Пензенский государственный университет

Кафедра «Вычислительная техника»

**ОТЧЕТ**

по лабораторной работе №1

по дисциплине:"Инженерная и компьютерная графика"

на тему: "ВЕКТОРНОЕ ПРЕДСТАВЛЕНИЕ ГРАФИЧЕСКИХ ОБЪЕКТОВ АФФИННЫЕ ПРЕОБРАЗОВАНИЯ"

**Выполнили** **студенты группы 23ВВВ3 бригады №5:**

Полиневский Вадим

Кизым Иван

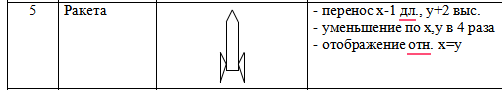
**Приняли:**

                                                         Финогоев А.А. Финогоев А.Г.

**Пенза 2025**

**Цель работы**: Изучение способов представления и визуализации растровых изображений, создание динамических изображений средствами спрайтовой анимации.

**Вариант задания:**



**Листинг *WindowsProject1*:**

**…**

LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)

{

switch (message)

{

case WM\_COMMAND:

{

int wmId = LOWORD(wParam);

switch (wmId)

{

case IDM\_ABOUT:

DialogBox(hInst, MAKEINTRESOURCE(IDD\_ABOUTBOX), hWnd, About);

break;

case IDM\_EXIT:

DestroyWindow(hWnd);

break;

default:

return DefWindowProc(hWnd, message, wParam, lParam);

}

}

break;

case WM\_PAINT:

{

PAINTSTRUCT ps;

HDC hdc = BeginPaint(hWnd, &ps);

show\_images(hdc);

movements\_images(200, 0);

movements\_images(-1, 2);

show\_images(hdc);

decrease\_images(4);

movements\_images(450, -400);

show\_images(hdc);

mapping\_X\_Y();

movements\_images(200, 0);

show\_images(hdc);

EndPaint(hWnd, &ps);

}

break;

case WM\_DESTROY:

PostQuitMessage(0);

break;

default:

return DefWindowProc(hWnd, message, wParam, lParam);

}

return 0;

}

**…**

**Листинг заголовочного файла *header.h* :**

#pragma once

#include "framework.h"

#include <vector>

using namespace std;

vector <int> vector\_X\_Points = { 100, 150, 150, 50, 50, 100, 50, 10, 10, 50, 150, 190, 190, 150};

vector <int> vector\_Y\_Points = { 0, 50 , 550, 550, 50, 0, 500, 600, 400, 500, 500, 600, 400, 500};

void movements\_images(int \_X\_movements, int \_Y\_movements) {

for (int i = 0; i < vector\_X\_Points.size(); i++) { vector\_X\_Points[i] = vector\_X\_Points[i] + \_X\_movements; }

for (int i = 0; i < vector\_X\_Points.size(); i++) { vector\_Y\_Points[i] = vector\_Y\_Points[i] - \_Y\_movements; }

}

void decrease\_images(int ratio) {

for (int i = 0; i < vector\_X\_Points.size(); i++) { vector\_X\_Points[i] = vector\_X\_Points[i] / ratio; }

for (int i = 0; i < vector\_X\_Points.size(); i++) { vector\_Y\_Points[i] = vector\_Y\_Points[i] / ratio; }

}

void mapping\_X\_Y() {

vector <int> temp\_vector = vector\_X\_Points;

for (int i = 0; i < vector\_X\_Points.size(); i++) {

vector\_X\_Points[i] = vector\_Y\_Points[i];

vector\_Y\_Points[i] = temp\_vector[i];

}

}

void show\_images(HDC \_hdc) {

for (int i = 0; i < vector\_X\_Points.size(); i++) {

if (i == 0 || i == 6 || i == 10) {

MoveToEx(\_hdc, vector\_X\_Points[i], vector\_Y\_Points[i], NULL);

continue;

}

LineTo(\_hdc, vector\_X\_Points[i], vector\_Y\_Points[i]);

