

16 地雷ゲーム

Windows についているゲーム mine sweeper (地雷掃滅隊) を作ります。

16.1 ゲーム解説

地雷原 (地雷が埋設されている野原) は, $m \times n$ 個の区画 (小さい正方形) からなる長方形で, いくつかの区画に地雷が埋められています。その他の区画には, その区画のまわりの 8 つの区画のうち何区画に地雷を埋めたか, 地雷が埋められている区画の数が書かれたプレートが埋められています。

地雷が埋められていない (プレートが埋められている) 区画は特別の処置をしないで掘り開けることができますが, 地雷が埋められている区画は爆発しないように特別の処置をして掘り開けます。

ゲームでは区画を左クリックすることが, 特別の処置をしないで (すなわち地雷がないと推理して) 掘り開けることを意味します。このとき, その区画に地雷があると爆発してゲームは終了してしまいます。推理どおり地雷がなければプレートが現れ, まわりに地雷がいくつあるかヒントが得られます。

右クリックは, 特別の処置をして (すなわち地雷があると推理して) 掘り開けることを意味します。このとき, 地雷があってもなくても掘り開けた跡だけが残ります。プレートがあった区画でも処理した跡だけが残るので, ヒントが得られないのはもちろん, まるで地雷があったかのようにしか見えません。

しかも安全に処理するための器材はちょうど地雷の個数だけしかないので, プレートの区画を誤って処理したときは, 残っているすべての地雷を処理することはできなくなります。

ただし, 地雷が埋められていない区画を全部掘り開ければゲームは成功です。

なお, 中央の区画には地雷は埋められていないので, この区画から掘り始めて行きます。

16.2 新規プログラム保存

フォルダー名	Jirai
ユニット名	JiraiU.pas
プロジェクト名	JiraiP.dpr

Form のプロパティ

Name	FormJirai
Caption	地雷ゲーム (学生証番号 氏名)
Position	poDeskTopCenter

メインパネルと終了ボタン

いつものように。

16.3 地雷原

地雷原は大, 中, 小 3 サイズを選べるようにします。

16.3.1 サイズ選択ラジオグループ

RadioGroup をメインパネルの中に置く。

```

Align      alTop
Name       RadioGroupSize
Caption    サイズ
Columns    3
Items      大
           中
           小
ItemIndex  0

```

16.3.2 地雷原イメージ

Image をメインパネルの外に置く。

```

Name       ImageHara
Hint       左クリック = ないだろう 右クリック = あるだろう
ShowHint   True

```

16.3.3 データ構造

```

const
  TateDai   = 21;           // 大場面の縦幅
  YokoDai   = 29;           // 大場面の横幅
  KukakuSize = 20;         // 区画の幅

type
  TJoutai = (kjMada, kjAruAtari, kjAruHazure, kjNaiAtari, kjNaiHazure);
  TKukaku = record
    Aru      : Boolean;     // 地雷がある
    Hint     : Byte;       // まわりの地雷の数
    Joutai   : TJoutai;    // 状態
  end;
  TBanchi = 0..YokoDai+1;
  THara   = array [0..TateDai+1, 0..YokoDai+1] of TKukaku;

TFormJirai = class(TForm)
  PanelMain: TPanel;
  ButtonClose: TButton;
  RadioGroupSize: TRadioGroup;
  ImageHara: TImage;
  procedure ButtonCloseClick(Sender: TObject);
private
  Hara      : THara;       // 地雷原
  GyouSuu  : TBanchi;    // 行数
  RetuSuu  : TBanchi;    // 列数

