Kurs:
Urządzenia Peryferyjne - Laboratorium

Joystick - sterowanie kursorem

Autor:
Aleksandra Chrustek, 263900
Jonasz Trocha, 263951

Prowadzący:
dr inż. Dariusz Caban

29.11.2023

1 Wstęp

Celem ćwiczenia było zapoznanie się z zasadami działania joysticka oraz napisanie oprogramowania odczytującego stan joysticka i przekładającego go na konkretne działania na poziomie aplikacji.

2 Informacje teoretyczne

2.1 Budowa i zasada działania kontrolerów typu: joystick, gamepad, mysz

2.1.1 Joystick

Budowa: Joystick to urządzenie wejściowe, które składa się z drążka (lub kilku drążków) oraz przycisków. Ruch drążka w różnych kierunkach i naciskanie przycisków generują sygnały wejściowe.
Zasada działania: Joystick rejestruje ruchy drążka oraz naciśnięcia przycisków, przekazując te informacje do komputera, który je interpretuje, na przykład w grach.

2.1.2 Gamepad

Budowa: Gamepad to kontroler, który zawiera przyciski akcji, drążki analogowe, trigger'y, oraz inne elementy, takie jak D-pad. Jest bardziej kompaktowy niż joystick.
Zasada działania: Gamepad działa na zasadzie przesyłania sygnałów elektrycznych z naciśniętych przycisków i ruchów drążków, które są interpretowane przez system komputerowy.

2.1.3 Mysz

Budowa: Mysz składa się z czujnika optycznego (lub laserowego), przycisków (zwykle lewy, prawy i środkowy), a czasem także rolki do przewijania.
Zasada działania: Ruch myszy po powierzchni generuje sygnały, które są przekazywane do komputera. Ruch kursora na ekranie jest proporcjonalny do ruchu myszy.

2.2 DirectX firmy Microsoft i jej komponent - biblioteka DirectInput DirectX

Definicja: DirectX to zbiór interfejsów programistycznych aplikacji (API), rozwijanych przez Microsoft, które są używane do obsługi zadań związanych z grafiką, dźwiękiem, wejściem, i innymi aspektami związanymi z multimediów w systemie Windows.
Składniki: DirectX składa się z różnych komponentów, takich jak Direct3D (do obsługi grafiki 3D), DirectSound (do obsługi dźwięku) i DirectInput (do obsługi urządzeń wejściowych).

2.3 DirectInput

Definicja: DirectInput to biblioteka wchodząca w skład DirectX, zapewniająca interfejs programistyczny do obsługi urządzeń wejściowych, takich jak joysticki, gamepady i myszy.
Funkcje: DirectInput umożliwia programistom obsługę różnych typów urządzeń wejściowych, zarówno analogowych, jak i cyfrowych. Pozwala na reakcję na ruchy, przyciski, oraz inne zdarzenia generowane przez te urządzenia.

2.4 Koncepcje: Hardware Abstraction Layer (HAL), Hardware Emulation Layer (HEL), Human Interface Device (HID)

2.4.1 Hardware Abstraction Layer (HAL)

Definicja: HAL to warstwa oprogramowania, która ukrywa szczegóły implementacyjne sprzętu przed systemem operacyjnym. Zapewnia jednolite API dla różnych rodzajów sprzętu.
Rola w DirectInput: W kontekście DirectInput, HAL pozwala na dostęp do urządzeń wejściowych, niezależnie od ich fizycznej implementacji. Dzięki temu, programy korzystające z DirectInput nie muszą się bezpośrednio martwić o szczegóły sprzętowe.

2.4.2 Hardware Emulation Layer (HEL)

Definicja: HEL to warstwa oprogramowania, która emuluje pewne funkcje sprzętu, gdy rzeczywisty sprzęt nie obsługuje tych funkcji lub gdy są one niedostępne.
Rola w DirectInput: W przypadku DirectInput, HEL może być wykorzystywane do emulacji pewnych cech urządzeń wejściowych, co pozwala na jednolite działanie programów niezależnie od dostępności konkretnego sprzętu.

