# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Національний університет “Львівська політехніка”**



**Інститут післядипломної освіти**

**ЗВІТ**

**Про виконання лабораторної роботи №1**

**«Робота з мультимедійними даними засобами WPF»**

**з дисципліни «Мультимедійні системи»**

Виконав:

слухач групи ПЗС-21

Гринчук Тарас

Прийняла:

доц. Павич Н.Я.

« »\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2014 р.

∑ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

ЛЬВІВ – 2014

**Тема роботи**: Робота з мультимедійними даними засобами WPF.

**Мета роботи**: Отримати практичні навики робота з мультимедійними даними засобами WPF.

## Теоретичні відомості

**Windows Presentation Foundation (WPF)** — графічна (презентаційна) підсистема в складі .NET Framework, що має пряме відношення до XAML. WPF разом з .NET Framework 3.0 вбудована в Windows Vista, а також доступна для установки в Windows XP Service Pack 2 і Windows Server 2003.

Це перше реальне оновлення технологічного середовища призначеного для користувача інтерфейсу з часу випуску Windows 95. Воно включає нове ядро, яке повинне замінити GDI і GDI+, використовувані на нинішній Windows-платформі. WPF є високорівневим об'єктно-орієнтованим функціональним шаром (англ. framework), що дозволяє створювати двовимірні та тривимірні інтерфейси.

XAML (скорочення від Extensible Application Markup Language — розширювана мова розмітки застосунків) є мовою розмітки, яку використовують для створення екземплярів об'єктів .NET. Хоча мова XAML — це технологія, що може бути застосовна до багатьох різних предметних областей, її головне призначення — конструювання інтерфейсів користувачів WPF. Інакше кажучи, документи XAML визначають розташування панелей, кнопок та інших елементів керування, що становлять вікна в застосунку WPF. Малоймовірно, що вам доведеться писати код XAML вручну. Замість цього ви використовуєте інструмент, що генерує необхідний код XAML. Існує кілька підмножин XAML:

WPF XAML включає елементи, що описують вміст WPF з розряду векторної графіки, елементів керування й документів. У цей час це найважливіше застосування XAML.

XPS XAML — частина WPF XAML, що визначає XML-подання відформатованих електронних документів. Вона опублікована як окремий стандарт XML Paper Specification (XPS).

Silverlight XAML — підмножина WPF XAML, призначена для Silverlight-застосунків. Можна відвідати сайт http://www.silverlight.net, щоб ознайомитися з деталями.

WF XAML включає елементи, що описують уміст Windows Workflow Foundation (WF). Додаткова інформація про WF доступна на сайті <http://www.wf.netfx3.com>

У процесі виконання кожний застосунок WPF представлено екземпляром класу System.Windows.Application. Цей клас відслідковує всі відкриті вікна у вашому застосунку, вирішує, коли ваш застосунок повинен бути зупиненим, і ініціює події застосунка, які ви можете обробляти для виконання ініціалізації або очищення. У WPF застосунок проходить через простий життєвий цикл. Одразу після запуску вашого застосунка створюється об'єкт застосунка. У процесі його виконання виникають різні події застосунка, які ви можете відслідковувати. І, нарешті, коли об'єкт застосунка звільняється, ваш застосунок завершується. Найпростіший спосіб використання класу Application полягає в його створенні вручну. Звичайно клас Application залишає ваш застосунок активним до тих пір, доки хоча б одне вікно залишається відкритим. Якщо вам не потрібно таке поводження, ви можете змінити Application.ShutdownMode. Якщо ви створюєте об'єкт Application вручну, вам необхідно установити властивість ShutdownMode перед запуском Run().

**2. Хід роботи**

Завдання:

1. Розробити відео плейєр засобами WPF. Програма повинна забезпечувати можливість вибору відео, а також елементи керування play, pause,stop.

Послідовність виконання:

В середовищі MS Visual Studio створимо новий проект WPF (рис. 1).