```

16.3.4 メソッド

```

(***** FormJirai の method *****)
procedure TFormJirai.KukakuWoKaku(Gyou,Retu : TBanchi);
  (* 区画を描く *)

  const
    Iro : array [TJoutai] of TColor
      = (clYellow,clLime,clLime,clAqua,clRed);           // 状態を表す色
    Kigou : array [TJoutai] of String[2]
      = (' ',' ',' ',' ',' ',' ');                 // 状態を表す記号
    Suuji : array [0..8] of String[2]
      = (' ',' 1 ',' 2 ',' 3 ',' 4 ',' 5 ',' 6 ',' 7 ',' 8 '); // ヒントの数字

  var
    X,Y : Word;
  begin
    Y := Gyou*KukakuSize;
    X := Retu*KukakuSize;
    with ImageHara.Canvas,
      Hara[Gyou,Retu] do
      begin
        Brush.Color := Iro[Joutai];
        if Joutai = kjNaiAtari
          then TextOut(X,Y,Suuji[Hint])
          else TextOut(X,Y,Kigou[Joutai]);
        end;
      end;
  end; (* KukakuWoKaku *)

procedure TFormJirai.HaraWoKaku;
  (* 地雷原を描く *)

  var
    Gyou,Retu : TBanchi;
  begin
    with ImageHara.Canvas do
      begin
        Brush.Color := clTeal;
        FillRect(ClipRect);
        for Gyou := 1 to GyouSuu do
          for Retu := 1 to RetuSuu do
            KukakuWoKaku(Gyou,Retu);
          end;
        end;
      end;
  end; (* HaraWoKaku *)

procedure TFormJirai.HaraWoKimeru(Size : Byte);
  (* Size に合わせて原を決める *)

  var
    Gyou,Retu : TBanchi;
  begin
    GyouSuu := TateDai-Size*6;
    RetuSuu := YokoDai-Size*10;
    with ImageHara do
      begin
        Height := (GyouSuu+2)*KukakuSize;
        Width := (RetuSuu+2)*KukakuSize;
        Left := (Parent.ClientWidth -Width-PanelMain.Width) div 2;
        Top := (Parent.ClientHeight-Height) div 2;
        with Canvas.Font do
          begin
            Name := 'ゴシック';
          end;
        end;
      end;
    end;
  end;

```

```

        Height := KukakuSize;
        Pitch := fpFixed;
    end;
end;
for Gyou := 0 to TateDai+1 do
    for Retu := 0 to YokoDai+1 do
        Hara[Gyou,Retu].Joutai := kjNaiAtari;
    HaraWoKaku;
    end; (* HaraWoKimeru *)

```

16.3.5 イベントハンドラー

```

(***** Event Handler *****)
procedure TFormJirai.ButtonCloseClick(Sender: TObject);
begin
    Close;
end; (* ButtonCloseClick *)

procedure TFormJirai.FormCreate(Sender: TObject);
begin
    ClientHeight := KukakuSize*(TateDai+4);
    ClientWidth := KukakuSize*(YokoDai+4)+PanelMain.Width;
    HaraWoKimeru(0);
end; (* FormCreate *)

procedure TFormJirai.RadioGroupSizeClick(Sender: TObject);
begin
    HaraWoKimeru(RadioGroupSize.ItemIndex);
end; (* RadioGroupSizeClick *)

```

16.3.6 実行

ラジオグループでサイズを変更すると地雷原のサイズが変更されます。
地雷原の外から中にマウスを入れたとき、ヒントが表示されます。

16.4 地雷を埋める

16.4.1 新問題ボタン

Button をメインパネルの中に置く。

Name	ButtonNew
Caption	新問題

16.4.2 変数

地雷の個数などを記憶する変数を Form の private 部に追加する。

```

private
  Hara      : THara;           // 地雷原
  GyouSuu   : TBanchi;       // 行数
  RetuSuu   : TBanchi;       // 列数
  KukakuSuu : Integer;
  JiraiSuu  : Integer;
  AruNokori : Integer;
  NaiNokori : Integer;
  KizaiNokori : Integer;

