

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 № 2 по дисциплине
«Математические основы компьютерной графики»

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

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

Челябинск 2022

1 ЗАДАНИЕ

1. Написать программу для построения пересечения двух прямоугольников. Предполагается, что стороны прямоугольников параллельны координатным осям. Для задания положения и размеров прямоугольников использовать генератор псевдослучайных чисел. Интерфейс программы должен содержать следующие элементы управления:

- создание фигур;
- построение решения;
- сохранение результата в файл;
- выход из программы.

2 МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Пусть x_0, y_0, w, h – соответственно координаты левого верхнего угла, ширина и высота прямоугольной области. При генерации квадратов мы выбираем координаты в диапазоне $x_0+5 < x < w-5$ и $y_0+5 < y < h-5$. $sq1x1, sq1y1, sq1x2, sq1y2$ координаты первого квадрата и $sq2x1, sq2y1, sq2x2, sq2y2$ координаты второго квадрата.

чтобы найти пересечение прямоугольников мы находим его верхний левый $x_1 y_1$ и нижний правый угол $x_2 y_2$:

$$x_1 = \begin{cases} sq1x1 & sq1x1 > sq2x1 \\ sq2x1 & sq1x1 \leq sq2x1 \end{cases}$$

$$x_2 = \begin{cases} sq1x2 & sq1x2 \leq sq2x2 \\ sq2x2 & sq1x2 > sq2x2 \end{cases}$$

$$y_1 = \begin{cases} sq1y1 & sq1y1 > sq2y1 \\ sq2y1 & sq1y1 \leq sq2y1 \end{cases}$$

$$y_2 = \begin{cases} sq1y2 & sq1y2 \leq sq2y2 \\ sq2y2 & sq1y2 > sq2y2 \end{cases}$$

Если $x_2 < x_1$ или $y_2 < y_1$ то квадраты не пересекаются

Иначе закрашиваем квадрат с координатами $x_1 y_1 x_2 y_2$

3 ТЕКСТ ПРОГРАММЫ

Файл main.cpp

```
#include "graphics.h"
#include "control.h"
#include "task.h"

int main()
{
    initwindow(600, 450);

    create_control(FILL_1, 0, 400, "start.bmp");
    create_control(FILL_2, 150, 400, "fill.bmp");
    create_control(SAVE, 300, 400, "save.bmp");
    create_control(EXIT, 450, 400, "exit.bmp");

    IMAGE *image;

    image = loadBMP("fon.bmp");
    putimage(0, 0, image, COPY_PUT);

    freeimage(image);

    int left = 0, top = 0, width = 600, height = 400;

    while (true)
    {
        while (mousebuttons() != 1);
        switch (select_control())
        {
            case NONE: break;
            case FILL_1: genPoint(left, top, width, height); break;
            case FILL_2: recPoint(left, top, width, height); break;
            case SAVE: save(); break;
            case EXIT: closegraph(); return 0;
        }
        while(mousebuttons() == 1);
    }
}
```

Файл task.h

```
#ifndef TASK_H
#define TASK_H

struct Square
{
    int x1, y1, x2, y2;
};

void genPoint(int, int, int, int);
void recPoint(int, int, int, int);
void save();

#endif
```

Файл task.cpp

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

Square sq[2];

void genPoint(int left, int top, int width, int height)
{
    srand(time(0));

    IMAGE *image;

    image = loadBMP("fon.bmp");
    putimage(0, 0, image, COPY_PUT);

    freeimage(image);

    setcolor(COLOR(255, 0, 0));
    for(int i = 0; i < 2; i++)
    {
        sq[i].x1 = rand()%(width-left-10)+5;
        sq[i].x2 = rand()%(width-left-10)+5;
        sq[i].y1 = rand()%(height-top-10)+5;
        sq[i].y2 = rand()%(height-top-10)+5;
        if(sq[i].x1 > sq[i].x2)
        {
            int x = sq[i].x1;
            sq[i].x1 = sq[i].x2;
            sq[i].x2 = x;
        }
    }
}
```

```

        if(sq[i].y1 > sq[i].y2)
        {
            int y = sq[i].y1;
            sq[i].y1 = sq[i].y2;
            sq[i].y2 = y;
        }
        rectangle(sq[i].x1, sq[i].y1, sq[i].x2, sq[i].y2);
    }
}

void recPoint(int left, int top, int width, int height)
{
    int x1 = fmax(sq[0].x1, sq[1].x1);
    int x2 = fmin(sq[0].x2, sq[1].x2);
    int y1 = fmax(sq[0].y1, sq[1].y1);
    int y2 = fmin(sq[0].y2, sq[1].y2);
    if(x2-x1 <= 0 || y2-y1 <= 0) return;
    setfillstyle(SOLID_FILL, COLOR(255, 0, 0));
    floodfill((x2-x1)/2+x1, (y2-y1)/2+y1, COLOR(255, 0, 0));
}

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

enum control_values { NONE = -1, EXIT, SAVE,
                     FILL_1, FILL_2, FILL_3, FILL_4,
                     FILL_5, FILL_6, FILL_7, FILL_8,
                     N_CONTROLS };

struct Control
{
    int left;
    int top;
    int right;
}

```

```

        int bottom;
};

void create_control(int, int, int, const char*);
int select_control();

