*Министерство образования Республики Беларусь*

*Учреждение образования*

*«Брестский государственный технический университет»*

*Кафедра ИИТ*

Лабораторная работа №1

По дисциплине: «Современные средства вычислительной техники»

Тема:«Использование DirectInput

для прямой работы с устройствами ввода»

Выполнил:

Студент 3-го курса

группы ИИ-23

Романюк А.П.

Проверил:

Кулеша В. И.

Брест 2025

**Цель:** Изучить программные инструменты для прямой работы с устройствами ввода. Научиться программировать клавиатуру и мышь.

**Задание:** Написать программу, которая перечисляет все устройства компьютера поддерживающие DirectInput, отслеживает нажатия 5 клавиш и мыши.

**Ход работы**

#define WIN32\_LEAN\_AND\_MEAN

#define UNICODE

#define \_UNICODE

#include <windows.h>

#include <dinput.h>

#include <vector>

#include <string>

#include <shellapi.h> // Для ShellExecute

#pragma comment(lib, "dinput8.lib")

#pragma comment(lib, "dxguid.lib")

#pragma comment(lib, "shell32.lib") // Для ShellExecute

LPDIRECTINPUT8 dinput;

LPDIRECTINPUTDEVICE8 keyboard;

LPDIRECTINPUTDEVICE8 mouse;

char keyBuffer[256];

DIMOUSESTATE mouseState;

std::vector<std::wstring> devices;

POINT clickPosition = { -1, -1 };

bool messageShown = false;

LRESULT CALLBACK WindowProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam);

void UpdateInput(HWND hwnd);

void EnumerateDevices();

void OpenApplication();

bool InitDirectInput(HINSTANCE hInstance, HWND hwnd) {

if (FAILED(DirectInput8Create(hInstance, DIRECTINPUT\_VERSION, IID\_IDirectInput8, (void\*\*)&dinput, NULL)))

return false;

if (FAILED(dinput->CreateDevice(GUID\_SysKeyboard, &keyboard, NULL)))

return false;

keyboard->SetDataFormat(&c\_dfDIKeyboard);

keyboard->SetCooperativeLevel(hwnd, DISCL\_BACKGROUND | DISCL\_NONEXCLUSIVE);

keyboard->Acquire();

if (FAILED(dinput->CreateDevice(GUID\_SysMouse, &mouse, NULL)))

return false;

mouse->SetDataFormat(&c\_dfDIMouse);

mouse->SetCooperativeLevel(hwnd, DISCL\_BACKGROUND | DISCL\_NONEXCLUSIVE);

mouse->Acquire();

EnumerateDevices();

return true;

}

BOOL CALLBACK EnumDevicesCallback(LPCDIDEVICEINSTANCE lpddi, LPVOID pvRef) {

std::vector<std::wstring>\* deviceList = reinterpret\_cast<std::vector<std::wstring>\*>(pvRef);

deviceList->push\_back(lpddi->tszProductName);

return DIENUM\_CONTINUE;

}

void EnumerateDevices() {

devices.clear();

dinput->EnumDevices(DI8DEVCLASS\_ALL, EnumDevicesCallback, &devices, DIEDFL\_ATTACHEDONLY);

}

void OpenApplication() {

ShellExecute(NULL, L"open", L"notepad.exe", NULL, NULL, SW\_SHOWNORMAL);

}

void UpdateInput(HWND hwnd) {

HRESULT hr = keyboard->GetDeviceState(sizeof(keyBuffer), (LPVOID)&keyBuffer);

if (FAILED(hr)) {

keyboard->Acquire();

return;

}

hr = mouse->GetDeviceState(sizeof(DIMOUSESTATE), (LPVOID)&mouseState);

if (FAILED(hr)) {

mouse->Acquire();

return;

}

if ((keyBuffer[DIK\_A] & 0x80) &&

(keyBuffer[DIK\_S] & 0x80) &&

(keyBuffer[DIK\_D] & 0x80) &&

(keyBuffer[DIK\_F] & 0x80) &&

(keyBuffer[DIK\_G] & 0x80)) {

OpenApplication();

}

}

void Render(HWND hwnd) {

PAINTSTRUCT ps;

HDC hdc = BeginPaint(hwnd, &ps);

wchar\_t buffer[512];

if (clickPosition.x != -1 && clickPosition.y != -1) {

swprintf(buffer, sizeof(buffer) / sizeof(wchar\_t), L"Click at: X=%ld, Y=%ld", clickPosition.x, clickPosition.y);

TextOut(hdc, 10, 30, buffer, wcslen(buffer));

}

int y = 50;

for (const auto& device : devices) {

TextOut(hdc, 10, y, device.c\_str(), device.size());

y += 20;

}

EndPaint(hwnd, &ps);

}

int WINAPI wWinMain(HINSTANCE hInstance, HINSTANCE, PWSTR, int nCmdShow) {

WNDCLASS wc = { 0 };

wc.lpfnWndProc = WindowProc;

wc.hInstance = hInstance;

wc.lpszClassName = L"DirectInputWindow";

RegisterClass(&wc);

HWND hwnd = CreateWindow(L"DirectInputWindow", L"DirectInput Example",

WS\_OVERLAPPEDWINDOW, 100, 100, 400, 300, NULL, NULL, hInstance, NULL);

ShowWindow(hwnd, nCmdShow);

if (!InitDirectInput(hInstance, hwnd)) return 0;

MSG msg = { 0 };

while (msg.message != WM\_QUIT) {

if (PeekMessage(&msg, NULL, 0, 0, PM\_REMOVE)) {

TranslateMessage(&msg);

DispatchMessage(&msg);

}

UpdateInput(hwnd);

}

keyboard->Unacquire();

keyboard->Release();

mouse->Unacquire();

mouse->Release();

dinput->Release();

return 0;

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Контент, сгенерированный ИИ, может содержать ошибки.}

LRESULT CALLBACK WindowProc(HWND hwnd, UINT uMsg, WPARAM wParam, LPARAM lParam) {

switch (uMsg) {

case WM\_LBUTTONDOWN:

clickPosition.x = LOWORD(lParam);

clickPosition.y = HIWORD(lParam);

InvalidateRect(hwnd, NULL, FALSE);

break;

case WM\_PAINT:

Render(hwnd);

break;

case WM\_DESTROY:

PostQuitMessage(0);

break;

default:

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

}

return 0;

}

**Вывод:** Изучил программные инструменты для прямой работы с устройствами ввода. Научился программировать клавиатуру и мышь.