МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



**ОТЧЁТ**

**по лабораторной работе № 3**

**«Заполнение треугольников*»***

по дисциплине:

**«Компьютерная графика»**

|  |  |
| --- | --- |
| Выполнил(а):  Дашкин С.М. Группа АТ-34 | Проверил:  Смагин С.М. |

Новосибирск

2024

**ЦЕЛЬ РАБОТЫ**

Изучить алгоритмы заполнения треугольника

**Задание**

1. Разработать программу, которая закрашивает треугольник по алгоритму заполнения треугольника.

**ХОД РАБОТЫ:**

1. **Функция generateGrid**

* generateGrid: Функция для создания сетки.
* HDC hdc: Контекст устройства для рисования.
* numColumns, numRows, cellSize: Количество столбцов, строк и размер ячейки сетки.
* Функция запрашивает у пользователя размер сетки и рисует вертикальные и горизонтальные линии, создавая сетку.

2. **Функция sorttreug**

* sorttreug: Функция для закрашивания треугольника.
* Ax, Ay, Bx, By, Cx, Cy: Координаты вершин треугольника.
* Функция сортирует вершины треугольника по координате Y и закрашивает его, используя алгоритм сканирования линий.

3. **Функция drawTriangle**

* drawTriangle: Функция для ввода координат вершин треугольника и его закрашивания.
* HDC hdc: Контекст устройства для рисования.
* int numColumns, int numRows, int cellSize: Количество столбцов, строк и размер ячейки.
* Пользователь вводит координаты вершин треугольника.
* Вызывает sorttreug для закрашивания треугольника.

4. **Основная функция main:**

* setlocale(LC\_ALL, "Rus"): Устанавливает локаль для русского языка.
* HWND hwnd = GetConsoleWindow(): Получает дескриптор окна консоли.
* HDC hdc = GetDC(hwnd): Получает контекст устройства для рисования.
* HPEN pen и HBRUSH brush: Создает перо и кисть для рисования.
* SelectObject(hdc, pen) и SelectObject(hdc, brush): Выбирает перо и кисть для рисования.
* int numColumns, numRows, cellSize = 3: Инициализирует переменные для количества столбцов, строк и размера ячейки.
* generateGrid(hdc, numColumns, numRows, cellSize): Вызывает функцию для создания сетки.
* drawTriangle(hdc, numColumns, numRows, cellSize): Вызывает функцию для рисования треугольника.
* DeleteObject(pen) и DeleteObject(brush): Удаляет перо и кисть.
* ReleaseDC(hwnd, hdc): Освобождает контекст устройства.
* int i; std::cin >> i: Ожидает ввода пользователя для завершения программы.

**ЛИСТИНГ:**

#include <iostream>

#include <windows.h>

#include <windowsx.h>

#include <locale.h>

#define Y\_OFFSET 150

#define X\_OFFSET 400

void generateGrid(HDC hdc, int& numColumns, int& numRows, int& cellSize) {

int i; std::cout << "Введите размер сетки(x y): ";

std::cin >> numColumns >> numRows;

cellSize \*= 100 / (float)numRows;

for (i = 1; i < numColumns + 1; i++) {

MoveToEx(hdc, X\_OFFSET + i \* cellSize, Y\_OFFSET, NULL);

LineTo(hdc, X\_OFFSET + i \* cellSize, Y\_OFFSET + numRows \* cellSize);

}

for (i = 0; i < numRows + 1; i++) {

MoveToEx(hdc, X\_OFFSET, Y\_OFFSET + i \* cellSize, NULL);

LineTo(hdc, X\_OFFSET + numColumns \* cellSize, Y\_OFFSET + i \* cellSize);

}

}

void sorttreug(HDC hdc, int Ax, int Ay, int Bx, int By, int Cx, int Cy, int cellSize, int numRows) {

int tmp;

int X[3] = { Ax, Bx, Cx };

int Y[3] = { Ay, By, Cy };

for (int i = 0; i < 2; i++)

for (int j = 0; j < 2; j++)

if (Y[j] < Y[j + 1]) {

tmp = Y[j + 1];

Y[j + 1] = Y[j];

Y[j] = tmp;

tmp = X[j + 1];

X[j + 1] = X[j];

X[j] = tmp;

}

Cy = Y[0]; By = Y[1]; Ay = Y[2];

