

5 解説 — 乱数, ペンモード —

5.1 乱数について

Random (値は $0 \leq r < 1$ の範囲の実数) または Random(n) (値は $0 \leq r < n$ の整数) を実行すると、でたらめな値が得られます。でたらめと言っても、乱数の種 (random seed) と呼ばれる初期値から始めて、ある数式に従って計算して値を出している、“全くでたらめ” という訳ではありません。“予測できない数” と言ったらよいでしょうか。

プログラムを実行するといつも同じ乱数の種から開始します。ですから、Moyou プログラムを実行して模様を描いて、いったん終了してもう一度実行すると前回と同じ模様を描きます。また、隣の人と比べても同じ模様になっています。

違う模様を描くためには、Randomize を実行します。Randomize を実行すると、そのときの Time (年月日時分秒ミリ秒) を元にして新しい乱数の種を作ります。Randomize は 1 つのプログラムで 1 回だけ最初に行うものです。ふつうは、Form の OnCreate ハンドラーの中に書きます。

```

procedure TFormMain.FormCreate(Sender: TObject);
begin
    Randomize;                                (* ここに書く *)
    WindowState := wsMaximized;
    ImageMain.Canvas.Rectangle(Rect(0,0,100,100));
end; {FormCreate}

```

たまたま、Random を実行する直前に Randomize を書いている人がいます。この部分を繰り返し実行すると、その都度 Time を乱数の種にして計算するので、乱数どころか 1 ミリ秒間同じ値を続けて発生します。

```

procedure TFormMain.ButtonRandomClick(Sender: TObject);
var
    N : Integer;
    X1,Y1,X2,Y2 : Integer;
    Iro : TColor;
begin
    for N := 1 to 100 do
        begin
            Randomize;                        (* ここに書くとうどうなるかな *)
            X1 := Random(100);
            Y1 := Random(100);
            X2 := Random(100);
            Y2 := Random(100);
            Iro := Random($1000000);
途中省略
        end;
        ButtonCopyClick(Sender);
    end;
end;

```

5.2 ペンモードについて

グラフィックスで色を指定するものが 3 つあります。

Pen.Color Rectangle, Ellipse などの輪郭 LineTo
 Font.Color TextOut の字
 Brush.Color Rectangle, Ellipse などの内部 TextOut の地 Floodfill

実際に描く色は、ふつうはペンやブラシの色そのもので描きますが、すでに画面に描かれている背景の色との組み合わせによって定まる別の色で描くようにすることもできます。ペンの色と背景の色との組み合わせによって実際に描く色を定める定め方を指定するのがペンモードです。

以前説明した光の三原色とそれを足し合わせたときの色を思い出してください。

青	緑	赤	
0	0	0	黒
0	0	1	赤
0	1	0	緑
0	1	1	黄
1	0	0	青
1	0	1	紫
1	1	0	水
1	1	1	白

ペンモード pmMerge は、ペンやブラシの色 P と背景の色 Q の各三原色について論理和 OR¹をした色で描きます。

Q	黒	赤	緑	黄	青	紫	水	白
P	000	001	010	011	100	101	110	111
黒	黒	赤	緑	黄	青	紫	水	白
000	000	001	010	011	100	101	110	111
赤	赤	赤	黄	黄	紫	紫	白	白
001	001	001	011	011	101	101	111	111
緑	緑	黄	緑	黄	水	白	水	白
010	010	011	010	011	110	111	110	111
黄	黄	黄	黄	黄	白	白	白	白
011	011	011	011	011	111	111	111	111
青	青	紫	水	白	青	紫	水	白
100	100	101	110	111	100	101	110	111
紫	紫	紫	白	白	紫	紫	白	白
101	101	101	111	111	101	101	111	111
水	水	白	水	白	水	白	水	白
110	110	111	110	111	110	111	110	111
白	白	白	白	白	白	白	白	白
111	111	111	111	111	111	111	111	111

¹OR 演算 : 0∨0 = 0, 0∨1 = 1, 1∨0 = 1, 1∨1 = 1

ペンモード pmMask は、ペンやブラシの色 P と背景の色 Q の各三原色について論理積 AND²をした色で描きます。

下の表を完成させなさい (色だけでいいです)。

Q P	黒 000	赤 001	緑 010	黄 011	青 100	紫 101	水 110	白 111
黒 000								
赤 001								
緑 010								
黄 011								
青 100								
紫 101								
水 110								
白 111								

