Афанасенко Корней 151004

Лабораторная 2

Lab2.c

#include <Windows.h>

#include <WinDef.h>

#include "Table.h"

#include "Stamp.h"

const wchar\_t\* const GameZoneName = L"Form";

HWND hwndGameWindow;

HDC hdcBackBuffer;

HANDLE thDraw;

int width, height;

HACCEL hAccelTable;

int type = 0;

void DrawToBackBuffer(HDC hdcBackBuffer) {

if (type) {

DrawStamp(hdcBackBuffer, width, height);

}

else

{

DrawTable(hdcBackBuffer, width, height);

}

}

void MySwapBuffers(HDC hdc, HDC hdcBackBuffer) {

BitBlt(hdc, 0, 0, width, height, hdcBackBuffer, 0, 0, SRCCOPY);

}

void Draw() {

while (1) {

Sleep(100);

InvalidateRect(hwndGameWindow, NULL, 0);

UpdateWindow(hwndGameWindow);

}

}

LRESULT CALLBACK GameZoneProc(HWND hWnd, UINT uMsg, WPARAM wParam, LPARAM lParam)

{

PAINTSTRUCT ps;

HDC hdc;

switch (uMsg)

{

case WM\_CREATE:

{

width = ((CREATESTRUCTA\*)lParam)->cx;

height = ((CREATESTRUCTA\*)lParam)->cy;

hdc = BeginPaint(hWnd, &ps);

hdcBackBuffer = CreateCompatibleDC(hdc);

HBITMAP hBitmap = CreateCompatibleBitmap(hdc, width, height);

SelectObject(hdcBackBuffer, hBitmap);

DeleteObject(hBitmap);

EndPaint(hWnd, &ps);

}

break;

case(WM\_GETMINMAXINFO):

{

((MINMAXINFO\*)lParam)->ptMinTrackSize.x = 300;

((MINMAXINFO\*)lParam)->ptMinTrackSize.y = 150;

}

break;

case(WM\_SIZE):

{

width = LOWORD(lParam);

height = HIWORD((lParam));

hdc = BeginPaint(hWnd, &ps);

hdcBackBuffer = CreateCompatibleDC(hdc);

HBITMAP hBitmap = CreateCompatibleBitmap(hdc, width, height);

SelectObject(hdcBackBuffer, hBitmap);

DeleteObject(hBitmap);

EndPaint(hWnd, &ps);

}

break;

case WM\_PAINT:

{

DrawToBackBuffer(hdcBackBuffer);

hdc = BeginPaint(hWnd, &ps);

MySwapBuffers(hdc, hdcBackBuffer);

EndPaint(hWnd, &ps);

}

break;

case WM\_COMMAND:

{

switch (LOWORD(wParam)) {

case 40001:

type = 0;

break;

case 40002:

type = 1;

break;

}

}

break;

case WM\_DESTROY:

{

DeleteDC(hdcBackBuffer);

PostQuitMessage(0);

}

break;

}

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

}

INT WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow) {

MSG msg;

WNDCLASSEX wcex;

// Регестрация оконого класса

memset(&wcex, 0, sizeof wcex);

wcex.cbSize = sizeof wcex;

wcex.style = CS\_GLOBALCLASS;

wcex.lpfnWndProc = GameZoneProc;

wcex.hCursor = LoadCursor(0, IDC\_ARROW);

wcex.hbrBackground = (HBRUSH)(COLOR\_BACKGROUND + 1);

wcex.lpszClassName = GameZoneName;

RegisterClassEx(&wcex);

// Создание окна

hwndGameWindow = CreateWindowEx(0, GameZoneName, GameZoneName, WS\_VISIBLE | WS\_OVERLAPPEDWINDOW,0, 0, 1060, 660, 0, 0, 0, NULL);

hAccelTable = LoadAccelerators(hInstance, 101);

thDraw = (HANDLE)\_beginthread(Draw, 0, 0);

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

{

if (!TranslateAccelerator(hwndGameWindow, hAccelTable, &msg)) {

TranslateMessage(&msg);

DispatchMessage(&msg);

}

}

CloseHandle(thDraw);

return msg.wParam;

}

Stamp.c

#include <Windows.h>

#include <math.h>

void DrawStamp(HDC hdc, int width, int height) {

SaveDC(hdc);

Rectangle(hdc, 0, 0, width, height);

int cX = 300;

int cY = 300;

int r3 = 240;

int r2 = 190;

int r1 = 140;

double angel = 45;

LPCSTR lpString = L"ABCDEFGHIJKLMNOPQRSTUVWXYZ01234567890";

int len = 38;

Ellipse(hdc, 50, 50, 550, 550);

Ellipse(hdc, 100, 100, 500, 500);

Ellipse(hdc, 150, 150, 450, 450);

Ellipse(hdc, 200, 200, 400, 400);

Ellipse(hdc, 250, 250, 350, 350);

Ellipse(hdc, 290, 290, 310, 310);

double changeAngel = 360.0 / len;

for (int i = len-1; i > -1; i--) {

HFONT hFont = CreateFont(24, 0, -(angel + 90)\*10, 0, FW\_NORMAL, FALSE, FALSE, FALSE,

ANSI\_CHARSET, OUT\_DEFAULT\_PRECIS, CLIP\_DEFAULT\_PRECIS,

DEFAULT\_QUALITY, DEFAULT\_PITCH | FF\_SWISS, "Arial");

SelectObject(hdc, hFont);

double radAngel = 3.14 \* angel / 180.0;

int x = cX + r1 \* cos(radAngel);

int y = cY + r1 \* sin(radAngel);

TextOut(hdc, x, y, &lpString[i \* 2], 1);

x = cX + r2 \* cos(radAngel);

y = cY + r2 \* sin(radAngel);

TextOut(hdc, x, y, &lpString[i \* 2], 1);

x = cX + r3 \* cos(radAngel);

y = cY + r3 \* sin(radAngel);

TextOut(hdc, x, y, &lpString[i \* 2], 1);

angel -= changeAngel;

DeleteObject(hFont);

}

RestoreDC(hdc, -1);

}

Stamp.h

#pragma once

void DrawStamp(HDC hdc, int width, int height);

Table.c

#include <Windows.h>

#define TEXT L"TextBox"

#define ROWS 4

#define COLUMNS 3

void DrawTable(HDC hdc, int width, int height) {

SaveDC(hdc);

Rectangle(hdc, 0, 0, width, height);

int tmpWidth = width / COLUMNS;

int tmpHeight = height / ROWS;

for (int i = ROWS-1; i > 0; i--) {

MoveToEx(hdc,0, tmpHeight\*i, NULL);

LineTo(hdc, width, tmpHeight \* i, NULL);

}

for (int i = COLUMNS-1; i > 0; i--) {

MoveToEx(hdc, tmpWidth\*i, 0, NULL);

LineTo(hdc, tmpWidth \* i, height, NULL);

}

RECT\* rect = malloc(sizeof(RECT));

for (int j = (ROWS); j > 0; j--) {

for (int i = (COLUMNS); i > 0; i--) {

rect->left = tmpWidth \* (i - 1);

rect->top = tmpHeight \* (j - 1);

rect->right = tmpWidth \* (i);

rect->bottom = tmpHeight \* (j);

DrawText(hdc, TEXT, -1, rect, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

}

}

free(rect);

RestoreDC(hdc, -1);

}

Table.h

#pragma once

void DrawTable(HDC hdc, int width, int height);