//---------------------------------------------------------------------------

#include <vcl\vcl.h>

#pragma hdrstop

#include "Math.h"

#include "Unit1.h"

//---------------------------------------------------------------------------

#pragma resource "\*.dfm"

int tx=300,ty=200;

int tx1=100,ty1=200;

float theta=0.23;

int m=0;

TForm1 \*Form1;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

int oct[8][2]= {{30,50},{50,30},{50,-30},{30,-50},{-30,-50},{-50,-30},{-50,30},{-30,50}

};

POINT p[8];

Form1->Refresh();

for(int i=0;i<8;i++)

{

p[i].x=oct[i][0]+tx;

p[i].y=oct[i][1]+ty;

}

Form1->Canvas->Polygon(p,sizeof(p)/sizeof(POINT)-1);

int oct1[8][2]= {{30,50},{50,30},{50,-30},{30,-50},{-30,-50},{-50,-30},{-50,30},{-30,50}

};

POINT p1[8];

//Form1->Refresh();

for(int i=0;i<8;i++)

{

p1[i].x=oct1[i][0]+tx1;

p1[i].y=oct1[i][1]+ty1;

}

Form1->Canvas->Polygon(p1,sizeof(p1)/sizeof(POINT)-1);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button2Click(TObject \*Sender)

{

Timer1->Enabled = true;

Timer2->Enabled = false;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button3Click(TObject \*Sender)

{

Timer1->Enabled = false;

Timer2->Enabled = true;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Timer1Timer(TObject \*Sender)

{

int oct[8][2]= {{30,50},{50,30},{50,-30},{30,-50},{-30,-50},{-50,-30},{-50,30},{-30,50}

};

POINT p[8];

Form1->Refresh();

for(int i=0;i<8;i++)

{

p[i].x=oct[i][0]\*cos(theta) - oct[i][1]\*sin(theta) + tx;

p[i].y=oct[i][0]\*sin(theta) + oct[i][1]\*cos(theta) + ty;

}

Form1->Canvas->Polygon(p,sizeof(p)/sizeof(POINT)-1);

tx+=10;

theta+=0.23;

int oct1[8][2]= {{30,50},{50,30},{50,-30},{30,-50},{-30,-50},{-50,-30},{-50,30},{-30,50}

};

POINT p1[8];

//Form1->Refresh();

for(int i=0;i<8;i++)

{

p1[i].x=oct1[i][0]\*cos(theta) - oct1[i][1]\*sin(theta) + tx1;

p1[i].y=oct1[i][0]\*sin(theta) + oct1[i][1]\*cos(theta) + ty1;

}

Form1->Canvas->Polygon(p1,sizeof(p1)/sizeof(POINT)-1);

Form1->Canvas->MoveTo(155+m,200);

Form1->Canvas->LineTo(250+m,200);

Form1->Canvas->MoveTo(350+m,200);

Form1->Canvas->LineTo(450+m,200);

Form1->Canvas->LineTo(450+m,100);

Form1->Canvas->LineTo(350+m,100);

Form1->Canvas->LineTo(350+m,30);

Form1->Canvas->LineTo(20+m,30);

Form1->Canvas->LineTo(20+m,200);

Form1->Canvas->LineTo(40+m,200);

tx1+=10;

theta+=0.23;

m+=10;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Timer2Timer(TObject \*Sender)

{

int oct[8][2]= {{30,50},{50,30},{50,-30},{30,-50},{-30,-50},{-50,-30},{-50,30},{-30,50}

};

POINT p[8];

Form1->Refresh();

for(int i=0;i<8;i++)

{

p[i].x=oct[i][0]\*cos(theta) - oct[i][1]\*sin(theta) + tx;

p[i].y=oct[i][0]\*sin(theta) + oct[i][1]\*cos(theta) + ty;

}

Form1->Canvas->Polygon(p,sizeof(p)/sizeof(POINT)-1);

tx-=10;

theta-=0.23;

int oct1[8][2]= {{30,50},{50,30},{50,-30},{30,-50},{-30,-50},{-50,-30},{-50,30},{-30,50}

};

POINT p1[8];

//Form1->Refresh();

for(int i=0;i<8;i++)

{

p1[i].x=oct1[i][0]\*cos(theta) - oct1[i][1]\*sin(theta) + tx1;

p1[i].y=oct1[i][0]\*sin(theta) + oct1[i][1]\*cos(theta) + ty1;

}

Form1->Canvas->Polygon(p1,sizeof(p1)/sizeof(POINT)-1);

Form1->Canvas->MoveTo(155+m,200);

Form1->Canvas->LineTo(250+m,200);

Form1->Canvas->MoveTo(350+m,200);

Form1->Canvas->LineTo(450+m,200);

Form1->Canvas->LineTo(450+m,100);

Form1->Canvas->LineTo(350+m,100);

Form1->Canvas->LineTo(350+m,30);

Form1->Canvas->LineTo(20+m,30);

Form1->Canvas->LineTo(20+m,200);

Form1->Canvas->LineTo(40+m,200);

tx1-=10;

theta-=0.23;

m-=10;

}

//---------------------------------------------------------------------------