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 № 3 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. Привести описание и схему алгоритма By для растрового представления линии.

2. Разработать подпрограмму для рисования линии (аналог процедуры line из графической библиотеки). Аргументы подпрограммы – координаты начальной и конечной точек. При реализации подпрограммы использовать для рисования только процедуру putpixel. Для определения текущего цвета рисования использовать функцию getcolor.

3. Разработать подпрограмму для рисования правильной звезды. Аргументы подпрограммы – координаты центра, радиус описанной окружности и число вершин. При создании контура звезды использовать собственную подпрограмму рисования линии. Для закраски фигуры использовать процедуру floodfill.

4. Написать программу для тестирования разработанных подпрограмм. Интерфейс программы должен содержать следующие элементы управления:

- увеличение/уменьшение числа вершин;
- увеличение/уменьшение размера (радиуса описанной окружности);
- сохранение результата в файл;
- выход из программы.

2 МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Пусть x_1, y_1, x_2, y_2 – соответственно координаты первой и второй точки. При рисовании линии цвет задан изначально col_0, col_1 и col_2 в формате RGB и будем говорить что он равет col, а координаты простовляемого пикселя x и y.

Рисование линии:

$$dx = |x_2 - x_1|.$$

$$dy = |y_2 - y_1|.$$

$$err = 0.$$

$$derr = 200.$$

Если dx не равен 0, то:

$$derr = 200 * dy/dx.$$

Если Изменение ошибки derr $<$ 200, то:

Если $x_1 > x_2$, то меняем точки местами.

$$y = y_1.$$

Если dy $= 0$ то и d $= 0$, иначе:

$$d = \frac{y_2 - y_1}{dy}.$$

Закрашиваем пиксель в начальной точке x_1, y_1 цвета col.

Потом перебераем все значения x от x_1 до x_2:

Ставим пиксель в координах x y и с цветом (200-err)*col/200 и в координах x y+d и с цветом err*col/200.

$$err = err + derr.$$

Если err $\geq$ 200:

$$y = y + d.$$

$$err = err - 200.$$

Если Изменение ошибки derr $\geq$ 200, то:

Если $y_1 > y_2$, то меняем точки местами.

$$x = x_1.$$

Если dx $= 0$ то и d $= 0$, иначе:

$$d = \frac{x_2 - x_1}{dx}.$$

Закрашиваем пиксель в начальной точке x_1, y_1 цвета col.

Потом переберaем все значения y от y_1 до y_2:

Ставим пиксель в координах x y и с цветом (200-err)*col/200 и в координах x+d y и с цветом err*col/200.

$$err = err + derr.$$

Если err ≥ 200:

$$x = x + d.$$

$$err = err - 200.$$

Файл main.cpp

```
#include <iostream>
#include "graphics.h"
#include <random>
#include <time.h>
#include <math.h>

#include "control.h"
#include "task.h"

using namespace std;

int main()
{
   setlocale(LC_ALL,"Russian");
   setlocale(LC_ALL,"rus");
   srand(time(0));
   initwindow(800, 600);

   setlinestyle(SOLID_LINE, 0, 1);
   setbkcolor(WHITE);

   srand(time(0));

   IMAGE *imager;
   imager = loadBMP("Red.bmp");

   IMAGE *imageg;
   imageg = loadBMP("Green.bmp");

   IMAGE *imageb;
   imageb = loadBMP("Blue.bmp");

   IMAGE *image;
   image = loadBMP("flag.bmp");

   int col[3][2] = {{200, 0}, {200, 0}, {200, 0}};
   int r = 100, wrt[4] = {0};
   int x = 400, y = 200, n = 5;
   int fil = 0, p = 0, t = 0;

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

      setfillstyle(SOLID_FILL, BLACK);
      setcolor(WHITE);
      rectangle(-1, -1, 801, 601);
```

```
bar(0, 0, 800, 600);
if(n==6) putimage(0, 0, image, COPY_PUT);

drawPol(x, y, n, r, col);

setlinestyle(SOLID_LINE, 0, 5);
setcolor(BLACK);
outtextxy(620, 440, "??????");
setcolor(WHITE);
rectangle(725, 425, 775, 475);
if(abs(mousex()-750) <= 25 and abs(mousey()-450) <= 25 and mous
{
   fil = 1-fil;
   while(mousebuttons());
}
if(fil)
{
   line(735, 435, 750, 450);
   line(750, 450, 780, 420);
   if(n > 4) fill(x, y, col);
}
setlinestyle(SOLID_LINE, 0, 1);

if(mousebuttons() && abs(95-mousex()) < 65 && abs(560-mousey())
{
   for(int i = 0; i < 4; i++) wrt[i] = 0;
   wrt[0] = 1;
}
if(mousebuttons() && abs(270-mousex()) < 85 && abs(560-mousey()
{
   for(int i = 0; i < 4; i++) wrt[i] = 0;
   wrt[1] = 1;
}
if(mousebuttons() && abs(470-mousex()) < 85 && abs(560-mousey()
{
   for(int i = 0; i < 4; i++) wrt[i] = 0;
   wrt[2] = 1;
}
if(mousebuttons() && abs(677-mousex()) < 92 && abs(560-mousey()
{
   for(int i = 0; i < 4; i++) wrt[i] = 0;
   wrt[3] = 1;
}

r = pole(30, 550, 125, 25, "???: ", r, wrt[0], t);
x = pole(190, 550, 165, 25, "X ????: ", x, wrt[1], t);
y = pole(390, 550, 165, 25, "Y ?????: ", y, wrt[2], t);
n = pole(590, 550, 180, 25, "??????????: ", n, wrt[3], t);

lineColor(7, col, 0, imager);
lineColor(270, col, 1, imageg);
lineColor(533, col, 2, imageb);
```

```
        if(butSave(650, 0)) save();

        t = t+1;
        t = t%8;

        setvisualpage(p);
        delay(10);
    }

    getch();
}
```