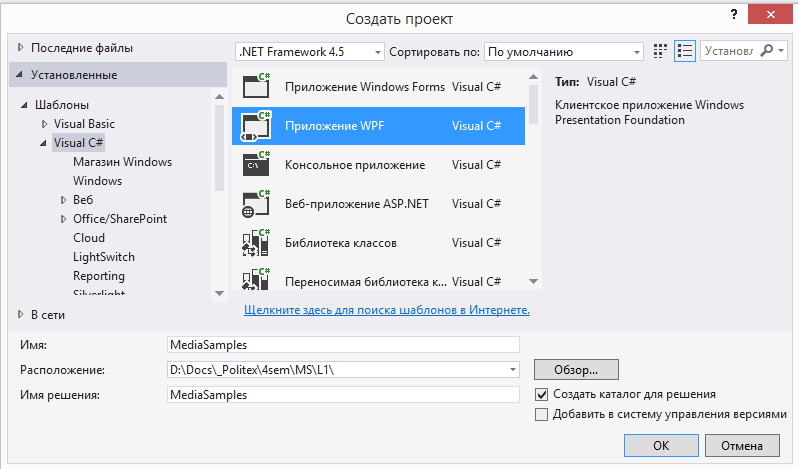


Рис. 1. Створення проекту WPF

Перетягнемо на форму елемент MediaElement для відображення відео та три кнопки: для старту відео, паузи та припинення програвання. Задамо їм відповідні до їх функціональних призначень імена (рис. 2).

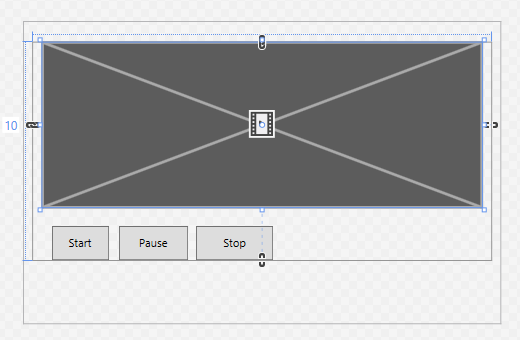


Рис. 2. Додавання елементів на форму

Додамо на форму три елементи Splitter для регулювання гучності, швидкості відтворення та переводу в довільну точку відео, а також ще одну кнопку для вибору відео-файлу (рис. 3).

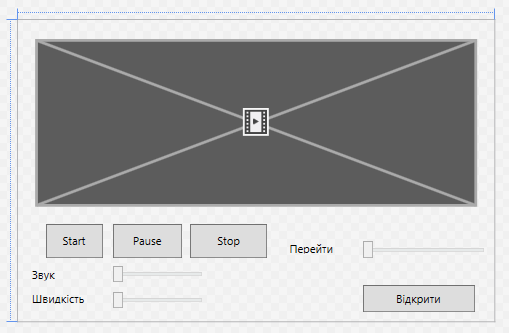


Рис. 3. Додавання елементів на форму

За допомогою редактора розмітки XAML, задамо кольори елементів, а також їх фонів (рис. 4).

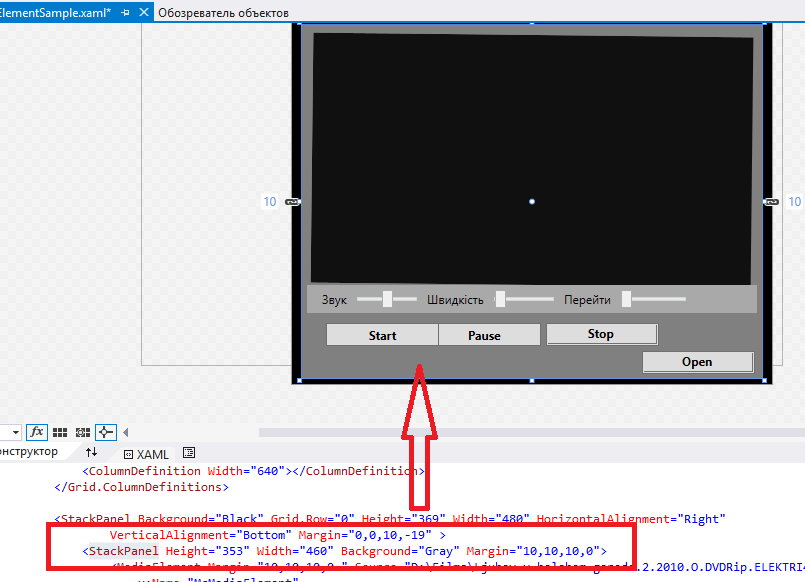


Рис. 4. Зміна кольору елементів за допомогою XAML

Для покращення дизайну замінимо кнопки курування на іконки і проставимо їм у відповідність зовнішні графічні файли, а також обробники натиску на них: (рис 5).

