

# Как Windows превратить в illuminators – создание окон произвольной формы в Delphi средствами Windows API

Вопрос создания непрямоугольных окон часто интересует начинающих программистов и время от времени обсуждается на форумах разработчиков в среде Delphi. А вообще, нужно ли это кому-нибудь? Ответ - да! Это уже было нужно таким известным фирмам, как Symantec (Norton Utilities, Norton CrashGuard), Microsoft (Приложение "Часы" в Windows NT4 может принимать круглую форму, Deluxe CD Player из MS Plus! 98 имеет вид прямоугольника со скругленными краями). У Borland Jbuilder 2 в окне начальной загрузки стрела крана "выскочила" за пределы прямоугольника.

Программы для видеокарт TV Capture фирмы AverMedia имитируют пульт управления. Окно переводчика Magic Goody принимает вид гуся, разгуливающего по экрану.

Список можно продолжить, а вывод такой: окно "хитрой" формы – это "изюминка" оформления Вашей программы, нечто запоминающееся, дополнительный плюс в борьбе за потенциального покупателя. Главное в этом – не переборщить. Вряд ли будет удобно работать с текстовым редактором в треугольном окне. Окна произвольной формы неплохо смотрятся при начальной загрузке (Splash) и, возможно, в качестве окна "О программе ...".

Как это делается? Средствами Delphi – достаточно просто. Приведенные ниже примеры можно также перевести в C++ Builder или Visual C++.



```
type
  TForm1 = class(TForm)
  // Данную процедуру будем использовать для задания
  // формы окна
    procedure FormCreate(Sender: TObject);
  protected
  // Данную процедуру необходимо переопределить,
  // чтобы иметь возможность перемещать окно "мышкой"
  // не только за заголовок, который в данном
  // случае отсутствует
    procedure WMNCHitTest(var Message: TWMNCHitTest);
      message WM_NCHITTEST;
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

---

При создании окна непрямоугольной формы используются API функции **CreateEllipticRgn, CreateRectRgn, CreatePolygonRgn, CreateRoundRectRgn, CombineRgn.**

---

```
procedure TForm1.FormCreate(Sender: TObject);
var
  R1, R2: HRgn;
begin
  // Создаем линзу (пересечение двух эллипсов)
  R1 := CreateEllipticRgn(-150,-300,363,400); // Задаем
  // координаты эллипса. Верхняя левая точка формы имеет
  // координаты 0,0

  R2 := CreateEllipticRgn(363+150,-300,5,400);
  CombineRgn(R1,R1,R2, RGN_AND); // Получаем пересечение

  // Отрезаем сверху слева
  R2:= CreateEllipticRgn(-40,-30,182,20);
  CombineRgn(R1,R1,R2, RGN_DIFF); // Вычитаем
```

```
// Отрезаем сверху справа
R2:= CreateEllipticRgn(180,-30,363+40,20);
CombineRgn(R1,R1,R2,RGN_DIFF);

// Назначаем полученный регион форме
SetWindowRgn(Handle, R1, True);

end;
```

---

Если Вы все ввели правильно, то при запуске проекта получите окно в виде щита, как у Norton CrashGuard. Как это получилось, можно понять из схемы 1. Зеленые эллипсы при пересечении образуют линзу, красные эллипсы вычитаются.

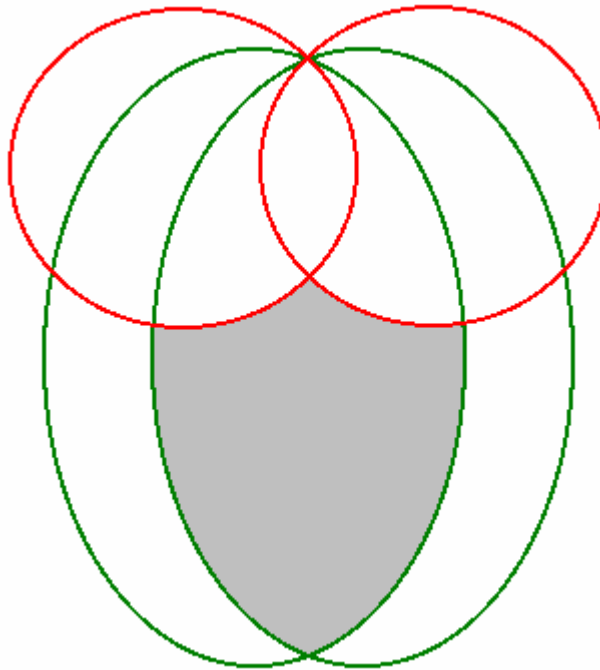


схема 1.

Размер формы должен позволить разместиться на ней щиту полностью, иначе Вы получите усеченный вариант изображения.

Переопределение функции **WMNCHitTest** позволит перетаскивать окно, захватив его мышкой.

---

