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-212
_____ Гаврилова А.Ю.
«____» _____ 2022 г.

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

Челябинск 2022

1 ЗАДАНИЕ

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

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

2 МАТЕМАТИЧЕСКАЯ МОДЕЛЬ

Координаты x и y – псевдослучайные числа в множестве точек $A_i(x_i,y_i)$, при i=1,...n

При это заданы координаты первого прямоугольника:(x_{min},y_{min}),(x_{max},y_{max}), и второго прямоугольника: $(fillbar.x0, fillbar.y0)$,(x,y)

Расположение прямоугольников на поле выбирается случайным образом, путем построения 4 точек.

Далее, определяется область пересечения прямоугольников проверкой нахождения выбранных точек в области другого прямоугольника. Закраска искомой области происходит по точкам:

$(fillbar.x0, fillbar.y0)$, $(fillbar.x1, fillbar.y1)$

3 ТЕКСТ ПРОГРАММЫ

Файл main.cpp

```
#include <cstdlib>
#include <time.h>
#include "graphics.h"
#include "control.h"
#include "task.h"

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

   create_control(CREATE_POINTS, 0, 360, "create.bmp");
   create_control(SOLUTION, 150, 360, "solution1.bmp");
   create_control(SAVE, 300, 360, "save1.bmp");
   create_control(EXIT, 450, 360, "EXIT1.bmp");
   bool sw = true;

   int left = 0, top = 0, width = 600, height = 400;
   srand(time(0));

   while(true){
      while(mousebuttons() != 1);
      switch(select_control()){
         case NONE:
            break;
         case CREATE_POINTS:
            if(sw){
               create_rectangles(width, height);
               sw = false;
            }
            break;
         case SOLUTION:
            solution();
            break;
         case SAVE:
            save();
            break;
         case EXIT:
            closegraph();
            return 0;
      }
   }
}
```

Файл task.h

```
#ifndef TASK_H // define TASK_H

struct fillbar{
   int x0, y0;
   int x1, y1;
};  //

void create_rectangles(int, int);
void solution();
void save();

#endif
```

Файл task.cpp

```
#include <math.h>
#include "graphics.h"
#include <cstdlib>
#include "task.h"
fillbar fillbar;
void create_rectangles(int width, int height)
{
   setcolor(YELLOW);
   setfillstyle(SOLID_FILL, YELLOW);
   setcolor(BLUE);
   int x, y, x_min, x_max, y_min,  y_max;

   //--
   x_min = rand() % (width - 100);
   x_max = rand() % (width - x_min - 50) + x_min + 50;
   y_min = rand() % (height - 100);
   y_max = rand() % (height - y_min - 50) + y_min + 50;
   rectangle(x_min, y_min, x_max, y_max);

   //--
   //   fillbar.x0 = rand() % (x_max - x_min) + x_min;
   fillbar.y0 = rand() % (y_max - y_min) + y_min;
   //  x = rand() % width;
   y = rand() % height;
   rectangle(fillbar.x0, fillbar.y0, x, y);

   //--    if(x > x_max)
      fillbar.x1 = x_max;
   else if(x < x_min)
      fillbar.x1 = x_min;
   else
      fillbar.x1 = x;
```

```
    if(y > y_max)
        fillbar.y1 = y_max;
    else if(y < y_min)
        fillbar.y1 = y_min;
    else
        fillbar.y1 = y;
}

void solution()
{
    setcolor(RED);
    int temp;
    if(fillbar.x0 > fillbar.x1)
    {
        temp = fillbar.x0;
        fillbar.x0 = fillbar.x1;
        fillbar.x1 = temp;
    }
    if(fillbar.y0 > fillbar.y1)
    {
        temp = fillbar.y0;
        fillbar.y0 = fillbar.y1;
        fillbar.y1 = temp;
    }
    setfillstyle(SOLID_FILL, BLUE);
    bar(fillbar.x0, fillbar.y0,fillbar.x1, fillbar.y1);
}

void save()
{
    int w, h;
    IMAGE *output;

    w = getmaxx() + 1;
    h = getmaxy() + 1;
    output = createimage(w, h);

    getimage(0, 0, w - 1, h - 1, output);
    saveBMP("output1.bmp", output);
    freeimage(output);
}
```

Файл control.h

```
#ifndef CONTROL_H
#define CONTROL_H

enum control_values { NONE = -1, SAVE, EXIT,
                      CREATE_POINTS, SOLUTION,
                      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;
}
```