2.4.3 Human Interface Device (HID)

Definicja: HID to standard definiujący sposób, w jaki urządzenia wejściowe, takie jak klawiatury, myszy i gamepady, komunikują się z komputerem.
Rola w DirectInput: DirectInput korzysta z protokołu HID, co umożliwia obsługę różnych urządzeń wejściowych zgodnych z tym standardem. HID ułatwia komunikację między urządzeniem a komputerem, niezależnie od konkretnego sprzętu.

3 Realizacja zadania

Na laboratorium pracowaliśmy z joystickiem Logitech Attack 3. Aplikacja jest zaprojektowana do obsługi wejścia z joysticka, umożliwiając sterowanie kursorem na ekranie w oparciu o ruchy joysticka. Wykorzystano wielowątkowość poprzez utworzenie oddzielnego wątku (joyThread), który zajmuje się ciągłym odczytem stanu joysticka. Do obsługi joysticka użyto biblioteki SharpDX.DirectInput, co umożliwia bezpośredni dostęp do urządzeń wejściowych. Naciśnięcie przycisku na joysticku (mes.Buttons[0]) powoduje symulację kliknięcia myszą w aktualnej pozycji kursora. Metoda Update jest wywoływana w pętli głównej, co pozwala na aktualizację interfejsu użytkownika na bieżąco w związku ze zmianami stanu joysticka. Wykorzystano DirectInput do pobrania informacji o dostępnych urządzeniach wejściowych, a następnie skonfigurowano i pozyskano joystick do dalszej obsługi.

Poniżej przedstawione zostały poszczególne metody do obsługi joysticka:

1. Konstruktor MainWindow: Ustawia początkowy stan okna i jego komponentów, takich jak przyciski i pola tekstowe.

```
public MainWindow()
{
    InitializeComponent();

}
```

2. Metoda MainForJoystick: Inicjalizuje DirectInput, uzyskuje dostęp do joysticka i wchodzi w nieskończoną pętlę, aby ciągle odczytywać stan joysticka. Następnie aktualizuje interfejs użytkownika, wywołując metodę Update na wątku interfejsu użytkownika.

```
void MainForJoystick()
{
    // Initialize DirectInput
    var directInput = new DirectInput();

    var joystickGuid = Guid.Empty;

    foreach (var deviceInstance in directInput.GetDevices(DeviceType.Joystick, Dev
        joystickGuid = deviceInstance.InstanceGuid;
```

```
    // Instantiate the joystick
    var joystick = new Joystick(directInput, joystickGuid);

    // Set BufferSize in order to use buffered data.
    joystick.Properties.BufferSize = 128;

    // Acquire the joystick
    joystick.Acquire();

    while (true)
    {
        var state = joystick.GetCurrentState();
        MyTextBox.Dispatcher.Invoke(new UpdateTextCallback(Update), state);
    }
}
```

3. Metoda Update: Sprawdza, czy aplikacja jest w swoim początkowym stanie. Ustawia tekst w elemencie MyTextBox na podstawie odczytanego stanu joysticka. Wywołuje metodę DoMouseClick, gdy naciśnięty jest przycisk na joysticku. Wypisuje informacje do konsoli dotyczące położenia joysticka. Ustawia pozycję kursora za pomocą funkcji SetCursorPos na podstawie odczytanego stanu joysticka.

```
private void Update(JoystickState mes)
{
    if (Initial == false)
    {
        SetCursorPos(0, 0);
        Initial = true;
    }

    MyTextBox.Text = mes.X.ToString() + " " + mes.Y.ToString();
    MyTextBox.Text = $"x: {mes.X}\r\n";
    MyTextBox.Text += $"y: {mes.Y}\r\n";

    if (mes.Buttons[0])
    {
        DoMouseClick(mes);
        //inkCanvas1.CaptureMouse();
    }
```

```
    Console.WriteLine($"x: {mes.X} y: {mes.Y} Nacisniety: {mes.Buttons[0]}");
    SetCursorPos((int)(mes.X* 0.059), (int)(mes.Y * 0.035));
}
```

4. Metoda Buttonn_Click: Czyści wszystkie rysunki na elemencie inkCanvas1.

```
private void Buttonn_Click(object sender, RoutedEventArgs e)
{
    this.inkCanvas1.Strokes.Clear();
}
```