ペンモード pmXor は、ペンやブラシの色 P と背景の色 Q の各三原色について排他的論理和 XOR³をした色で描きます。

下の表を完成させなさい (色だけでいいです)。

Q P	黒 000	赤 001	緑 010	黄 011	青 100	紫 101	水 110	白 111
黒 000								
赤 001								
緑 010								
黄 011								
青 100								
紫 101								
水 110								
白 111								

ペンモード pmNot はペンの色に関係なく背景の色を補色に反転します。

ペンモード pmNotXor は P と Q の排他的論理和の否定、すなわち同値 Iff⁴をした色で描きます。

pmNot と pmXor , pmNotXor は同じ図形を 2 回描くと元に戻るの、背景を壊したくない時に便利です。

²AND 演算 : $0 \& 0 = 0, 0 \& 1 = 0, 1 \& 0 = 0, 1 \& 1 = 1$

³XOR 演算 : $0 \oplus 0 = 0, 0 \oplus 1 = 1, 1 \oplus 0 = 1, 1 \oplus 1 = 0$

⁴同値演算 : $0 \equiv 0 = 1, 0 \equiv 1 = 0, 1 \equiv 0 = 0, 1 \equiv 1 = 1$

5.3 確認プログラム

ペンモードの機能を確認するプログラム PenMode を作ります。

5.3.1 新規アプリケーション

フォルダー	Sen	以前作ったもの
ユニット プロジェクト	PenModeP	

5.3.2 フォーム

Name	FormMain
Caption	ペンモード確認

5.3.3 メインパネルと終了ボタン

いつもと同じように、右端にメインパネル、その中の下側に終了ボタンを作ってください。

5.3.4 メインイメージ

Image をフォームの中、メインパネルの外に置く。

Name	ImageMain
Width	800
Height	600

Form が Image より小さいので、スクロールバーが現れます。スクロールバーがちょうど消えるまで、Form を広げてください。

5.3.5 初期化ボタン

Button をメインパネルの中、終了ボタンの上側に置く。

Name	ButtonSyokika
Caption	初期化

OnClick ハンドラーと初期化メソッド

イベントハンドラーは、オブジェクトインスペクターのイベント欄をダブルクリックしてスケルトンなどを自動記入させて記入すること、メソッドは、TFormMain の private 部にヘッダーを宣言することを忘れないようにしてください。

```

procedure TFormMain.ButtonSyokikaClick(Sender: TObject);
begin
  Syokika;
end;

```

```

(***** 一般のプロシージャ *****)
(***** フォームのメソッド *****)
procedure TFormMain.Syokika;
{ 画面半分を黒, 半分を白に初期化する }
begin
  with ImageMain,
    Canvas do
    // ImageMain と
    // ImageMain.Canvas を省略
    begin
      Pen.Mode := pmCopy;
      Brush.Color := clWhite;
      Rectangle(400,0,Width,Height); // ImageMain の Width,Height
      Brush.Color := clBlack;
      Rectangle(0,0,400,Height);
    end;
end; {Syokika}
(***** イベントハンドラー *****)

```

5.3.6 描画ボタン

Button を初期化ボタンの上側に置く。

Name	ButtonByouga
Caption	描画

OnClick ハンドラーと描画メソッド

```

procedure TFormMain.ButtonByougaClick(Sender: TObject);
begin
  TriColor(200,310);
  TriColor(600,310);
end;

```

トリコロール TriColor はフランスの三色旗ですが, ここでは三つの色の円を描きます。

```

(***** フォームのメソッド *****)
procedure TFormMain.Circle(X,Y,R : Integer);
{ 中心が (X,Y) 半径が R の円を描く }
begin
  ImageMain.Canvas.Ellipse(X-R,Y-R,X+R,Y+R);
end; {Circle}

procedure TFormMain.TriColor(X0,Y0 : Integer);
{ 赤緑青の3つの円と, 補色の3つの円を描く }
begin
  with ImageMain.Canvas do // ImageMain.Canvas を省略
  begin
    Brush.Color := clRed;
    Circle(X0,Y0-186,100);
    Brush.Color := clLime;
    Circle(X0-50,Y0-100,100);
    Brush.Color := clBlue;
    Circle(X0+50,Y0-100,100);
  end;
end;

