Mateusz Krawczak 241318

Karol Jaskółka 241306

Grupa: Pon P 17:00

Data wykonania ćwiczenia: 28.10.2019

**Urządzenia Peryferyjne**

**Ćwiczenie 10 – Obsługa joysticka USB z wykorzystaniem DirectInput**

# **Wstęp**

Na pierwszych zajęciach laboratoryjnych otrzymaliśmy zadanie polegające na napisaniu programu pozwalającego na prostą obsługę joysticka podłączonego do komputera  
za pomocą portu USB. Zadania do wykonania:

* Korzystając z przykładowej aplikacji stwierdzić obecność joysticka podłączonego do portu USB komputera
* Napisać program, odczytujący nazwę zainstalowanego joysticka, wykorzystując w tym celu API DirectInput
* Napisać program ilustrujący działanie joysticka: stwierdzający naciśnięcie myszy, zmianę położenia drążka (w przestrzeni 2D) oraz suwaka.
* Napisać program zastępujący działanie myszy. Program ma umożliwiać sterowanie kursorem za pomocą joysticka oraz obsługę przycisków fire jako kliknięć myszy.
* Napisać program realizujący prosty edytor graficzny - rysowanie przy pomocy Joysticka

Kod naszego programu został napisany w języku C#.

# **Zagadnienia**

HAL / HEL :

## DirectX wykorzystuje dwa sterowniki, do sterowania sprzętem:

## -warstwę abstrakcji sprzętu Hardware Abstraction Layer (HAL)

## -warstwę emulacji sprzętowej Hardware Emulation Layer (HEL)

## Po zainicjowaniu DirectX sprawdza sprzęt, czy obsługuje on określone funkcje. Jeżeli je posiada wtedy zostanie użyta warstwa HAL, żeby uzyskać tą funkcjonalność. W przeciwnym przypadku warstwy HEL użyta będzie do emulacji potrzebnych funkcjonalności. Innymi słowy :

## HAL - wykorzystuje funkcjonalność sprzętową

## HEL - implementuje własną funkcjonalność. Pozwala to na łatwiejsze rozwijanie na kolejnych generacjach sprzętu.

## Port USB:

Uniwersalna magistrala szeregowa (Universal Serial Bus) to powszechny port umożliwiający komunikację pomiędzy komputerem, a dowolnym urządzeniem peryferyjnym. Opracowany został między innymi przez Microsoft, IBM i Intel. Urządzenia podłączone za pomocą portu USB mogą być automatycznie wykryte i rozpoznane przez system. Z upływem lat powstały kolejne generacje portu USB zwiększając prędkość transmisji danych przez ten port.

* + - * USB 1.1: do 1,5 MB / s
      * USB 2.0: do 60 MB / s
      * USB 3.1 gen. 1: do 625 MB / s
      * USB 3.1 gen. 2: do 1,25 GB / s

## HID (Human Interface Device):

Są to urządzenia podłączane do komputera, które wysyłają do niego informacje generowane przez człowieka np. mysz, klawiatura, joystick. Urządzenia HID obecnie produkowane są z myślą, aby móc je podłączać   
za pomocą portu USB.

## COM (Component Object Model):

Jest to standard definiowania i tworzenia interfejsów opracowany przez firmę Microsoft oraz w oparciu o zestaw adekwatnych bibliotek. Na podstawie standardu COM zostało opracowanych dużo niskopoziomowych API dla produktów firmy Microsoft – między innymi DirectX.

## Biblioteka DirectInput:

Powyższa biblioteka umożliwia obsługę urządzeń wejściowych takich jak mysz, klawiatura czy joystick. Wchodzi ona w skład zestawu funkcji API DirectX firmy Microsoft. Za jej pomocą możliwe jest odczytywanie informacji   
o podłączanych urządzeniach oraz przypisywanie im określonych akcji (przykładowo po wciśnięciu określonego przycisku). Biblioteka ta najczęściej stosowana jest podczas tworzenia gier komputerowych oraz aplikacji interaktywnych.

1. **Kod programu**

Cały program znajduję się pod tym linkiem <https://github.com/matson19/UP/tree/master/Lab1%20-%20joystick>. Tutaj zamieszczone zostaną jedynie fragmenty kodu.

Podana funkcja opisuje działanie przycisku pierwszego.

private void button1\_Click(object sender, EventArgs e)

{

int selectedDevice = listBox1.SelectedIndex; // zaznaczone urzadzenie staje sie naszym kontrolerem

SharpDX.DirectInput.Joystick joystick = new SharpDX.DirectInput.Joystick(directInput, deviceList.ElementAt(selectedDevice).InstanceGuid);

joystick.Acquire();

mouse = new Mouse(joystick); //dzieki temu wlaczamy mozliwosc emulacji myszy

new Thread(new ThreadStart(mouse.WlaczEmulacje)).Start();

timer1.Start();

}

Funkcja time1\_Tick zmienia status checkBox’ów po tym jak naciśniemy dany przycisk na joysticku.

private void timer1\_Tick(object sender, EventArgs e)

{

// X

positionX.Text = mouse.positionX.ToString();

// Y

positionY.Text = mouse.positionY.ToString();

// fire

if (mouse.joystick.GetCurrentState().Buttons[0]) { checkBox3.Checked = true; } else { checkBox3.Checked = false; }

// button 2

if (mouse.joystick.GetCurrentState().Buttons[1]) { checkBox2.Checked = true; } else { checkBox2.Checked = false; }

// button 3

if (mouse.joystick.GetCurrentState().Buttons[2]) { checkBox1.Checked = true; } else { checkBox1.Checked = false; }

// button 4

if (mouse.joystick.GetCurrentState().Buttons[3]) { checkBox4.Checked = true; } else { checkBox4.Checked = false; }

// button 5

if (mouse.joystick.GetCurrentState().Buttons[4]) {checkBox5.Checked = true;} else {checkBox5.Checked = false;}

}

1. **Podsumowanie**

Podczas pierwszych zajęć laboratoryjnych udało nam się zrealizować odczytanie obecności i nazwy joysticka podłączonego do portu USB, odczytywanie aktywności przycisków oraz pozycję wychylenie drążka oraz program zastępujący działanie myszy.