて、そのキャラクタの存在箇所に対応する数値を記録したものです。こうすることで、場所を指定するとそこにどのキャラクタが存在するか、即座に判断できるようになります。

サンプルとして右の図を見てください。このように、登場するキャラクタが

- ・プレイヤーが操作するキャラクタ
- ・敵キャラクター
- ・壁

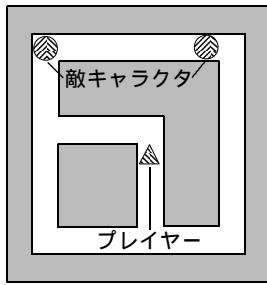
の3種類で、フィールドが $10 \times 10$ の大きさのゲームを考えてみましょう。この場合、仮想画面として $10 \times 10$ のサイズの記録箇所を用意します。そして、画面上ではグラフィックによって表示されているキャラクタを数字に置き換え、そのキャラクタが存在する箇所に数値を記録します。

右の図のような仮想画面を見るとわかるように、

- ・プレイヤーキャラクタが存在する箇所に「1」
- ・敵キャラクタが存在する箇所に「2」
- ・壁には「3」
- ・何も無い場所には「0」

が記録されるわけです。

このような仮想画面を使うことによるメリットはいくつかあります。まず、通路の判断が即座にできること。そして、キャラクター同士の衝突判断に使いやすいことなどです。



壁があるフィールド上のゲーム

3	3	3	3	3	3	3	3	3	3
3	2	0	0	0	0	2	0	3	3
3	0	3	3	3	3	3	0	3	3
3	0	3	3	3	3	3	0	3	3
3	0	0	0	0	0	3	3	0	3
3	0	3	3	3	1	3	3	0	3
3	0	3	3	3	0	3	3	0	3
3	0	3	3	3	0	3	3	0	3
3	0	0	0	0	0	0	0	0	3
3	3	3	3	3	3	3	3	3	3

左の状態を仮想画面に投影する

## "2次元配列"で仮想画面を実現

仮想画面をプログラムで実現する場合は、"配列"を用います。プログラム中での仮想画面は、座標から存在キャラクターを引き出す、データベースのような働きをします。配列の添字を座標に対応させ、対応する添字の要素にキャラクターの番号を記録しておけば、仮想画面と同じ動作ができるのです。

配列の添字と座標の対応については、配列を"2次元配列"にすることによって、x軸とy軸を直接対応付けることができます。

具体的には、2次元配列の1つ目の添字をy座標、2つ目の添字をx座標に対応させてやれば、

```
array[y][x]      arrayは10×10の2次元配列
```

で座標(x, y)の仮想画面にアクセスできるわけです。

たとえば、上の図の例で、座標(7, 1)の仮想画面に敵キャラクターの数値を記録させるには、

```
array[1][7] = 2;
```

とします。また、座標(5, 3)の仮想画面に、壁が存在するかどうかを調べるには、

```
if(array[3][5] == 3)
```

とすればいいことになります。

## 仮想画面のメリット

このような仮想画面を利用すると、壁の判定、衝突検出などに関係する対応キャラクターの選択などの処理が、大幅に簡略化および高速化できます。

壁の判定については、キャラクターの進行方向の仮想画面を調べることによって、壁があるかどうかですぐにわかります。壁であれば移動方向を変える処理を行えば、壁によって仕切られた通路をキャラクターが移動していくようになります。

移動方向を変えるときも、新しい移動先に壁があるかどうかを仮想画面によって調べ、移動できる方向を新しく設定してやればよいのです。

## 仮想画面のメリット

仮想画面に記録する数値を、より複雑なものにすることによって、配列で管理されている複数のキャラクターの添字を直接求めることもできます。

前述の例では、プレイヤーキャラクターが「1」、敵キャラクター「2」、壁は「3」というぐあいに数値を付けました。敵キャラクターが複数いる場合、仮想画面からの情報が「2」であったとすると、敵キャラクターが存在することはわかるものの、どの敵キャラクターかはわかりません。

複数敵キャラクターが存在する場合、それらは配列で管理されていることが多いはずですが、このような場合、敵キャラクターの数値を変更し、「10」や「0xF0」などの計算しやすい数値に、敵キャラクターのデータを格納している配列の添字を加えたものを仮想画面に記録します。すると、仮想画面からのデータが敵キャラクターを示すものであった場合、たとえば13や0xF3とすると、「配列の4つ目に格納されている敵キャラクターが存在する」ということが即座にわかります。