#endif

```

Файл control.cpp

```

#include "graphics.h"
#include "control.h"

Control controls[N_CONTROLS];

void create_control(int i, int left, int top,
                   const char *file_name)
{
    IMAGE *image;

    image = loadBMP(file_name);
    putimage(left, top, image, COPY_PUT);

    controls[i].left    = left;
    controls[i].top     = top;
    controls[i].right   = left + imagewidth(image) - 1;
    controls[i].bottom  = top  + imageheight(image) - 1;

    freeimage(image);
}

int select_control()
{
    int x, y;

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

    for (int i = 0; i < N_CONTROLS; i++)
    {
        if (x > controls[i].left && x < controls[i].right &&
            y > controls[i].top  && y < controls[i].bottom)
        {
            return i;
        }
    }

    return NONE;
}

```

4 РЕЗУЛЬТАТ РАБОТЫ

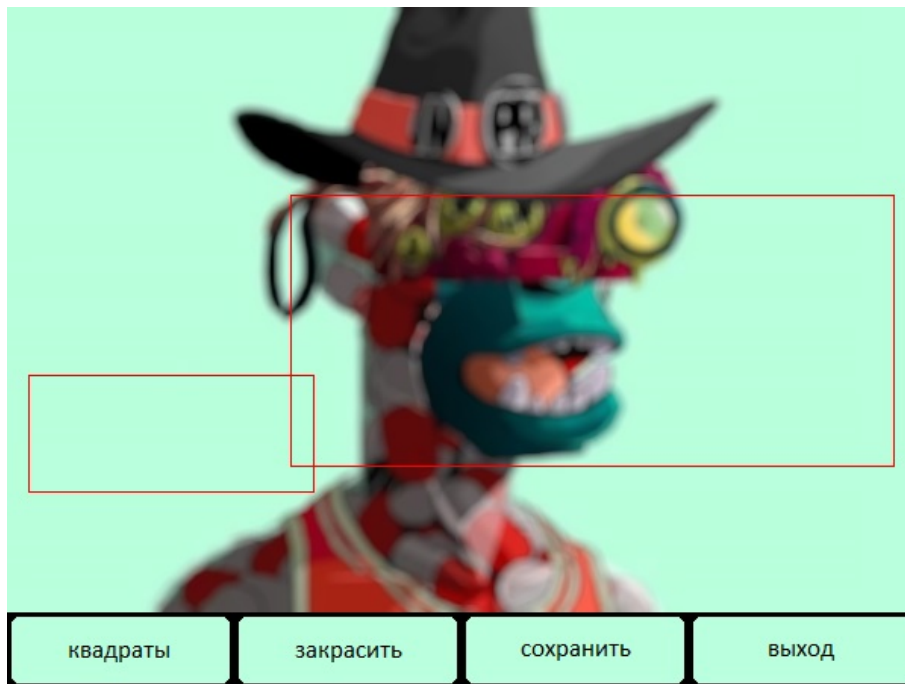


Рисунок 4.1 – Результат выполнения программы (функция `creatPoint`, кнопка "Установить положение друзей")

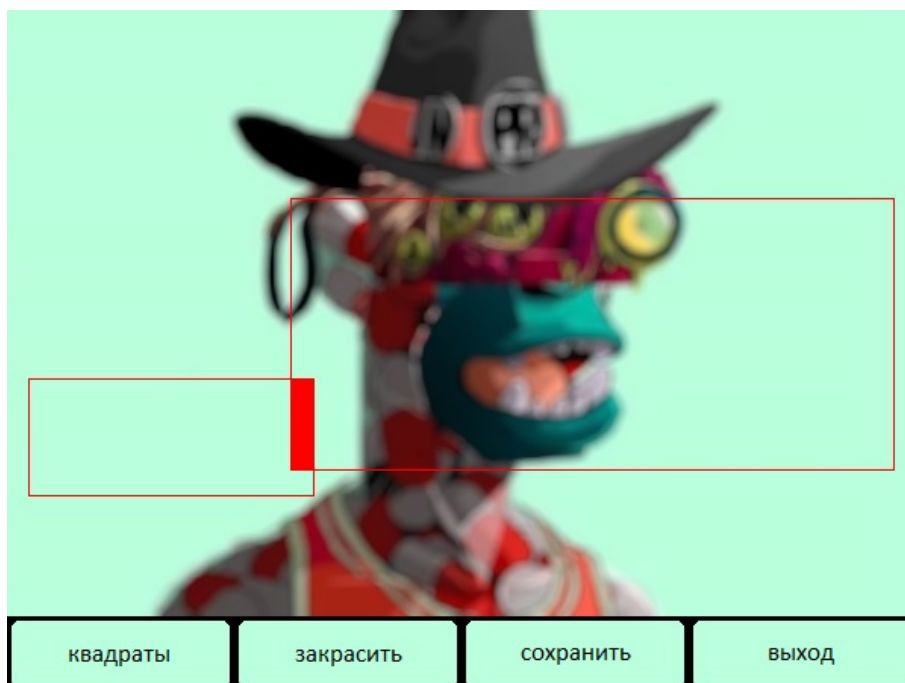


Рисунок 4.2 – Результат выполнения программы (функция `treat`, кнопка "Найти дальних друзей")