

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 № 5 по дисциплине
«Математические основы компьютерной графики»

Автор работы,
студент группы ЕТ-212
_____ Шафикова М.А.
« ____ » _____ 2022 г.

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

Челябинск 2022

1 ЗАДАНИЕ

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

Пусть координаты четырех данных точек P_0, P_1, P_2, P_3 заданы двумерными координатами.

1) Кубическая кривая Безье с заданными точками будет иметь уравнение:

$$(1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t) t^2 P_2 + t^3 P_3$$

где $t \in [0,1]$ - параметр прямой.

2) Алгоритм Чайкина. В процессе соединения точек по алгоритму Чайкина получаем новые точки Q и R. Координаты этих точек рассчитываются по формулам:

$$dx = V_1.x - V_0.x;$$

$$dy = V_1.y - V_0.y;$$

Где V_1 и V_0 это координаты i -ой точки и $(i+1)$ -ой точки.

$$Q.x = V_0.x + dx * 0.25$$

$$Q.y = V_0.y + dy * 0.25$$

$$R.x = V_0.x + dx * 0.75$$

$$R.y = V_0.y + dy * 0.75$$

3 ТЕКСТ ПРОГРАММЫ

Файл main.cpp

```
#include <iostream>
#include <controls.h>
#include <graphics.h>
#include <task.h>

using namespace std;

int main()
{
    bool flag=false;
    initwindow(WIDTH, HEIGHT, "lines");
    create_back("bg.jpg");
    create_controls(482,725, STOP);
    create_controls(543,554, CHAIKIN);
    create_controls(723,488, BEZ);
    create_controls(633,740, SAVE);
    create_controls(738,639, EXIT);
    while(1)
    {
        while(mousebuttons()!=1)
        {
            coordinats(mousex(),mousey());
        }
        switch(select_control())
        {
            case FIELD:
                if(flag) break;
                createpoint(mousex(),mousey());
                break;
            case STOP:
                flag?flag=false:flag=true;
                delay(150);
                break;
            case CHAIKIN:
                if(!flag) break;
                chaikin();
                break;
            case BEZ:
                if(!flag) break;
                bez();
                break;
            case SAVE:
                save();
                break;
            case EXIT:
                return 0;
        }
    }
}
```

}

Файл task.h

```
#ifndef TASK_H
#define TASK_H

struct point
{
    int x0;
    int y0;
};

void save();
void chaikin();
void bez();
void createpoint(int,int);
void coordinats(int,int);

#endif
```

Файл task.cpp

```
#include <task.h>
#include <graphics.h>
#include <controls.h>
#include <string.h>

using namespace std;

point points[4];
int count_points=0;

void createpoint(int x0,int y0)
{
    if(count_points>3)
    {
        putpixel(points[count_points%4].x0,points[count_points%4].y0, C
        putpixel(points[count_points%4].x0+1,points[count_points%4].y0+
        putpixel(points[count_points%4].x0-1,points[count_points%4].y0-
        putpixel(points[count_points%4].x0+1,points[count_points%4].y0-
        putpixel(points[count_points%4].x0-1,points[count_points%4].y0+
    }
    points[count_points%4].x0=x0;
    points[count_points%4].y0=y0;
    putpixel(x0,y0, COLOR(45,48,71));
    putpixel(x0+1,y0+1, COLOR(45,48,71));
    putpixel(x0-1,y0-1, COLOR(45,48,71));
    putpixel(x0+1,y0-1, COLOR(45,48,71));
    putpixel(x0-1,y0+1, COLOR(45,48,71));
    count_points++;
    delay(150);
}
```

```

}

void save()
{
    IMAGE *picture = createimage(WIDTH+1, HEIGHT+1);
    getimage(0,0,WIDTH,HEIGHT, picture);
    saveBMP("result.bmp",picture);
}

void chaikin()
{
    setcolor(COLOR(192,74,188));
    int dx, dy, j;
    int npoints=4;
    int nlines=3;
    point *A,*B;
    A = new point[npoints];
    for(int i=0;i<4;i++) A[i]=points[i];
    for(int i=0;i<6;i++)
    {
        B=new point[nlines*2];
        j = 0;
        for(int k=0;k<nlines;k++)
        {
            dx=A[k+1].x0-A[k].x0;
            dy=A[k+1].y0-A[k].y0;
            B[j].x0=A[k].x0+0.25*dx;
            B[j].y0=A[k].y0+0.25*dy;
            j++;
            B[j].x0=A[k].x0+0.75*dx;
            B[j].y0=A[k].y0+0.75*dy;
            j++;
        }
        delete[] A;
        A = B;
        npoints=2*nlines;
        nlines=npoints-1;
    }
    A[0]=points[0];
    A[nlines-1]=points[3];
    for(int i=0; i<nlines-1;i++)
    {
        line(A[i].x0,A[i].y0,A[i+1].x0,A[i+1].y0);
    }
    delete[] A;
}

void bez()
{
    int x,y;
    double t;
    for(t=0.0;t<=1.0;t+=0.0001)

```

```

    {
        x=(1-t)*(1-t)*(1-t)*points[0].x0+3*(1-t)*(1-t)*t*points[1].x0+3
        y=(1-t)*(1-t)*(1-t)*points[0].y0+3*(1-t)*(1-t)*t*points[1].y0+3
        putpixel(x,y,COLOR(45,48,71));
    }
}

void coordinats(int x, int y)
{
    string text;
    text = "X:" + to_string(x) + " Y:" + to_string(y);
    setbkcolor(COLOR(65,157,120));
    setcolor(COLOR(224,164,88));
    setfillstyle(SOLID_FILL, COLOR(65,157,120));
    bar(700,780,800,800);
    outtextxy(700,780, text.c_str());
}

```

Файл controls.h

```

#ifndef CONTROLS_H
#define CONTROLS_H
#define HEIGHT 800
#define WIDTH 800

enum CONTROLS {STOP, CHAIKIN, BEZ, SAVE, EXIT, FIELD, N_CONTROLS};

struct CONTROL
{
    int x0;
    int y0;
};

void create_controls(int, int, int);
void create_back(const char*);
int select_control();

#endif

```

Файл controls.cpp

```

#include <controls.h>
#include <graphics.h>

CONTROL controls[N_CONTROLS];

void create_controls(int x0, int y0, int n)
{
    controls[n].x0=x0;
    controls[n].y0=y0;
}

```



```

}

int select_control()
{
    int x, y;

    x = mousex();
    y = mousey();

    for (int i = 0; i < N_CONTROLS; i++)
    {
        if((x-controls[i].x0)*(x-controls[i].x0)+(y-controls[i].y0)*(y-
            return i;
        if((x-800)*274-(y-218)*(-302)<0)
            return FIELD;
    }
    return -1;
}

void create_back(const char *image_name)
{
    IMAGE *image=loadBMP(image_name);
    putimage(0,0,image,COPY_PUT);
    freeimage(image);
}

```

4 РЕЗУЛЬТАТ РАБОТЫ



Рисунок 4.1 – Результат выполнения программы



Рисунок 4.2 – Результат выполнения программы, розовая линия это алгоритм Чайкина, черная кривая Безье