

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Институт естественных и точных наук
Кафедра прикладной математики и программирования

ОТЧЕТ

о выполнении лабораторной работы № 6 по дисциплине
«Математические основы компьютерной графики»

Автор работы,
студент группы ЕТ-211
_____ Савонин М.В.
« ____ » _____ 2022 г.

Руководитель работы,
старший преподаватель
_____ Шелудько А.С.
« ____ » _____ 2022 г.

Челябинск 2022

1 ЗАДАНИЕ

1. Написать программу для построения гладкой кривой по четырем опорным точкам. При выборе опорных точек текущие координаты указателя мыши должны отображаться в графическом окне. Интерфейс программы должен содержать следующие элементы управления:

- выбор опорных точек;
- построение кубической кривой Безье;
- построение кривой по алгоритму Чайкина;
- сохранение результата в файл;
- выход из программы.

2 МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

КРИВАЯ БЕЗЪЕ

Пусть x , y координаты центра, w , h длинна и ширина ромба, i номер угла и fi уголего наклона, то расчёт параметров ромба буду вычислять следующим способом. Если мы нажали на угол ромба, то:

Если угол чётный то:

$$r = (sqrt((mousex() - x)^2 + (mousey - y)^2) - w)/2.$$

$$w = w + r.$$

Иначе:

$$r = (sqrt((mousex() - x)^2 + (mousey - y)^2) - h)/2.$$

$$h = h + r.$$

$$x = x + cos(fi + i * M_PI/2) * r.$$

$$y = y + sin(fi + i * M_PI/2) * r.$$

Иначе:

$$x_1 = mousex() - x.$$

$$y_1 = mouseY() - y.$$

$$fi = asin(y_1 / (x_1 * x_1 + y_1 * y_1)^{1/2}).$$

Если $x_1 < 0$:

$$fi = M_PI - fi.$$

3 ТЕКСТ ПРОГРАММЫ

Файл main.cpp

```
#define _USE_MATH_DEFINES
#include <iostream>
#include <math.h>
#include "graphics.h"

using namespace std;

struct Point
{
    int x, y;
};

void change_figure(Point romb[2], int i, double &fi)
{
    if(i < 5)
    {
        int r;
        if(i % 2 == 0)
        {
            r = sqrt(pow(mousex() - romb[0].x, 2) + pow(mousey() - romb[0].y, 2));
            romb[1].x += r/2;
        }
        else
        {
            r = sqrt(pow(mousex() - romb[0].x, 2) + pow(mousey() - romb[0].y, 2));
            romb[1].y += r/2;
        }
        romb[0].x += cos(fi + i * M_PI / 2) * r / 2;
        romb[0].y += sin(fi + i * M_PI / 2) * r / 2;
    }
    else
    {
        int x = mouseX() - romb[0].x;
        int y = mouseY() - romb[0].y;
        fi = asin(y / pow(x * x + y * y, 1.0 / 2));
        if(x < 0)
        {
            fi = M_PI - fi;
        }
    }
}

void save()
{
    int width, height;
    IMAGE *output;

    width = getmaxx() + 1;
```

```

height = getmaxy() + 1;
output = createimage(width, height);

getimage(0, 0, width - 1, height - 1, output);
saveBMP("output.bmp", output);
freeimage(output);
}

int main()
{
    initwindow(1200, 800);
    Point romb[2] = {{600, 400}, {200, 200}};
    double fi = -M_PI/2;
    int p = 0;
    int topull = 0;

    while(1)
    {
        p = 1-p;
        setactivepage(p);

        setfillstyle(SOLID_FILL, BLACK);
        bar(0, 0, 1200, 800);
        setcolor(WHITE);
        for(int i = -1; i < 3; i++)
        {
            double points[2] = {abs(romb[1].x*cos(i*M_PI/2)+romb[1].y*sin(i*M_PI/2)),
                                abs(romb[1].x*sin(i*M_PI/2)-romb[1].y*cos(i*M_PI/2))};
            line(romb[0].x+cos(fi+i*M_PI/2)*points[0], romb[0].y+sin(fi+i*M_PI/2)*points[0],
                romb[0].x+cos(fi+i*M_PI/2)*points[0], romb[0].y+sin(fi+i*M_PI/2)*points[0]);
            circle(romb[0].x+(romb[1].x+20)*cos(fi), romb[0].y+(romb[1].x+20)*sin(fi), 20);
            if(topull != -2 and mousebuttons())
            {
                change_figure(romb, topull, fi);
            }
            else
            {
                topull = -2;
                for(int i = -1; i < 3; i++)
                {
                    double points = abs(romb[1].x*cos(i*M_PI/2)+romb[1].y*sin(i*M_PI/2));
                    if(abs(mousex()-(romb[0].x+cos(fi+i*M_PI/2)*points)) < 10)
                    {
                        topull = i;
                        change_figure(romb, i, fi);
                    }
                }
                if(abs(mousex()-(romb[0].x+(romb[1].x+20)*cos(fi))) < 8 and mouseY() < 8)
                {
                    topull = 5;
                    change_figure(romb, 5, fi);
                }
            }
        }
    }
}

```

```

    }

    rectangle(1150, 750, 1200, 800);
    line(1160, 760, 1190, 790);
    line(1190, 760, 1160, 790);
    rectangle(1050, 750, 1150, 800);
    outtextxy(1080, 765, "Save");
    if(abs(mousex()-1175) < 25 and abs(mousey()-775) < 25 and mouse
    {
        closegraph();
        return 0;
    }
    if(abs(mousex()-1100) < 50 and abs(mousey()-775) < 25 and mouse

    setvisualpage(p);
    delay(50);
}
}

```

4 РЕЗУЛЬТАТ РАБОТЫ

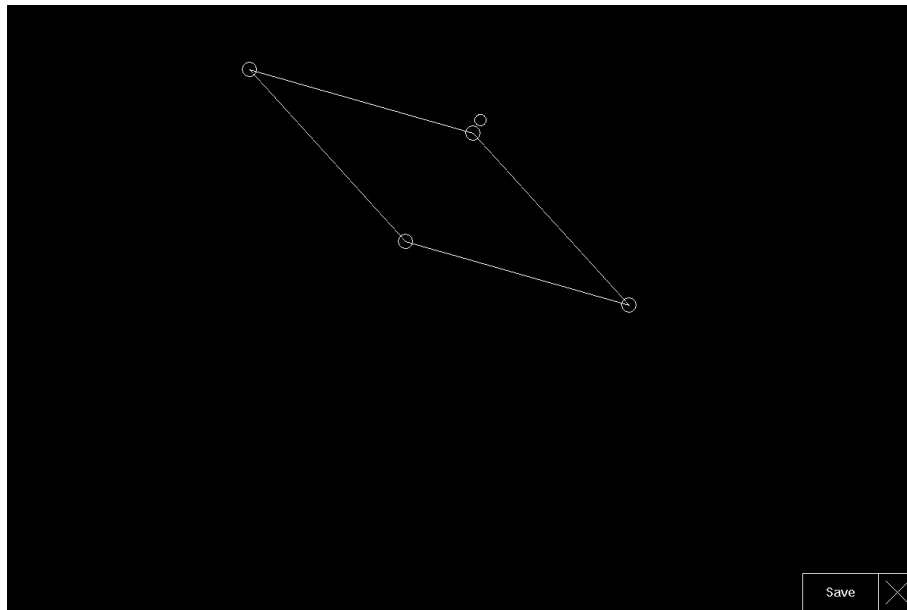


Рисунок 4.1 – Результат выполнения программы