Файл task.h

```
#ifndef TASK_H
#define TASK_H

struct Point;

Point *addPoint(int, int, Point*);
void lineVu(int, int, int, int, int[3][2]);
void drawPol(int, int, int, int, int[3][2]);
void fill(int, int, int[3][2]);
void save();

#endif
```

Файл task.cpp

```
#include <math.h>
#include <iostream>
#include "graphics.h"
#include "task.h"

using namespace std;

struct Point
{
   int x, y;
   Point *next;
};

Point *addPoint(int x, int y, Point *one)
{
   if(one == NULL)
   {
      one = new Point;
      one->x = x;
      one->y = y;
      one->next = NULL;
      return one;
   }
   one->next = addPoint(x, y, one->next);
   return one;
}

void lineVu(int x1, int y1, int x2, int y2, int col[3][2])
{
   int dx = abs(x2-x1);
   int dy = abs(y2-y1);
   int err = 0;
   int derr = 200;
```

```
if(dx != 0) derr = 200*(dy)/(dx);
if(derr < 200)
{
   if(x1 > x2)
   {
      int a = x1;
      x1 = x2;
      x2 = a;
      a = y1;
      y1 = y2;
      y2 = a;
   }
   int y = y1;
   int d = dy == 0 ? 0 : (y2-y1)/dy;
   putpixel(x1, y1, COLOR(col[0][0], col[1][0], col[2][0]));
   for(int x = x1; x <= x2; x++)
   {
      putpixel(x, y, COLOR((200-err)*col[0][0]/200, (200-err)*col[
      if(err>2) putpixel(x, y+d, COLOR(err*col[0][0]/200, err*col[
      err = err+derr;
      if(err >= 200)
      {
         y += d;
         err -= 200;
      }
   }
   putpixel(x2, y2, COLOR(col[0][0], col[1][0], col[2][0]));
}
else
{
   if(y1 > y2)
   {
      int a = x1;
      x1 = x2;
      x2 = a;
      a = y1;
      y1 = y2;
      y2 = a;
   }
   derr = 40000/derr;
   int x = x1;
   int d = dx == 0 ? 0 : (x2-x1)/dx;
   putpixel(x1, y1, COLOR(col[0][0], col[1][0], col[2][0]));
   for(int y = y1; y <= y2; y++)
   {
      putpixel(x, y, COLOR((200-err)*col[0][0]/200, (200-err)*col[
      if(err>2) putpixel(x, y+d, COLOR(err*col[0][0]/200, err*col[
      err = err+derr;
      if(err >= 200)
      {
         x += d;
         err -= 200;
```

```
            }
        }
        putpixel(x2, y2, COLOR(col[0][0], col[1][0], col[2][0]));
    }
}

void drawPol(int x, int y, int n, int r, int col[3][2])
{
    Point *one = NULL;

    for(int i = 0; i < n*2+1; i++)
    {
        int yp, xp;
        if(i%2)
        {
            xp = x+int(r*cos(i*3.14*1/n));
            yp = y+int(r*sin(i*3.14*1/n));
        }
        else
        {
            int m = (n-1)/2;
            m = m<2 ? 2 : m;
            xp = x+int(r*(cos(3.14/n*m)/cos(3.14/n*(m-1)))*cos(i*3.14*1/
            yp = y+int(r*(cos(3.14/n*m)/cos(3.14/n*(m-1)))*sin(i*3.14*1/
        }
        one = addPoint(xp, yp, one);
    }

    Point *two = one->next;
    while(two != NULL)
    {
        lineVu(one->x, one->y, two->x, two->y, col);
        one = two;
        two = one->next;
    }
}

void fill(int x, int y, int col[3][2])
{
    if(getpixel(x, y) == 0)
    {
        putpixel(x, y, COLOR(col[0][0], col[1][0], col[2][0]));
        fill(x+1, y, col);
        fill(x-1, y, col);
        fill(x, y+1, col);
        fill(x, y-1, col);
    }
}

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);
}
```

