ФГБОУ ВО

Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет) «МГТУ им. Н.Э. Баумана»

Факультет ИУ «Информатика и системы управления»

Кафедра ИУ-3 «Информационные системы и телекоммуникации»

Отчет по лабораторной работе 8

**«Win32API, анимация, таймер»**

По курсу «Информатика»

Выполнил: Еньшин Б. А.

Группа ИУ3-12

Вариант 9

Проверил: Видьманов Д. А.

Москва 2019

**Цель работы:**

Ознакомиться с дополнительными знаниями для построения классического оконного приложения Windows. Узнать о правильных способах использовать таймер. Научиться обрабатывать события, связанные с этим таймером. Научиться рисовать в окне графику и создавать простейшую анимацию. Создать оконное приложение с анимацией.

**Постановка задания:**

Для рисунка из предыдущей лабораторной работы (по вариантам) создать движение фигуры в любую сторону (вправо, влево, вверх, вниз) при нажатии левой кнопки мыши в окне и при нажатии правой кнопки мыши остановить движение.

**Ниже приведен код программы lab8\_source.cpp**

#include <windows.h>

#include <time.h>

#define DIVISIONS 10

LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM);

HINSTANCE hInst;

BOOL Line(HDC hdc, int x1, int y1, int x2, int y2)

{

MoveToEx(hdc, x1, y1, NULL); //сделать текущими координаты x1, y1

return LineTo(hdc, x2, y2);

}

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance,

PSTR szCmdLine, int iCmdShow)

{

LPCSTR szAppName = "HelloWin";

HWND hwnd;

MSG msg;

WNDCLASSEX wndclass;

wndclass.cbSize = sizeof(wndclass);

wndclass.style = CS\_HREDRAW | CS\_VREDRAW;

wndclass.lpfnWndProc = WndProc;

wndclass.cbClsExtra = 0;

wndclass.cbWndExtra = 0;

wndclass.hInstance = hInstance;

wndclass.hIcon = LoadIcon(NULL, IDI\_APPLICATION);

wndclass.hCursor = LoadCursor(NULL, IDC\_ARROW);

wndclass.hbrBackground = (HBRUSH)GetStockObject(WHITE\_BRUSH);

wndclass.lpszMenuName = NULL;

wndclass.lpszClassName = L"HelloWin";

wndclass.hIconSm = LoadIcon(NULL, IDI\_APPLICATION);

RegisterClassEx(&wndclass);

hwnd = CreateWindowA(szAppName,

"Ракета",

WS\_OVERLAPPEDWINDOW,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

CW\_USEDEFAULT,

NULL,

NULL,

hInstance,

NULL);

ShowWindow(hwnd, iCmdShow);

UpdateWindow(hwnd);

while (GetMessage(&msg, NULL, 0, 0))

{

TranslateMessage(&msg);

DispatchMessage(&msg);

}

return msg.wParam;

}

LRESULT CALLBACK WndProc(HWND hwnd, UINT iMsg, WPARAM wParam, LPARAM lParam)

{

// Массив, хранящий текущее состояние секции

static BOOL fState[DIVISIONS][DIVISIONS];

static int cxBlock, cyBlock;

HDC hdc;

PAINTSTRUCT ps;

RECT rect;

static HBRUSH hbrush, hbrush1, hbrush2, hbrush3, hbrush4;

static int x, y;

static bool flagtimer;

switch (iMsg)

{

case WM\_CREATE:

srand(time(0));

x = 500;

y = 200;

hdc = GetDC(hwnd);

return 0;

case WM\_LBUTTONDOWN:

SetTimer(hwnd, 10, 25, NULL);

return 0;

case WM\_RBUTTONDOWN:

KillTimer(hwnd, 10);

return 0;

case WM\_TIMER:

y -= 5;

InvalidateRect(hwnd, NULL, TRUE);

return 0;

case WM\_SIZE:

cxBlock = LOWORD(lParam) / DIVISIONS;

cyBlock = HIWORD(lParam) / DIVISIONS;

return 0;

case WM\_PAINT:

hbrush = CreateSolidBrush(RGB(0, 64, 0));

hbrush1 = CreateSolidBrush(RGB(255, 255, 255));

hbrush2 = CreateSolidBrush(RGB(0, 52, 0));

hbrush3 = CreateSolidBrush(RGB(255, 100, 0));

hbrush4 = CreateSolidBrush(RGB(0, 0, 255));

hdc = BeginPaint(hwnd, &ps);

