unit Unit1;

interface

uses

Winapi.Windows, Winapi.Messages, System.SysUtils, System.Variants, System.Classes, Vcl.Graphics,

Vcl.Controls, Vcl.Forms, Vcl.Dialogs,JPEG,extCtrls,System.Types;

type

TForm1 = class(TForm)

image1: TImage;

procedure FormCreate(Sender: TObject);

procedure FormMouseWheel(Sender: TObject; Shift: TShiftState;

WheelDelta: Integer; MousePos: TPoint; var Handled: Boolean);

private

FOrgImgBounds: TRect;

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.FormCreate(Sender: TObject);

begin

DoubleBuffered := True;

image1.Picture.LoadFromFile('C:\Temp\pic.jpg');

image1.Height := Round(image1.Width \* image1.Picture.Height / image1.Picture.Width);

FOrgImgBounds := image1.BoundsRect;

Image1.proportional:=True;

end;

procedure TForm1.FormMouseWheel(Sender: TObject; Shift: TShiftState;

WheelDelta: Integer; MousePos: TPoint; var Handled: Boolean);

const

ZoomFactor: array[Boolean] of Single = (0.9, 1.1);

var

R: TRect;

begin

MousePos := image1.ScreenToClient(MousePos);

with image1, MousePos do

if PtInRect(ClientRect, MousePos) and ((WheelDelta > 0) and

(Height < Self.ClientHeight) and (Width < Self.ClientWidth)) or

((WheelDelta < 0) and (Height > 20) and (Width > 20)) then

begin

R := BoundsRect;

R.Left := Left + X - Round(ZoomFactor[WheelDelta > 0] \* X);

R.Top := Top + Y - Round(ZoomFactor[WheelDelta > 0] \* Y);

R.Right := R.Left + Round(ZoomFactor[WheelDelta > 0] \* Width);

R.Bottom := R.Top + Round(ZoomFactor[WheelDelta > 0] \* Height);

BoundsRect := R;

Handled := True;

end;

end;

end.