Нарисовать ночное небо

#include <windows.h>

//#include <unistd.h>

#include <stdbool.h>

//#include < stdio.h >

#include <dos.h>

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

int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, PSTR szCmdLine, int iCmdShow)

//iCmdshow 1-выводится окно

{

static char 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(BLACK\_BRUSH);

wndclass.lpszMenuName = NULL;

wndclass.lpszClassName = szAppName;

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

RegisterClassEx(&wndclass);

hwnd = CreateWindow(szAppName,

"Star",

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)

{

HDC hdc;

PAINTSTRUCT ps;

RECT rect;

HBRUSH hBrush;

static bool s;

switch (iMsg)

{

case WM\_CREATE:

SetTimer(hwnd, 1, 1000, 0);

s = false;

break;

case WM\_TIMER:

InvalidateRect(hwnd, NULL, TRUE);

UpdateWindow(hwnd);

hdc = GetDC(hwnd);

if (s) {

for (int i = 0; i < 100; i += 1) {

InvalidateRect(hwnd, NULL, TRUE);

UpdateWindow(hwnd);

HPEN hPen0 = CreatePen(PS\_SOLID, 1, RGB(255, 99, 255));

SelectObject(hdc, hPen0);

POINT pol[11];

pol[0].x = 10 + i;

pol[0].y = 85 ;

pol[1].x = 85 + i;

pol[1].y = 75 ;

pol[2].x = 110 + i;

pol[2].y = 10 ;

pol[3].x = 135 + i;

pol[3].y = 75;

pol[4].x = 210 + i;

pol[4].y = 85 ;

pol[5].x = 160 + i;

pol[5].y = 125;

pol[6].x = 170 + i;

pol[6].y = 190 ;

pol[7].x = 110 + i;

pol[7].y = 150 ;

pol[8].x = 50 + i;

pol[8].y = 190 ;

pol[9].x = 60 + i;

pol[9].y = 125 ;

pol[10].x = 10 + i;

pol[10].y = 85 ;

Polyline(hdc, pol, 11);

hBrush = CreateSolidBrush(RGB(77, 166, 255));

SelectObject(hdc, hBrush);

Ellipse(hdc, 503 + i, 150, 581 + i, 214);

Sleep(3);

}

}

{

for (int i = 0; i < 100; i += 1) {

InvalidateRect(hwnd, NULL, TRUE);

UpdateWindow(hwnd);

HPEN hPen0 = CreatePen(PS\_SOLID, 1, RGB(255, 99, 255));

SelectObject(hdc, hPen0);

POINT pol[11];

pol[0].x = 10 - i;

pol[0].y = 85;

pol[1].x = 85 - i;

pol[1].y = 75;

pol[2].x = 110 - i;

pol[2].y = 10;

pol[3].x = 135 - i;

pol[3].y = 75 ;

pol[4].x = 210 - i;

pol[4].y = 85 ;

pol[5].x = 160 - i;

pol[5].y = 125;

pol[6].x = 170 - i;

pol[6].y = 190;

pol[7].x = 110 - i;

pol[7].y = 150 ;

pol[8].x = 50 - i;

pol[8].y = 190 ;

pol[9].x = 60 - i;

pol[9].y = 125 ;

pol[10].x = 10 - i;

pol[10].y = 85 ;

Polyline(hdc, pol, 11);

hBrush = CreateSolidBrush(RGB(77, 166, 255));

SelectObject(hdc, hBrush);

Ellipse(hdc, 503 + i, 150, 581 + i, 214);

Sleep(3);

}

}

ReleaseDC(hwnd, hdc);

s = !s;

break;

case WM\_DESTROY://Заканчивает программу

KillTimer(hwnd, 1);

PostQuitMessage(0);

return 0;

}

return DefWindowProc(hwnd, iMsg, wParam, lParam);//Все сообщения, не обрабатываемые оконной процедурой, возвр значения сообщ

}

