世界はオブジェクトの海に浮かぶ

.NET Frameworkで楽しむオブジェクト指向

第6回 たまにはお遊びプログラミング －その2－

επιστημη
えぴすてーめー

Level

1 2 3 4 5

Technology Tools

- ❑ Visual Basic
- ☑ Visual C#
- ❑ Visual C++
- ❑ SQL Server
- ❑ Oracle
- ❑ Access
- ❑ ASP.NET
- ❑ Other:

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、http://www.shoeisha.com/mag/windev/からダウンロード可能です。

マインスイーパ書き直し

前回、思いつくままゴリゴリとMineSweeperの殴り書きを試みました。MineSweeperのキモは抑えることができたように思いますが、出来映えとしてはまだまだ“デキソコナイ”あるいは“作りかけ”です。せめて小中学校のパソコンクラブ員が秋の文化祭に出品する程度にゲームとしての体裁を整えておかないと、どう見ても“これで完成”と言うには程遠い。前回こしらえたコードを見直し、あちこちに手を加えて改造と改良を試みます。

ゲーム開始

前回作りかけのMineSweeperはアプリケーションの起動と同時に地雷原の初期化（地雷の埋設）を行なっていますが、この初期化部には二度と処理が戻ってくることはなく、ゲームのリプレイができません。再度ゲームを楽しむには一旦アプリケーションを終了し再起動することになります。これではあまりにお粗末ですから、［GO!］ボタンをひとつ（ついでに残り地雷数と経過秒数表示も）配置し、そいつがクリックされたときに初期化を行なうようにします（図1）。

それに併せてタイマー（経過秒数）の表示。MineSweeperは地雷原に埋まったすべての地雷を除去する早さを競うゲームですからね。フォーム（MineForm）にタイマーコントロールを置き、一定時間ごとに発せられるタイムアウトイベントを受理して経過秒数を表示します。タイマーの起動は区画のどれかを最初にクリックした瞬間としましょう。

［GO!］ボタンのクリック後、“最初の区画クリック”を検出するためフラグ「firstSweep_」をひとつ用

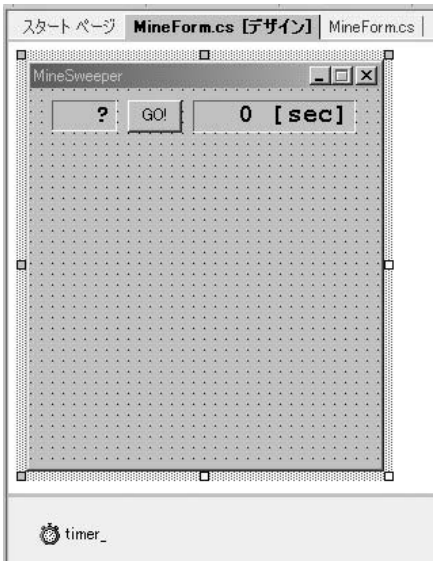
意しました。タイマーコントロールがイベントを発するインターバルをぴったり1秒（＝1000ms）に設定し、タイムアウトイベントでカウンタを“＋1”して表示するのが楽なんでしょうけど、お勉強ついでに違うやり方で。System.Environment.TickCountプロパティはシステム起動からの経過時間をミリ秒単位で返します。なのでゲーム開始時にこの値を保持しておき、タイムアウトイベント発生の瞬間に得たSystem.Environment.TickCountから差し引けば、タイマーインターバルをどんな値に設定しようがゲーム開始からの正しい経過時間が得られるというわけ。

```
class MineForm : Form {
   ・
   ・
  private int tickBias_;
  private bool firstSweep_;

  /// 最初のボタンクリック時の処理
  private void firstSweep(Point point) {
    if ( firstSweep_ ) {
      field_.arrangeMines(point);
      timer_.Enabled = true;  // タイマー開始
      tickBias_ = System.Environment.TickCount;
      firstSweep_ = false;
    }
  }

  /// ボタンクリック
  private void button_Click(object sender,
                            System.EventArgs e) {
    Button button = (Button)sender;
    Point point = (Point)button.Tag;
    firstSweep(point);
    field_.sweep(point);
  }
   ・
   ・
```

もうひとつは地雷の埋設。ゲーム開始時に定められた数の地雷を区画内に（ランダムに）埋めてますが、運が悪いプレイヤーは最初のクリックでいきなり地雷を踏んづけてゲームオーバーとなってしまいます。これではさすがのプレイヤーも意気消沈、戦意喪失。

地雷の埋設は最初のクリックまで保留し、最初のクリック時にその区画を避けて埋設することにしました。あるいは従来通り埋設を済ませておき、最初のクリックで

図1：UIをちょっと改良



いきなり地雷だったときは地雷の置かれていない区画に移設してもよいでしょう。

```
class MineField {
   ・
   ・
  private int mines_;

  /// 地雷数をnに設定
  public void setMineNumber(int n) {
    mines_ = n;
  }

  /// n個の地雷を埋める。
  /// ただしpointは除外する
  public void arrangeMines(Point point) {
    foreach ( MineBlock block in block_ ) {
      block.mine = false;
    }
    Random random = new Random();
    for ( int i = 0; i < mines_; ++i ) {
      int x;
      int y;
      do {
        x = random.Next(xSize_);
        y = random.Next(ySize_);
        // すでに地雷を埋めた位置、
        // およびpointを避ける
      } while ( block_[x,y].mine ||
                (x == point.X && y == point.Y) );
      block_[x,y].mine = true;
    }
  }
```