Файл control.h

```
#ifndef CONTROL_H
#define CONTROL_H

int pole(int, int, int, int, std::string, int, int&, int);
void lineColor(int, int[3][2], int, IMAGE*);
bool butSave(int, int);

#endif
```

Файл control.cpp

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

#include "control.h"

using namespace std;

int pole(int left, int top, int widht, int height, string Text, int d
{
    if(wrt)
    {
        if(kbhit())
        {
            int key = getch();
            if(key == 13) wrt = 0;
            if(key == 8) data = data/10;
            if(key > 47 && key < 58) data = data*10+key-48;
        }

        setfillstyle(SOLID_FILL, WHITE);
        bar(left-5, top-5, left+widht, top+height);
        setcolor(BLACK);
        if(t > 3)
        {
            char text[30];
            sprintf(text, "%s%d", Text.c_str(), data);
```

```
            outtextxy(left, top, text);
        }
        else
        {
            char text[30];
            sprintf(text, "%s%d|", Text.c_str(), data);
            outtextxy(left, top, text);
        }
    }
    else
    {
        setfillstyle(SOLID_FILL, WHITE);
        bar(left-5, top-5, left+widht, top+height);
        setcolor(BLACK);
        char text[30];
        sprintf(text, "%s%d", Text.c_str(), data);
        outtextxy(left, top, text);
    }
    return data;
}

void lineColor(int x, int col[3][2], int num, IMAGE *image)
{
    putimage(x, 500, image, COPY_PUT);

    setfillstyle(SOLID_FILL, WHITE);
    bar(col[num][0]+x-3, 500, col[num][0]+x+3, 529);


    if(mousebuttons())
    {
        if(abs(mousex()-col[num][0]-x) < 4 and abs(mousey() - 515) < 15
        {
            col[num][1] = 1;
        }
    }
    else
    {
        col[num][1] = 0;
    }

    if(col[num][1])
    {
        col[num][0] = mousex()-x;
        col[num][0] = (col[num][0]+abs(col[num][0]))/2;
        col[num][0] = col[num][0]>255 ? 255 : col[num][0];
    }
}