```

16.4.3 メソッド追加

```

procedure TFormJirai.JiraiWoUmeru;
  (* 地雷を埋める *)

var
  Gyou,Retu : TBanchi;
  GyouMid,RetuMid : TBanchi;
  G,R : TBanchi;
  Kosuu : Integer;
begin
  // 個数
  KukakuSuu := GyouSuu*RetuSuu;
  JiraiSuu := KukakuSuu div 5;
  // 全区画を空にする
  for Gyou := 1 to GyouSuu do
    for Retu := 1 to RetuSuu do
      with Hara[Gyou,Retu] do
        begin
          Aru := False;
          Hint := 0;
          Joutai := kjNaiAtari;
        end;
      // 中央の 3 x 3 区画に地雷を埋めないために仮に「ある」ことにする
      GyouMid := (GyouSuu+1) div 2;
      RetuMid := (RetuSuu+1) div 2;
      for Gyou := GyouMid-1 to GyouMid+1 do
        for Retu := RetuMid-1 to RetuMid+1 do
          Hara[Gyou,Retu].Aru := True;
        // ランダムに地雷を埋める
      for Kosuu := 1 to JiraiSuu do
        begin
          repeat
            Gyou := Random(GyouSuu)+1;
            Retu := Random(RetuSuu)+1;
            until not Hara[Gyou,Retu].Aru; // すでにあったら選びなおし
            with Hara[Gyou,Retu] do
              begin
                Aru := True;
                Joutai := kjAruAtari;
              end;
            for G := Gyou-1 to Gyou+1 do // 周りの区画のヒントを + 1
              for R := Retu-1 to Retu+1 do
                Inc(Hara[G,R].Hint);
            end;
          // 中央の 3 x 3 区画を元に戻す

```

```

    for Gyou := GyouMid-1 to GyouMid+1 do
      for Retu := RetuMid-1 to RetuMid+1 do
        Hara[Gyou,Retu].Aru := False;
      HaraWoKaku;
    end; (* JiraiWoUmeru *)

```

16.4.4 HaraWoKimeru を変更

```
HaraWoKaku;
```

を

```
JiraiWoUmeru;
```

に変更する。

16.4.5 実行

新問題ボタンを押したら、新しく地雷を埋めなおすように、イベントハンドラーを追加してください。

ただし、プログラムを確認するために、ゲームを成功して終了した状態を表示します。地雷を埋めた区画は処理済の穴の跡が出ます。ヒントの数字がまわりの地雷の個数になっていることを確認しなさい。また中央の区画とそのまわりには地雷がないことを確認しなさい。

確認できたら、

```
Joutai := kjNaiAtari;
```

と

```
Joutai := kjAruAtari;
```

を

```
Joutai := kjMada;
```

に書き換えて、ゲーム開始前の状態を表示するようにしなさい。

16.5 ゲーム実行

16.5.1 開始ボタンと中止ボタン

Button をメインパネルの中に置く。

Name	ButtonStart	ButtonGiveUp
Caption	ゲーム開始	ゲーム中止

16.6 イベントハンドラー変更と追加

```
procedure TFormJirai.FormCreate(Sender: TObject);
begin
  ClientHeight := KukakuSize*(TateDai+4);
  ClientWidth := KukakuSize*(YokoDai+4)+PanelMain.Width;
  HaraWoKimeru(0);
  JiraiWoUmeru;
  ImageHara.Enabled := False;
  RadioGroupSize.Enabled := True;
  ButtonNew.Enabled := True;
  ButtonStart.Enabled := True;
  ButtonGiveUp.Enabled := False;
end; (* FormCreate *)

procedure TFormJirai.ButtonStartClick(Sender: TObject);
begin
  StandBy;
  ImageHara.Enabled := True;
  RadioGroupSize.Enabled := False;
  ButtonNew.Enabled := False;
  ButtonStart.Enabled := False;
  ButtonGiveUp.Enabled := True;
end; (* ButtonStartClick *)

procedure TFormJirai.ButtonGiveUpClick(Sender: TObject);
begin
  GameOver(False);
end; (* ButtonGiveUpClick *)

procedure TFormJirai.ImageHaraMouseDown(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
var
  Gyou, Retu : Byte;
  Bakuhatu : Boolean;
begin
  Gyou := Y div KukakuSize;
  Retu := X div KukakuSize;
  if (Gyou in [1..GyouSuu]) and (Retu in [1..RetuSuu])
  then begin
    case Button of
      mbRight : Arudarou(Gyou, Retu);
      mbLeft : Naidarou(Gyou, Retu);
    end;
    Bakuhatu := Hara[Gyou, Retu].Joutai = kjNaiHazure;
    if (AruNokori = 0) or (NaiNokori = 0) or Bakuhatu
    then GameOver(not Bakuhatu);
  end;
end; (* ImageHaraMouseDown *)
```