SelectObject(hdc, hbrush);

HRGN Rgn2, Rgn3, Rgn4\_l, Rgn4\_r;

Rectangle(hdc, x, y, x+100, y+250);

POINT ptArray1[3], ptArray2[3], ptArray3\_l[4], ptArray3\_r[4];

ptArray2[0].x = x; ptArray2[0].y = y;

ptArray2[1].x = x+50; ptArray2[1].y = y-100;

ptArray2[2].x = x+100; ptArray2[2].y = y;

Rgn2 = CreatePolygonRgn(ptArray2, sizeof(ptArray2) / sizeof(POINT), ALTERNATE);

FillRgn(hdc, Rgn2, hbrush2);

ptArray1[0].x = x-10; ptArray1[0].y = y+250;

ptArray1[1].x = x+50; ptArray1[1].y = y+350;

ptArray1[2].x = x+110; ptArray1[2].y = y+250;

Rgn3 = CreatePolygonRgn(ptArray1, sizeof(ptArray1) / sizeof(POINT), ALTERNATE);

FillRgn(hdc, Rgn3, hbrush3);

ptArray3\_l[0].x = x; ptArray3\_l[0].y = y+200;

ptArray3\_l[1].x = x-30; ptArray3\_l[1].y = y+230;

ptArray3\_l[2].x = x-30; ptArray3\_l[2].y = y+300;

ptArray3\_l[3].x = x; ptArray3\_l[3].y = y+270;

Rgn4\_l = CreatePolygonRgn(ptArray3\_l, sizeof(ptArray3\_l) / sizeof(POINT), ALTERNATE);

FillRgn(hdc, Rgn4\_l, hbrush2);

ptArray3\_r[0].x = x+100; ptArray3\_r[0].y = y+200;

ptArray3\_r[1].x = x+130; ptArray3\_r[1].y = y+230;

ptArray3\_r[2].x = x+130; ptArray3\_r[2].y = y+300;

ptArray3\_r[3].x = x+100; ptArray3\_r[3].y = y+270;

Rgn4\_r = CreatePolygonRgn(ptArray3\_r, sizeof(ptArray3\_r) / sizeof(POINT), ALTERNATE);

FillRgn(hdc, Rgn4\_r, hbrush2);

HRGN Rgn\_illume;

Rgn\_illume = CreateEllipticRgn(x+30, y+40, x+70, y+80);

FillRgn(hdc, Rgn\_illume, hbrush4);

EndPaint(hwnd, &ps);

return 0;

case WM\_DESTROY:

PostQuitMessage(0);