```
procedure TForm1.WMNCHitTest(var Message: TWMNCHitTest);
begin
  inherited;
  Message.Result := HTCAPTION;
end;
```

---

Теперь немного усложним задачу. Создадим фигурный щит, как показано на схеме 2.

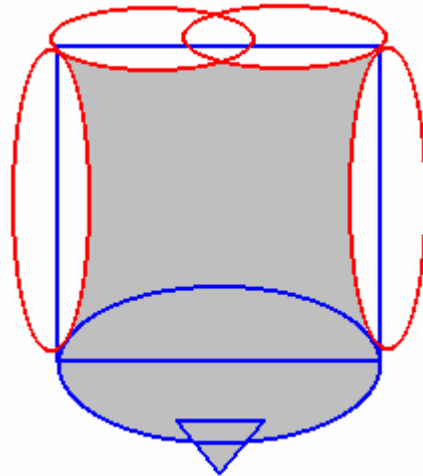


схема 2.

Синие фигуры складываются, красные вычитаются. Исходный код приведен ниже.

```
procedure TForm1.FormCreate(Sender: TObject);
var
  R1,R2 : HRgn;
  P : array [0..2] of TPoint;
  // Массив для создания полигона
begin
  // Создаем начальный прямоугольник
  R1 :=CreateRectRgn(0,0,270,233);

  // Отрезаем сверху слева
  R2:= CreateEllipticRgn(-20,-20,135+5,20);
  CombineRgn(R1,R1,R2,RGN_DIFF);

  // Отрезаем сверху справа
  R2:= CreateEllipticRgn(135-5,-20,270+20,20);

  CombineRgn(R1,R1,R2,RGN_DIFF);

  // Отрезаем левый бок
  R2:= CreateEllipticRgn(-15,10,8,233);
  CombineRgn(R1,R1,R2,RGN_DIFF);

  // Отрезаем правый бок
  R2:= CreateEllipticRgn(270-7,10,270+16,233);
  CombineRgn(R1,R1,R2,RGN_DIFF);

  // Добавляем овал снизу
  R2:= CreateEllipticRgn(0,150,270,320);
  CombineRgn(R1,R1,R2,RGN_OR);

  // Добавляем острие (треугольник) снизу
  P[0] := Point(135-40, 310);
  P[1] := Point(135+40, 310);
  P[2] := Point(135, 335);
  R2 := CreatePolygonRgn(P, 3, WINDING);
  CombineRgn(R1,R1,R2,RGN_OR);

  // Назначаем итоговый регион нашей форме
  SetWindowRgn(Handle, R1, True);
end;
```

Попробуем создать что-либо еще более сложное и непохожее на щит.  
Создание головы коровы (Схема 3).

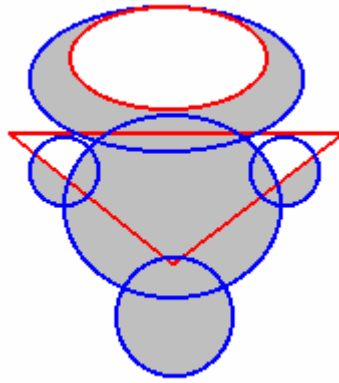


схема 3.

---

