

Obsługa myszy

1. Tworzymy aplikację okienkową. Nazwijmy projekt „Mysz”.
2. Dodajemy plik nagłówkowy *Program07.h* i wypełniamy go następującym kodem:

```
const char NazwaProgramu[] = "Program07";
const int MaxX = 400;
const int MaxY = 300;
void Rysuj(HDC hdc, int x, int y)
{
    Ellipse(hdc, x-25, y-25, x+25, y+25);
}
```

3. W pliku *main.cpp* dołączamy plik *Program07.h*:

```
#include "Program07.h"
```

4. W funkcji *CreateWindowEx()* zmieniamy:
 - "Windows App", na stałą *NazwaProgramu*,
 - 544 na *MaxX+8*,
 - 375 na *MaxY+34*,
5. W funkcji *WindowProcedure()* umieszczamy:

```
PAINTSTRUCT ps;
HDC hdc;
oraz:
```

```
case WM_PAINT:
    hdc = BeginPaint(hwnd, &ps);
    Rysuj(hdc);
    EndPaint(hwnd, &ps);
    break;
```

oraz nowość:

```
case WM_MOUSEMOVE:
    x_kursora = LOWORD(IParam);
    y_kursora = HIWORD(IParam);
    InvalidateRect(hwnd, NULL, TRUE);
    break;
```

Jest to komunikat obsługujący zdarzenie przesuwania myszy po obszarze roboczym. (pozostałe możliwe komunikaty w dalszej części materiału).

IParam – to dwuwierszowe słowo zwracające bieżącą pozycję kursora – młodsze – x; starsze – y;

BOOL InvalidateRect(HWND hwnd, CONST RECT obszar, BOOL tlo);* - to funkcja, którą będzie wymuszać odmalowanie na nowo okna roboczego po każdym pobraniu współrzędnych.

- *NULL* – oznacza, że odrysowania wymaga cały obszar roboczy
- *TRUE* – nakazuje również odmalowanie tła;

6. Przed SWITCH'em należy zadeklarować:

```
static int x_kursora, y_kursora;
```

Dlaczego static? Bo wartości nie mogą być usuwane po opuszczeniu klasy okna.

7. Do wywołania funkcji *Rysuj(hdc)* dodajemy dwa parametry: *x_kursora*, *y_kursora*.
8. Uruchamiamy program. Narysowane koło powinno „podążać” za kursorem.

Obsługiwane komunikaty myszy:

WM_MOUSEMOVE — kursor myszy został przesunięty nad obszarem roboczym okna.

WM_LBUTTONDOWN — został naciśnięty lewy przycisk myszy.

WM_LBUTTONUP — lewy przycisk myszy został zwolniony.

WM_LBUTTONDBLCLK — lewy przycisk myszy został naciśnięty dwukrotnie.

WM_RBUTTONDOWN — został naciśnięty prawy przycisk myszy.

WM_RBUTTONUP — prawy przycisk myszy został zwolniony.

WM_RBUTTONDBLCLK — prawy przycisk myszy został naciśnięty dwukrotnie.

Zadania do rozwiązania:

1. Zmodyfikuj powyższy program w taki sposób, aby naciśnięcie lewego przycisku myszy powiększało koło, a naciśnięcie prawego przycisku myszy pomniejszało koło.
2. Napisz program, który będzie wyświetlał napis – twoje imię i nazwisko. Napis będzie koloru zielonego. Pojedyncze kliknięcie LPM będzie przesunęło napis we wskazane miejsce, dwukrotne kliknięcie PPM zmieni kolor na czerwony i dwukrotne kliknięcie LPM przywróci kolor zielony.