```

```

    Brush.Color := clAqua;
    Circle(X0,Y0+100,100);
    Brush.Color := clFuchsia;
    Circle(X0-50,Y0+186,100);
    Brush.Color := clYellow;
    Circle(X0+50,Y0+186,100);
  end;
end; {TriColor}

```

$\text{Ellipse}(l, t, r, b)$ は $l \leq x < r$ かつ $t \leq y < b$ の長方形に内接する楕円を描くプロシージャです。長方形が正方形のときには円になります。

5.3.7 実行

まだ、ペンモードがデフォルト（初期値）の `pmCopy` なので、円が重なっている部分は、後から描いた円が前に描いた円を消してしまいます。

5.3.8 ペンモード指定用

`RadioGroup` をメインパネルの中に置く。

```

Align      alTop
Name       RadioGroupMode
Items      pmCopy
           pmMerge (OR)
           pmMask (AND)
           pmXor
           pmNotXor
           pmNot

```

OnClick ハンドラー

```

procedure TFormMain.RadioGroupModeClick(Sender: TObject);
begin
  with ImageMain.Canvas do
    begin
      case RadioGroupMode.ItemIndex of
        0 : Pen.Mode := pmCopy;      // ペンブラシの色（通常はこれ）
        1 : Pen.Mode := pmMerge;    // 画面の色とペンブラシの色との OR
        2 : Pen.Mode := pmMask;    // 画面の色とペンブラシの色との AND
        3 : Pen.Mode := pmXor;     // 画面の色とペンブラシの色との XOR
        4 : Pen.Mode := pmNotXor;   // 画面の色とペンブラシの色との XOR の補色
        5 : Pen.Mode := pmNot;     // 画面の色 の補色
      end;
    end;
  TriColor(200,310);
  TriColor(600,310);
end;

```

5.3.9 実行

ラジオグループでモードを変更すると、新しいモードで描画します。描画ボタンをクリックすると同じモードで再描画します。

円が重なっている部分は、円を描くたびに色の論理計算をするので、pmXor 以下はちょっと複雑です。実際に描く（クリックする）前に何色になるか考えてください。

次のことを確認してください。

1. pmMerge では、黒地に三原色の円を重ねて描くと 8 色現れる。色を重ねると明るくなる。
2. pmMask では、白地に三原色の補色の円を重ねて描くと 8 色現れる。色を重ねると暗くなる。
3. 描画ボタンをクリックする（同じ図を再描画する）と、色が変化するモードと変化しないモードがあること。変化する場合は、元の背景に戻っている。
4. pmXor のとき、背景が黒だとペンの色、白だとペンの色の補色で描く。pmNotXor だとその逆である。

5.3.10 背景に花

L ドライブの“大矢”に Hana.bmp という画像が入っています。それを自分のフォルダー Z:\Joron\Sen にコピーしてください。下記からダウンロードすることもできます。

<http://www.ss.u-tokai.ac.jp/~ooya/Jugyou/JoronB/Hana.bmp>

Button を描画ボタンと初期化ボタンの間に置く。

Name	ButtonHana
Caption	花に初期化

OnClick ハンドラー

```
procedure TFormMain.ButtonHanaClick(Sender: TObject);
begin
  ImageMain.Picture.LoadFromFile('Z:\Joron\Sen\Hana.bmp');
end;
```

もちろん、ファイル名のパス Z:\Joron\Sen は Hana.bmp を保存した自分のフォルダーを書きます。

Hana.Bmp は 800×600 の画像ですから、ImageMain を置いたときに Width=800, Height=600 にしたのです。サイズが変わってないことを確認してください。

5.3.11 実行

画像の上に描くと様子がよくわかりますね。

5.3.12 試行

わざと ImageMain のサイズを変えて実行してみてください。サイズが合わないと、こういう現象が起きます。

5.3.13 模様プログラム修正

以前作成した模様プログラムを次のように修正して実行し、できた模様を保存すると、PenMode のプログラムで利用できます。

- ImageMoyou を 800×600 にする。
- 出力ファイル '模様.bmp' に上と同じパスをつける。

ついでに、Form の OnCreate ハンドラーに Randomize を追加しましょう。