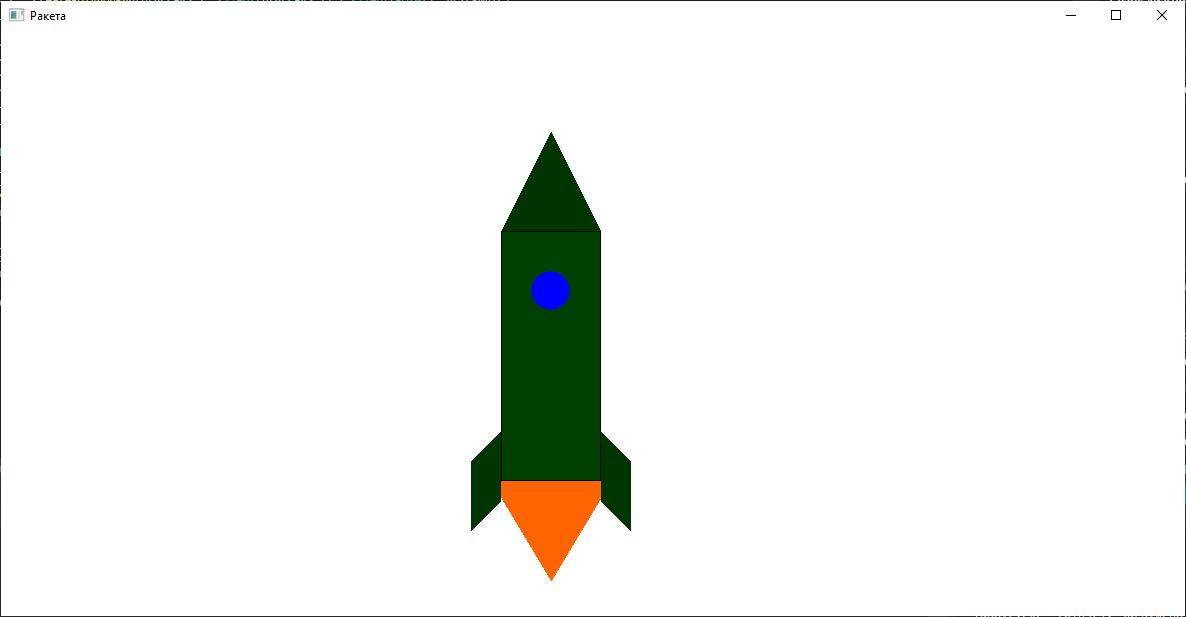
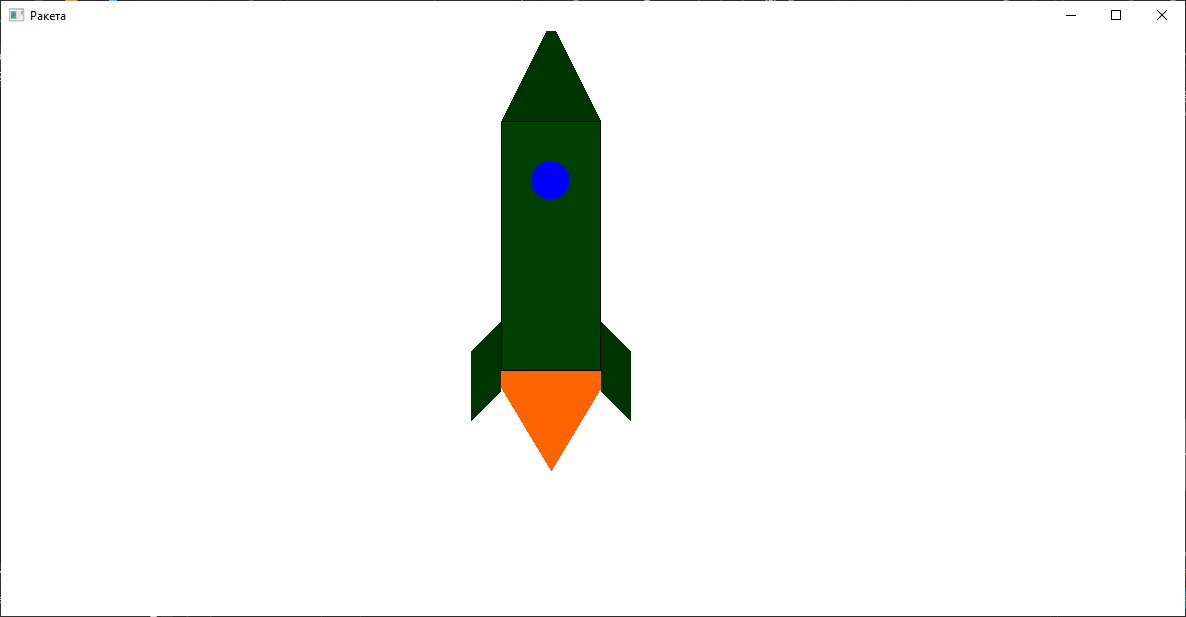
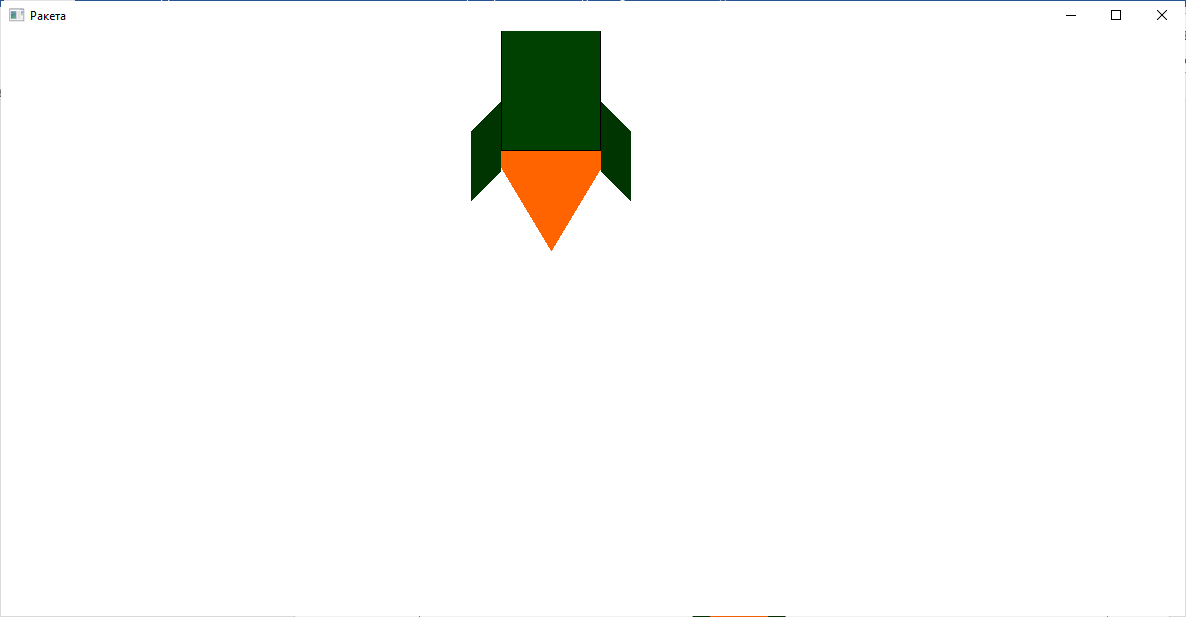
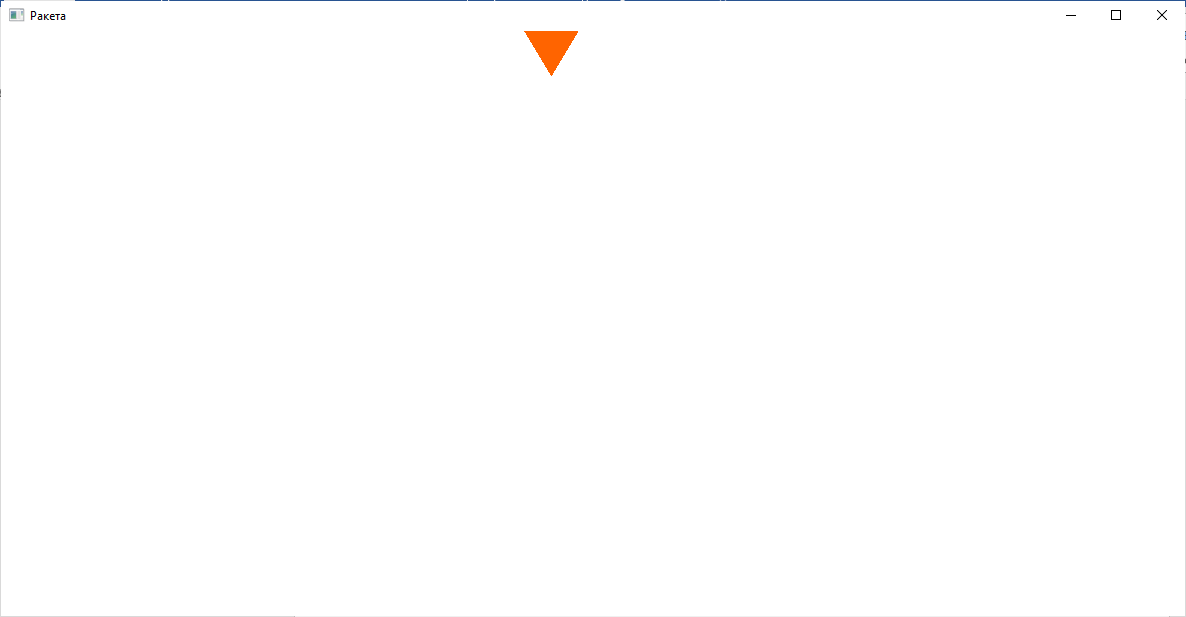
return 0;

}

return DefWindowProc(hwnd, iMsg, wParam, lParam);

}

**Скриншоты приложения:**

**Вывод:**

В ходе данной лабораторной работы я улучшил навыки разработки классических оконных приложений на Win32API. Я научился создавать графику и создавать анимацию. Также я научился обрабатывать такие ситуации, как нажатие кнопки мыши и отсчет таймера. Данные знания были подкреплены решением поставленной задачи на создание оконного приложения на Win32API.