Cx = X[0]; Bx = X[1]; Ax = X[2];

int x1, x2;

for (int sy = Ay; sy <= Cy; sy++) {

x1 = Ax + (sy - Ay) \* (Cx - Ax) / (Cy - Ay);

if (sy < By)

x2 = Ax + (sy - Ay) \* (Bx - Ax) / (By - Ay);

else {

if (Cy == By)

x2 = Bx;

else

x2 = Bx + (sy - By) \* (Cx - Bx) / (Cy - By);

}

if (x1 > x2) {

tmp = x1;

x1 = x2;

x2 = tmp;

}

for (int x = x1; x <= x2; x++) {

Rectangle(hdc, X\_OFFSET + (x - 1) \* cellSize, Y\_OFFSET + (numRows - sy) \* cellSize, X\_OFFSET + x \* cellSize, Y\_OFFSET + (numRows - sy + 1) \* cellSize);

}

}

}

void drawTriangle(HDC hdc, int numColumns, int numRows, int cellSize) {

int point1[2], point2[2], point3[2];

std::cout << "Введите координаты первой точки(x y): ";

std::cin >> point1[0] >> point1[1];

std::cout << "Введите координаты второй точки(x y): ";

std::cin >> point2[0] >> point2[1];

std::cout << "Введите координаты третьей точки(x y): ";

std::cin >> point3[0] >> point3[1];

sorttreug(hdc, point1[0], point1[1], point2[0], point2[1], point3[0], point3[1], cellSize, numRows);

}

int main()

{

setlocale(LC\_ALL, "Rus");

HWND hwnd = GetConsoleWindow();

HDC hdc = GetDC(hwnd);

HPEN pen = CreatePen(PS\_SOLID, 1, RGB(0, 0, 255));

HBRUSH brush = CreateSolidBrush(RGB(255, 0, 0));

SelectObject(hdc, pen);

SelectObject(hdc, brush);

int numColumns, numRows, cellSize = 3;

generateGrid(hdc, numColumns, numRows, cellSize);

drawTriangle(hdc, numColumns, numRows, cellSize);

DeleteObject(pen);

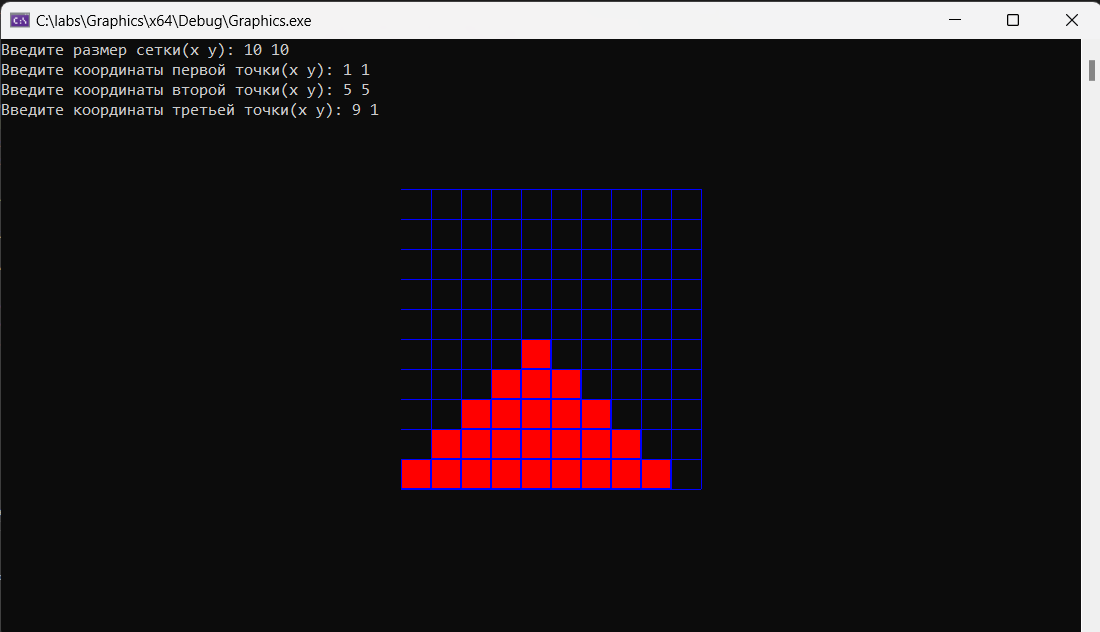
DeleteObject(brush);

ReleaseDC(hwnd, hdc);

int i; std::cin >> i;

}

**ТЕСТИРОВАНИЕ:**

****