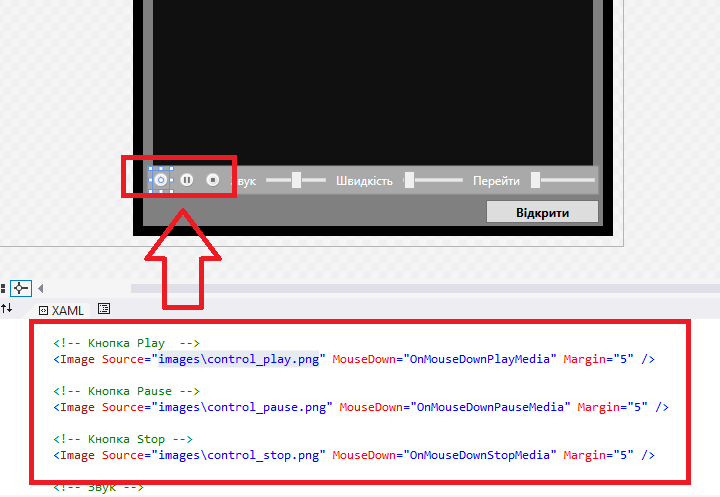
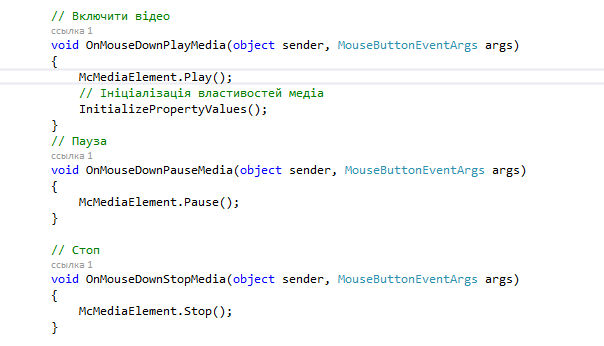
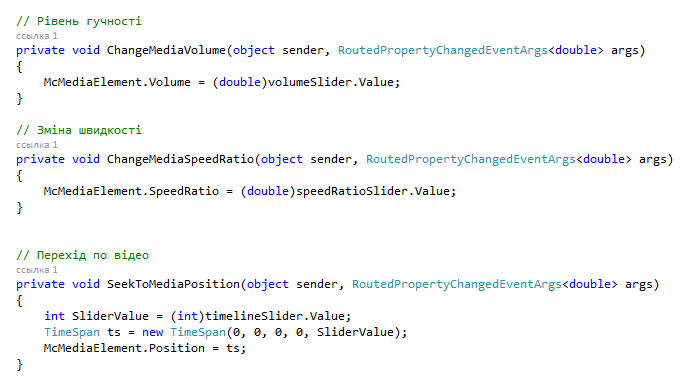


Рис. 5. Заміна кнопок на іконки

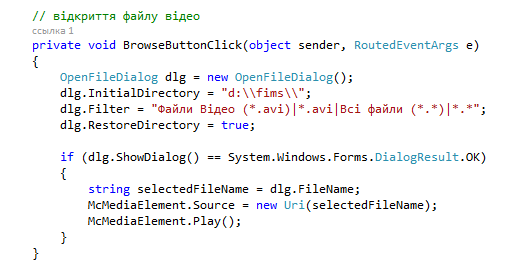
Реалізуємо методи для опрацювання подій натиску на іконки:



Також потрібно опрацювати методи зміни елементів: рівня звуку, швидкості програвання, позиції програвання:



А також опрацюємо натиск по кнопці відкриття відео-файла:



**3. Текст програми**

//MediaElementSample.xaml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Windows.Forms;

namespace MediaSamples

{

/// <summary>

/// Interaction logic for Page1.xaml

/// </summary>

public partial class Page1 : Page

{

public Page1()

{

InitializeComponent();

}

private void Element\_MediaEnded(object sender, RoutedEventArgs e)

{

}

// Включити відео

void OnMouseDownPlayMedia(object sender, MouseButtonEventArgs args)

{

McMediaElement.Play();

// Ініціалізація властивостей медіа

InitializePropertyValues();

}

// Пауза

void OnMouseDownPauseMedia(object sender, MouseButtonEventArgs args)

{

McMediaElement.Pause();

}

// Стоп

void OnMouseDownStopMedia(object sender, MouseButtonEventArgs args)

{

McMediaElement.Stop();

}

// Рівень гучності

private void ChangeMediaVolume(object sender, RoutedPropertyChangedEventArgs<double> args)

{

McMediaElement.Volume = (double)volumeSlider.Value;

}

// Зміна швидкості

private void ChangeMediaSpeedRatio(object sender, RoutedPropertyChangedEventArgs<double> args)

{

McMediaElement.SpeedRatio = (double)speedRatioSlider.Value;

}

// Ініціалізація макс. точки повзунка кількістю мілісекунд кліпу

private void Element\_MediaOpened(object sender, EventArgs e)