5. Metoda ConnectButton_Click: Obsługuje zdarzenie kliknięcia przycisku łączenia (ConnectButton). Tworzy nowy wątek (joyThread) do obsługi wejścia z joysticka i uruchamia go.

```
private void ConnectButton_Click(object sender, RoutedEventArgs e)
{
    joyThread = new Thread(new ThreadStart(MainForJoystick));
    joyThread.Start();
}
```

6. Metoda DisconnectButton_Click: Obsługuje zdarzenie kliknięcia przycisku rozłączania (DisconnectButton). Kończy wątek joyThread, co efektywnie kończy obsługę joysticka.

```
private void DisconnectButton_Click(object sender, RoutedEventArgs e)
{
    joyThread.Join();
}
```

7. Metoda DoMouseClick: Symuluje kliknięcie myszą na podstawie stanu joysticka. Wywołuje funkcję mouse_event z odpowiednimi parametrami, aby zasymulować naciśnięcie i puszczenie lewego przycisku myszy na podstawie odczytanego stanu joysticka.

```
public void DoMouseClick(JoystickState mes)
{
    mouse_event(MOUSEEVENTF_LEFT_DOWN | MOUSEEVENTF_LEFT_UP, (uint)(mes.X * 0.059
}
```

4 Wnioski

Zadanie miało na celu wprowadzenie w obszar programowania urządzeń peryferyjnych, konkretnie joysticka. Realizacja tego zadania okazała się wymagająca, ponieważ wiązała się z koniecznością zaznajomienia się z zasadami działania joysticka oraz zrozumieniem, jak obsługa tego rodzaju urządzenia wygląda z perspektywy komputera. W konsekwencji, musieliśmy znaleźć sposoby na efektywne odczytywanie stanu joysticka i przekładanie go na konkretne działania na poziomie aplikacji. W trakcie laboratorium udało się zaimplementować metody obsługujące odczytywanie stanu joysticka, przenoszenie tego stanu na interfejs użytkownika oraz symulację kliknięć myszą. Ponadto, zastosowanie wielowątkowości w obszarze obsługi wejścia umożliwia aplikacji responsywne reagowanie na dynamicznie zmieniający się stan joysticka. W rezultacie, zadanie dostarczyło nam praktycznej wiedzy na temat programowania urządzeń peryferyjnych, a zdobyte umiejętności mogą być użyteczne w projektach obejmujących obsługę różnorodnych urządzeń wejściowych.

Grupa B: Aleksandra Chmielek, Jonasz Trocha 29.11.2023

Zadanie 5: Joystick - sterowanie kursorem

1. Do wykonania zadania użyjemy biblioteki DirectInput
2. Jako pierwsze mieliśmy za zadanie napisanie programu wczytującego nazwę joysticka: wyliczenie urządzeń podpiętych do komputera i ~~wybór~~ ~~aktywacja~~ aktywacja wybranego. Kod związany z tym pkt. umieściliśmy w metodzie MainForJoystick.
3. Następnie mieliśmy dodać funkcję wyliczanie elementów/osi joysticka i przyporządkowanie im zakresów. Kod odpowiedzialny za to również umieściliśmy w MainForJoystick
4. Odczytywanie stanu (przycisk, potencjometry, pozycja drążka) joysticka - jest to robione również w metodzie MainForJoystick, w nieskończonej pętli, która aktualizuje stan joysticka.
5. Przejęcie / zdublowanie działania myszy, sterowanie kursorem oraz rysowanie na ekranie za pomocą joysticka: obsługujemy to w metodzie Update: funkcja SetCursorPos kontroluje pozycję kursora na podstawie współrzędnych X i Y joysticka.
6. Stworzyliśmy także okienko w celu możliwości wyświetlenia obsługi joysticka, jak również aby wyświetlać na bieżąco koordynaty ~~jo~~ kursora na ekranie.

Koordynaty kursora obliczane są na podstawie współrzędnych joysticka: X przemnożony jest o 0,059, a Y o 0,035 w celu odpowiedniego przeskalowania do ekranu

Cabo
29.11.23