bool butSave(int left, int top)
{
    setcolor(WHITE);
```

```
    bar(left, top, left+150, top+50);
    setcolor(BLACK);
    outtextxy(left+30, top+15, "????/");
    return abs(mousex()-left-75) < 75 && abs(mousey()-top-25) < 25 &&
}
```

4 РЕЗУЛЬТАТ РАБОТЫ

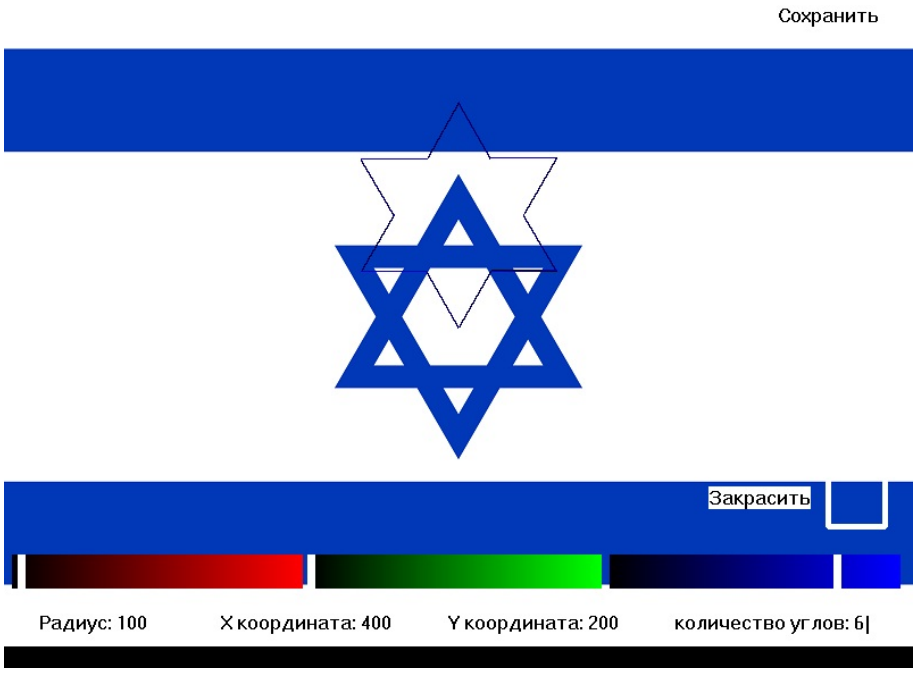

Рисунок 4.1 – Результат выполнения программы (Пример 1)

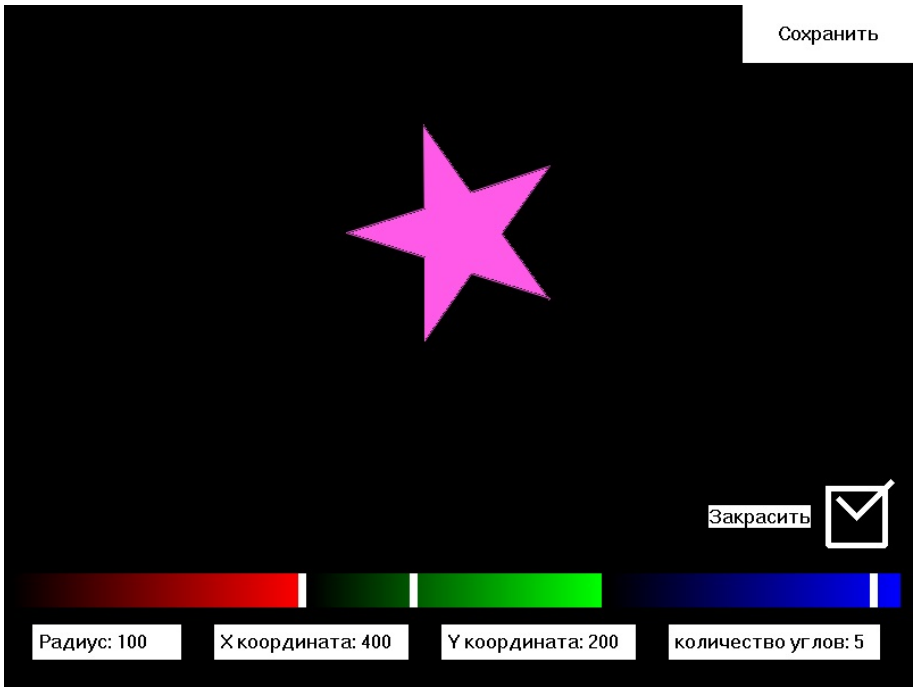

Рисунок 4.2 – Результат выполнения программы (Пример 2)

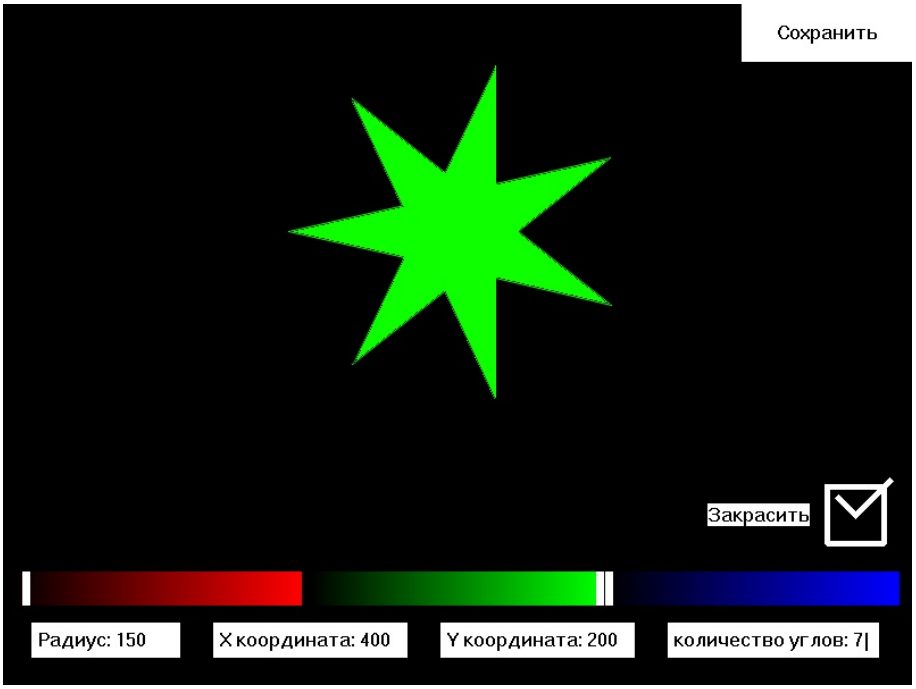

Рисунок 4.3 – Результат выполнения программы (Пример 3)