}

}

**Модель графического объекта в виде описания координат точек и соединяющих их отрезков**

LineTo(hdc, 150, 50);

LineTo(hdc, 150, 550);

LineTo(hdc, 50, 550);

LineTo(hdc, 50, 50);

LineTo(hdc, 100, 0);

MoveToEx(hdc, 50, 500, NULL);

LineTo(hdc, 10, 600);

LineTo(hdc, 10, 400);

LineTo(hdc, 50, 500);

MoveToEx(hdc, 150, 500, NULL);

LineTo(hdc, 190, 600);

LineTo(hdc, 190, 400);

LineTo(hdc, 150, 500);

**Аффинные преобразования**

**Сдвиг изображения:**

void movements\_images(int \_X\_movements, int \_Y\_movements) {

for (int i = 0; i < vector\_X\_Points.size(); i++) { vector\_X\_Points[i] = vector\_X\_Points[i] + \_X\_movements; }

for (int i = 0; i < vector\_X\_Points.size(); i++) { vector\_Y\_Points[i] = vector\_Y\_Points[i] - \_Y\_movements; }

}

**Уменьшение по x, y в *int ratio* раз**

void decrease\_images(int ratio) {

for (int i = 0; i < vector\_X\_Points.size(); i++) { vector\_X\_Points[i] = vector\_X\_Points[i] / ratio; }

for (int i = 0; i < vector\_X\_Points.size(); i++) { vector\_Y\_Points[i] = vector\_Y\_Points[i] / ratio; }

}

**Отображение относительно x = y;**

void mapping\_X\_Y() {

vector <int> temp\_vector = vector\_X\_Points;

for (int i = 0; i < vector\_X\_Points.size(); i++) {

vector\_X\_Points[i] = vector\_Y\_Points[i];

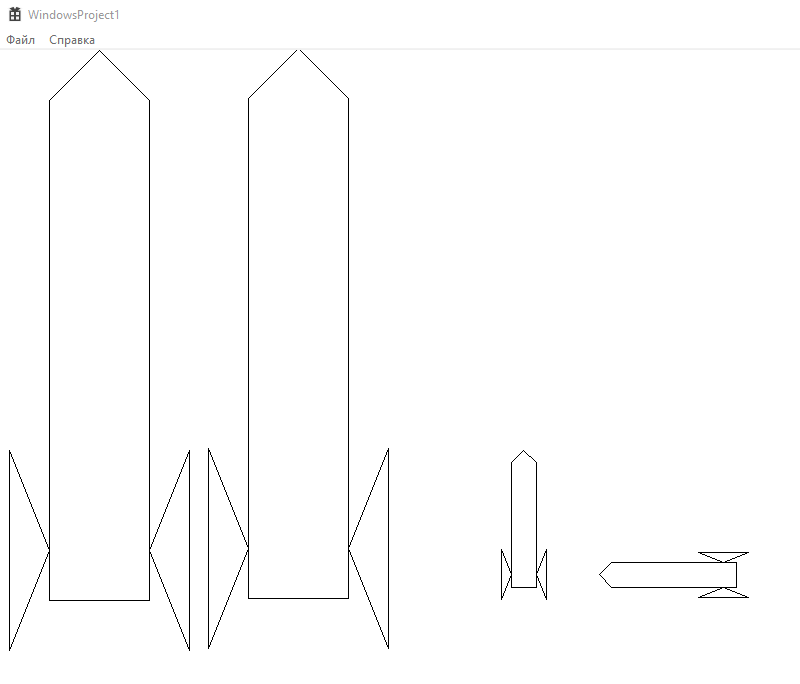
vector\_Y\_Points[i] = temp\_vector[i];

}

}

**Распечатка с изображением графического объекта всех выполненных аффинных преобразований**

До преобразований слева.



**Вывод:**

В результате выполнения лабораторной работы получили навыки разработки и отладки программ с помощью средств среды Visual Studio и Windows GDI.. А также изучены методы афинных преобразований.