16.6.1 メソッド追加

```
procedure TFormJirai.StandBy;
  (* スタンバイ *)
```

```

var
  Gyou,Retu : Byte;
begin
  for Gyou := 1 to GyouSuu do
    for Retu := 1 to RetuSuu do
      Hara[Gyou,Retu].Joutai := kjMada;
    AruNokori := JiraiSuu;
    NaiNokori := KukakuSuu-JiraiSuu;
    KizaiNokori := JiraiSuu;
  end; (* StandBy *)

procedure TFormJirai.GameOver(Seikou : Boolean);
  (* ゲーム終了 *)
var
  Gyou,Retu : Byte;
begin
  // 未開の区画を開ける
  for Gyou := 1 to GyouSuu do
    for Retu := 1 to RetuSuu do
      with Hara[Gyou,Retu] do
        begin
          if Joutai = kjMada
            then if Seikou
                  then if Aru
                        then Joutai := kjAruAtari
                             else Joutai := kjNaiAtari
                       else if Aru
                             then Joutai := kjNaiHazure
                                  else Joutai := kjNaiAtari;
          end;
        HaraWoKaku;
        if Seikou
          then ShowMessage(#13+' おめでとう '+#13)
          else ShowMessage(#13+' 残念でした '+#13);
        with FormJirai do
          begin
            ImageHara.Enabled := False;
            RadioGroupSize.Enabled := True;
            ButtonNew.Enabled := True;
            ButtonStart.Enabled := True;
            ButtonGiveUp.Enabled := False;
          end;
        end; (* GameOver *)

procedure TFormJirai.Arudarou(Gyou,Retu : TBanchi);
  (* 地雷があるだろうと思って掘る *)
begin
  with Hara[Gyou,Retu] do
    if (Joutai = kjMada) and (KizaiNokori > 0)
      then begin
        if Aru
          then begin
            Joutai := kjAruAtari;
            Dec(AruNokori);
          end
        else begin
            Joutai := kjAruHazure;
            Dec(NaiNokori);
          end;
      end;
  end;

```



```
        end;
        Dec(KizaiNokori);
        KukakuWoKaku(Gyou,Retu);
    end;
end; (* Arudarou *)

procedure TFormJirai.Naidarou(Gyou,Retu : TBanchi);
    (* 地雷がないだろうと思って掘る *)
var
    G,R : TBanchi;
begin
    with Hara[Gyou,Retu] do
        if Joutai = kjMada
            then begin
                if Aru
                    then begin
                        Joutai := kjNaiHazure;
                        Dec(AruNokori);
                    end
                else begin
                    Joutai := kjNaiAtari;
                    Dec(NaiNokori);
                    if Hint = 0
                        then for G := Gyou-1 to Gyou+1 do
                            for R := Retu-1 to Retu+1 do
                                Naidarou(G,R);
                            end;
                        end;
                    KukakuWoKaku(Gyou,Retu);
                end;
            end;
end; (* Naidarou *)
```

16.6.2 実行

一応ゲームを実行できます。

16.7 問題

16.7.1 個数表示

メインパネルに PanelJiraiSuu, PanelAruNokori, PanelNaiNokori, PanelKizaiNokori を置いて、これらの数を表示するようにしなさい。

16.7.2 時間

タイマーを使って、ゲーム開始からの経過時間を表示するようにしなさい。一定時間たったら地雷が爆発するようにすると面白いですね。