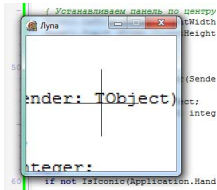


Как написать приложение, которое может увеличивать часть экрана рабочего стола, вроде лупы? Такие особенности часто применяются во многих приложениях Windows.



Создайте новое приложение. На форму поместите компонент **TImage** (*Image1*). Его свойство

Align

выставьте в

alClient

, а свойство

Visible

в

False

. Затем

TPanel

(

Panel1

), на которой будут находиться остальные компоненты. Это сделано для того, чтобы можно было легко скрывать/отображать компоненты управления и настройки. Свойству

Caption

присвойте пустую строку. На

Panel1

поместите компонент

TGroupBox

(

GroupBox1

). Свойство

Caption

этого компонента установите в

Кратность увеличения

. Внутри его поместите

TTrackBar

(

Slider

). Свойства

Max

=

4

```
,  
Min  
=  
1  
,  
PageSize  
=  
1  
,  
ShowSelRange  
=  
True
```

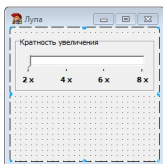
. Под ним поместите компоненты

TLabel

с надписями

2x, 4x, 6x и 8x

, как показано на рисунке.



Ниже поместите **TCheckBox** (*cbShowCross*). Свойство **Checked** установите в *True*. А также

TTimer

```
(  
Timer1
```

```
), свойство
```

Interval

равен

```
25  
.
```

Вся идея состоит в том, чтобы отслеживать положение курсора мыши и увеличивать изображение вокруг него в несколько раз (от 2-х до 8-ми).

Теперь запишем некоторый код. В событии **OnResize** формы запишите:

```
procedure TForm1.FormResize(Sender: TObject); begin { Устанавливаем панель по  
центру формы } Panel1.Left:= (Form1.ClientWidth div 2) - Panel1.Width div 2;
```

```
Panel1.Top:= (Form1.ClientHeight div 2) - Panel1.Height div 2; Image1.Picture:= nil; end;
```

Далее напишем обработчик сообщения **WM_GETMINMAXINFO**:

```
private { Private declarations } procedure WMGetMinMaxInfo( var Msg:
TWMMGetMinMaxInfo); message WM_GETMINMAXINFO; { ... } procedure
TForm1.WMGetMinMaxInfo(var Msg: TWMMGetMinMaxInfo); begin inherited; {
Минимальный размер формы } Msg.MinMaxInfo^.ptMinTrackSize:= Point(158, 177); {
Максимальный размер формы } Msg.MinMaxInfo^.ptMaxTrackSize:= Point(350, 350); end;
```

Теперь будем отслеживать события в таймере:

```
procedure TForm1.Timer1Timer(Sender: TObject); var srect, drect, posForm: TRect;
iWidth, iHeight, dmX, dmY: integer; itmpX, itmpY: real; c: TCanvas; hDesktop: HWND;
cur: TPoint; begin if not IsIconic(Application.Handle) then begin hDesktop:=
GetDesktopWindow; GetCursorPos(cur); posForm:= Rect(Form1.Left, Form1.Top,
Form1.Left + Form1.Width, Form1.Top + Form1.Height); if not PtInRect(posForm, cur)
then begin if Panel1.Visible then Panel1.Visible:= false; if not Image1.Visible then
Image1.Visible:= true; iWidth:= Image1.Width; iHeight:= Image1.Height; drect:=
Rect(0, 0, iWidth, iHeight); itmpX:= iWidth / (Slider.Position * 4); itmpY:= iHeight /
(Slider.Position * 4); srect:= Rect(cur.X, cur.Y, cur.X, cur.Y); InflateRect(srect,
round(itmpX), round(itmpY)); { перемещаем, если выходит за пределы } if srect.Left
Screen.Width then OffsetRect(srect, -(srect.Right - Screen.Width), 0); if srect.Bottom
> Screen.Height then OffsetRect(srect, 0, -(srect.Bottom - Screen.Height)); c:=
TCanvas.Create; try c.Handle:= GetDC(GetDesktopWindow);
Image1.Canvas.CopyRect(drect, c, srect); finally ReleaseDC(hDesktop, c.Handle);
c.Free; end; if cbShowCursor.Checked then begin with Image1.Canvas do
begin dmX:= Slider.Position * 2 * (cur.X - srect.Left); dmY:= Slider.Position *
2 * (cur.Y - srect.Top); { рисуем перекрестье } MoveTo(dmX - (iWidth div 4),
dmY); LineTo(dmX + (iWidth div 4), dmY); MoveTo(dmX, dmY - (iHeight div 4));
LineTo(dmX, dmY + (iHeight div 4)); end; end;
Application.ProcessMessages; end else begin if not Panel1.Visible then
Panel1.Visible:= true; if Image1.Visible then Image1.Visible:= false; end;
end; end;
```

И, наконец, отключаем таймер при выходе из приложения:

```
procedure TForm1.FormDestroy(Sender: TObject); begin Timer1.Interval:= 0; end;
```

Исходник программы находится [здесь](#).