```
procedure TForm1.FormCreate(Sender: TObject);
var
  R1,R2 : HRgn;
  P : array [0..2] of TPoint;
begin
  R1:= CreateEllipticRgn(135-110, 45, 135-40, 105);
  // Овал, образующий ухо
  R2:= CreateEllipticRgn(135+40, 45, 135+110, 105);
  // Овал, образующий ухо
  CombineRgn (R1,R1,R2,RGN_OR);

  // Отсекаем треугольником лишнюю часть овалов,
  // тем самым завершаем формирование ушей
  P[0] := Point(135-140, 40);
  P[1] := Point(135+140, 40);
  P[2] := Point(135, 140);
  R2 := CreatePolygonRgn(P, 3, WINDING);
  CombineRgn (R1,R1,R2,RGN_DIFF);

  // Формируем полумесяц рогов
  R2:= CreateEllipticRgn(135-90, 0, 135+90, 100);
  CombineRgn (R1,R1,R2,RGN_OR);

  R2:= CreateEllipticRgn(135-70, -5, 135+70, 70);
  CombineRgn (R1,R1,R2,RGN_DIFF);

  // Формируем морду
  R2:= CreateEllipticRgn(135-70, 60, 135+70, 170);
  CombineRgn (R1,R1,R2,RGN_OR);

  R2:= CreateEllipticRgn(135-40, 150, 135+40, 210);
  CombineRgn (R1,R1,R2,RGN_OR);

  // Назначаем итоговый регион нашей форме
  SetWindowRgn(Handle, R1, True);
end;
```

---

До сих пор в примерах мы рассматривали регионы с абсолютными значениями линейных величин. Ниже приведен пример прямоугольного окна, которое масштабирует свою форму в зависимости от его размера. Исходный код, приведенный ниже, создает окно в виде бабочки, причем бабочка использует максимально высоту и ширину исходной формы.

---

```

procedure TForm1.FormCreate(Sender: TObject);
var
  R1, R2 : HRgn;
  P : array [0..2] of TPoint;
  X : Word;
begin
  // левое верхнее крыло
  R1 :=CreateEllipticRgn(Round(-Width*0.4),
0, Round(Width*0.49), Round(Height*1.1));
  // правое верхнее крыло
  R2 :=CreateEllipticRgn(Round(Width*0.51),
0, Round(Width*1.4), Round(Height*1.1));
  CombineRgn(R2, R1, R2, RGN_OR);
  // отсекаем лишнее от верхних крыльев,
  // остаются линзы на пересечении эллипсов
  R1 :=CreateEllipticRgn(0, Round(-Height*0.3),
Width, Round(Height*0.71));
  CombineRgn(R1, R1, R2, RGN_AND);

  //эллипс - основа нижних крыльев
  R2 :=CreateEllipticRgn(Round(Width*0.1),
Round(Height*0.65), Round(Width*0.9), Height);
  CombineRgn(R1, R1, R2, RGN_OR);
  // вырезаем эллипс - разрез между нижних крыльев
  R2 :=CreateEllipticRgn(Round(Width*0.3),
Round(Height*0.7), Round(Width*0.7), Round(Height*1.5));
  CombineRgn(R1, R1, R2, RGN_DIFF);

  // вертикальный эллипс - туловище бабочки
  R2 :=CreateEllipticRgn(Round(Width*0.46),
Round(Height*0.3), Round(Width*0.54),
Round(Height*0.8));
  CombineRgn(R1, R1, R2, RGN_OR);

  // голова - круг; за основу берем меньшую
  // из двух величин - высоты и ширины окна
  X := Width;
  if Height < X then X := Height;
  X := Round(X/18);
  R2 :=CreateEllipticRgn(Round(Width*0.5)-X,
Round(Height*0.3)-X, Round(Width*0.5)+X,
Round(Height*0.3)+X);
  CombineRgn(R1, R1, R2, RGN_OR);

