

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
_____ Малахов В.А.
« ____ » _____ 2022 г.

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

Челябинск 2022

1 ЗАДАНИЕ

1. Написать программу расставления случайных точек и отрисовки квадрата с минимальной площадью в который будут входить все точки. Для задания координат точек использовать генератор псевдослучайных чисел. Интерфейс программы должен содержать следующие элементы управления:

- создание множества точек;
- 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 точек мы выбираем координаты в диапазоне $x_0+15 < x < w-15$ и $y_0+15 < y < h-15$, где x, y координаты точки.

Чтобы найти квадрата с минимальной площадью в который будут входить все точки мы перебираем все точки и находим $x_{min}, x_{max}, y_{min}$ и y_{max} :

Если выбранная точка имеет x или y который больше наших x_{max}, y_{max} или меньше x_{min}, y_{min} , то мы мы записываем подходящие значения.

Потом рисуем квадрат с координатами $x_{min}, y_{min}, x_{max}$ и y_{max} .

3 ТЕКСТ ПРОГРАММЫ

Файл main.cpp

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

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

    setfillstyle(SOLID_FILL, COLOR(155, 255, 255));
    bar(0, 0, 600, 400);

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

    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

#define NP 10

struct Point
{
    int x, y;
};

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"

Point points[NP];

void genPoint(int left, int top, int width, int height)
{
    srand(time(0));
    setfillstyle(SOLID_FILL, COLOR(100, 255, 255));
    bar(left, top, width, height);

    IMAGE *image;
    image = loadBMP("iglu.bmp");
    for(int i = 0; i < NP; i++)
    {
        points[i].x = rand()%(width-30)+15;
        points[i].y = rand()%(height-30)+15;

        putimage(points[i].x-15, points[i].y-15, image, XOR_PUT);
    }
    freeimage(image);
}

void recPoint(int left, int top, int width, int height)
{
    int xMin = width, yMin = height, xMax = left, yMax = top;
```

```

    for(int i = 0; i < NP; i++)
    {
        if(points[i].x > xMax){
            xMax = points[i].x;
        }
        if(points[i].y > yMax){
            yMax = points[i].y;
        }
        if(points[i].x < xMin){
            xMin = points[i].x;
        }
        if(points[i].y < yMin){
            yMin = points[i].y;
        }
    }
    setcolor(BLACK);
    rectangle(xMin-15, yMin-15, xMax+15, yMax+15);
}

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

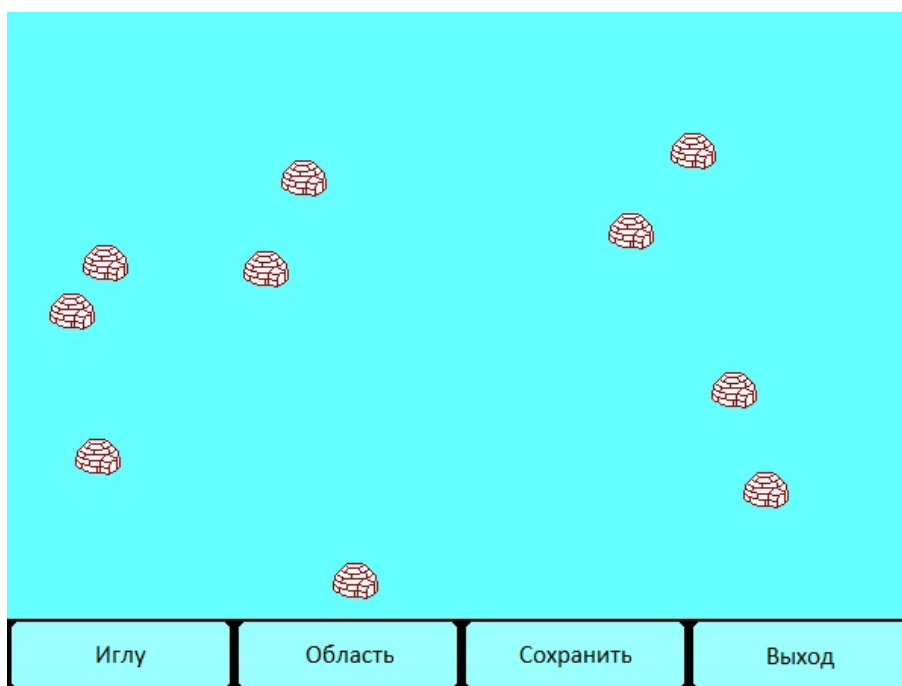


Рисунок 4.1 – Результат выполнения программы (функция `creatPoint`, кнопка "Установить положение иглу")

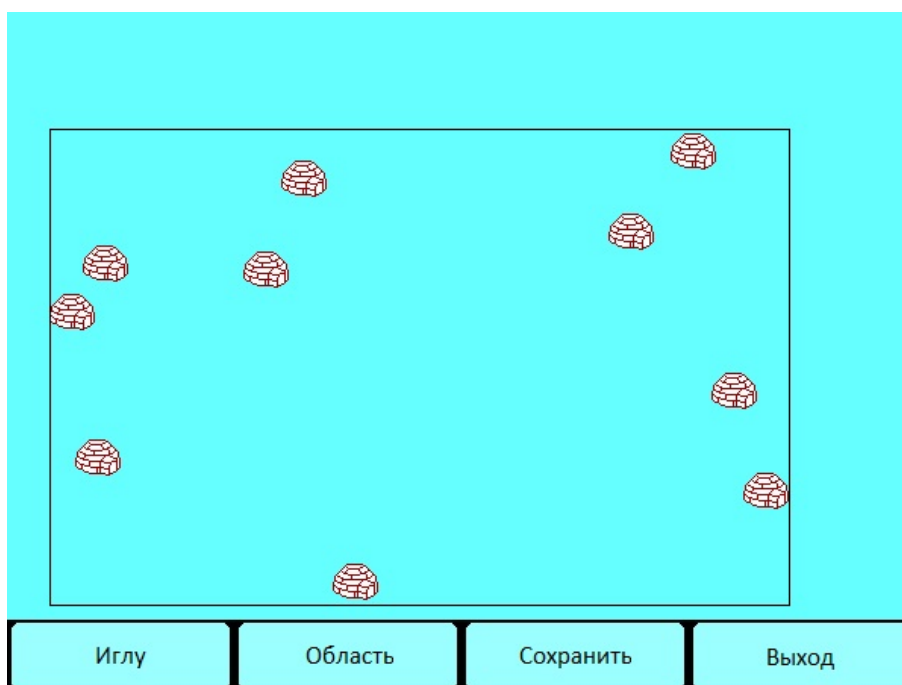


Рисунок 4.2 – Результат выполнения программы (функция `treat`, кнопка "Отрисовка области")