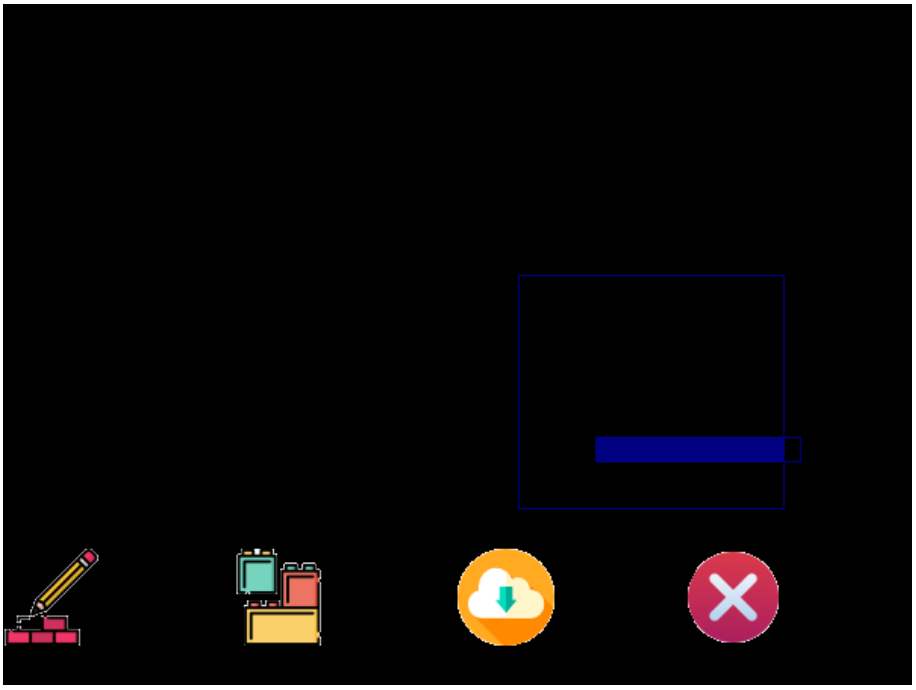

Рисунок 4.1 – Результат выполнения программы

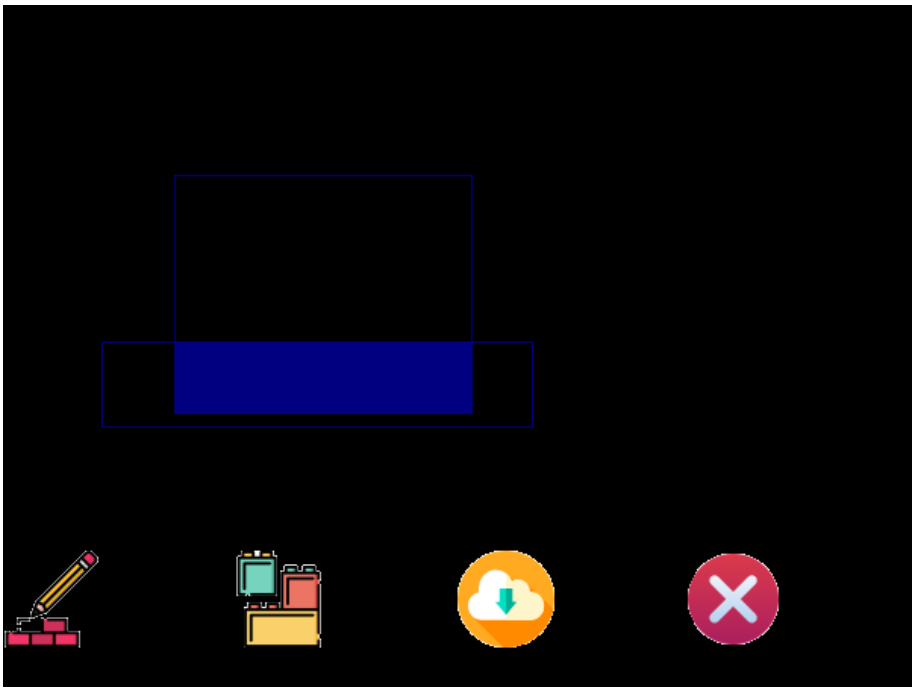

Рисунок 4.2 – Результат выполнения программы