  // левый усик
  P[0] := Point(Round(Width*0.5), Round(Height*0.3));
  P[1] := Point(Round(Width*0.35), Round(Height*0.01));
  P[2] := Point(Round(Width*0.355)+1, 0);
  R2 := CreatePolygonRgn(P, 3, WINDING);
  CombineRgn(R1, R1, R2, RGN_OR);

  // правый усик
  P[0] := Point(Round(Width*0.5), Round(Height*0.3));
  P[1] := Point(Round(Width*0.655)+1, Round(Height*0.01));
  P[2] := Point(Round(Width*0.65), 0);
  R2 := CreatePolygonRgn(P, 3, WINDING);
  CombineRgn(R1, R1, R2, RGN_OR);

  // острие на крыле слева снизу
  P[0] := Point(Round(Width*0.15), Height);
  P[1] := Point(Round(Width*0.2), Round(Height*0.8));
  P[2] := Point(Round(Width*0.3), Round(Height*0.9));
  R2 := CreatePolygonRgn(P, 3, WINDING);
  CombineRgn(R1, R1, R2, RGN_OR);

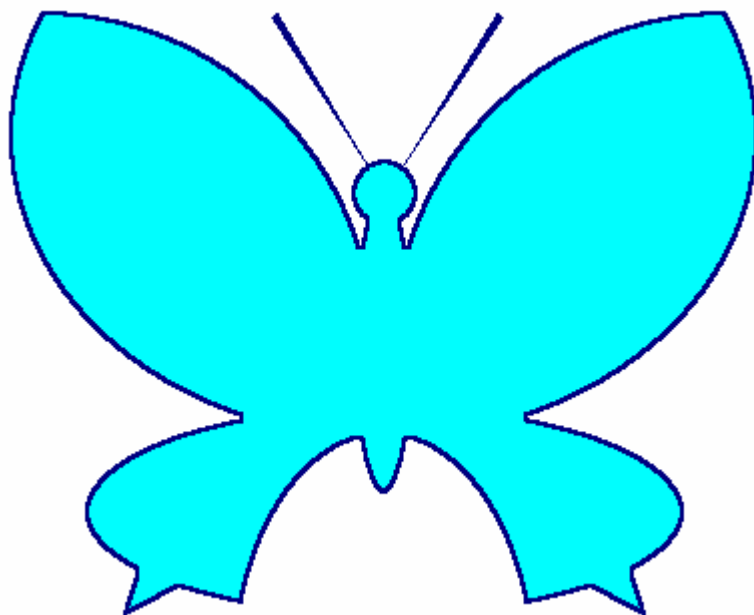
  // острие на крыле справа снизу
  P[0] := Point(Round(Width*0.85), Height);
  P[1] := Point(Round(Width*0.8), Round(Height*0.8));

```

```
P[2] := Point(Round(Width*0.7), Round(Height*0.9));
R2 := CreatePolygonRgn(P, 3, WINDING);
CombineRgn(R1, R1, R2, RGN_OR);

// Назначаем полученный регион форме
SetWindowRgn(Handle, R1, True);
end;
```

---



---

Если грамотно разложить фигуру на элементарные составляющие, то Вам вполне по силам создать окно абсолютно любой формы. Это похоже на детскую игру "конструктор", только Ваши "кубики" намного разнообразнее.

Для завершения проекта необходимо создать фоновую картинку, которая подчеркнет границы нового окна. И обязательно установить свойство формы `Scaled = False`, иначе фоновая картинка и форма могут "разъехаться" при использовании нестандартных видеорежимов или стилей оформления Windows.

В заключение следует сказать, что существуют готовые компоненты и библиотеки компонент для решения подобных задач, например, `CoolForm`, `TPlasmaForm`. Однако при использовании компонент от сторонних производителей могут возникнуть проблемы лицензионности их использования и проблемы перехода на новую версию компилятора. А приведенные в данной статье примеры компилируются без изменений в исходном коде на Borland Delphi 3.0 - 2010 и, вероятно, будут совместимы с последующими версиями.

***Данный файл (статья) может копироваться и распространяться свободно при условии указания оригинального авторства.***

© Георгий Чернилевский  
master\_of\_storm@hotmail.com