{

timelineSlider.Maximum = McMediaElement.NaturalDuration.TimeSpan.TotalMilliseconds;

}

// Стоп медіа, коли воно завершиться

private void Element\_MediaEnded(object sender, EventArgs e)

{

McMediaElement.Stop();

}

// Перехід по відео

private void SeekToMediaPosition(object sender, RoutedPropertyChangedEventArgs<double> args)

{

int SliderValue = (int)timelineSlider.Value;

TimeSpan ts = new TimeSpan(0, 0, 0, 0, SliderValue);

McMediaElement.Position = ts;

}

void InitializePropertyValues()

{

// Встановлення початкових значень швидкості та рівня звуку

McMediaElement.Volume = (double)volumeSlider.Value;

McMediaElement.SpeedRatio = (double)speedRatioSlider.Value;

}

// відкриття файлу відео

private void BrowseButtonClick(object sender, RoutedEventArgs e)

{

OpenFileDialog dlg = new OpenFileDialog();

dlg.InitialDirectory = "d:\\fims\\";

dlg.Filter = "Файли Відео (\*.avi)|\*.avi|Всі файли (\*.\*)|\*.\*";

dlg.RestoreDirectory = true;

if (dlg.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

string selectedFileName = dlg.FileName;

McMediaElement.Source = new Uri(selectedFileName);

McMediaElement.Play();

}

}

}

}

//MediaElementSample.xaml

<Page x:Class="MediaSamples.Page1"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

>

<Grid Width="640" Height="480" VerticalAlignment="Top" HorizontalAlignment="Center" >

<Grid.RowDefinitions>

<RowDefinition Height="480"></RowDefinition>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="640"></ColumnDefinition>

</Grid.ColumnDefinitions>

<StackPanel Background="Black" Grid.Row="0" Grid.Column="1" Height="340" Width="480" HorizontalAlignment="Right"

VerticalAlignment="Bottom" Margin="10,10,10,10" >

<StackPanel Height="320" Width="460" Background="Gray" Margin="0,10,0,0">

<MediaElement Margin="10,10,10,0 " Source="D:\Films\Ljubov.v.bolshom.gorode.2.2010.O.DVDRip.ELEKTRI4KA.[uniongang.tv].avi"

Name="McMediaElement"

Width="450" Height="250" LoadedBehavior="Manual" UnloadedBehavior="Stop" Stretch="Fill"

MediaOpened="Element\_MediaOpened" MediaEnded="Element\_MediaEnded"/>

<StackPanel Background="DarkGray" HorizontalAlignment="Center" Width="450" Orientation="Horizontal">

<!-- Кнопка Play -->

<Image Source="images\control\_play.png" MouseDown="OnMouseDownPlayMedia" Margin="5" />

<!-- Кнопка Pause -->

<Image Source="images\control\_pause.png" MouseDown="OnMouseDownPauseMedia" Margin="5" />

<!-- Кнопка Stop -->

<Image Source="images\control\_stop.png" MouseDown="OnMouseDownStopMedia" Margin="5" />

<!-- Звук -->

<TextBlock Foreground="White" VerticalAlignment="Center" Margin="5" >Звук</TextBlock>

<Slider Name="volumeSlider" VerticalAlignment="Center" ValueChanged="ChangeMediaVolume"

Minimum="0" Maximum="1" Value="0.5" Width="70"/>

<!-- Швидкість відео -->

<TextBlock Foreground="White" Margin="5" VerticalAlignment="Center">Швидкість</TextBlock>

<Slider Name="speedRatioSlider" VerticalAlignment="Center" ValueChanged="ChangeMediaSpeedRatio"

Value="1" Width="70" />

<!-- Прокрутка відео -->

<TextBlock Foreground="White" Margin="5" VerticalAlignment="Center">Перейти</TextBlock>

<Slider Name="timelineSlider" Margin="5" ValueChanged="SeekToMediaPosition" Width="70"/>

</StackPanel>

</StackPanel>

</StackPanel>

<StackPanel HorizontalAlignment="Center" VerticalAlignment="Bottom" Grid.Row="1" Grid.Column="0" Grid.ColumnSpan="3"></StackPanel>

<Button Height="23" x:Name="BrowseButton" Width="113" Grid.Column="1" HorizontalAlignment="Right"

Margin="0,0,24,23.04" VerticalAlignment="Bottom"

FontWeight="Bold" Click="BrowseButtonClick" Content="Відкрити"/>

</Grid>

</Page>

**4. Результат виконання програми**

Запустимо програму на виконання та допомогою кнопки «Відкрити» спробуємо перегляти вибраний нами відео-файл (рис. 6):

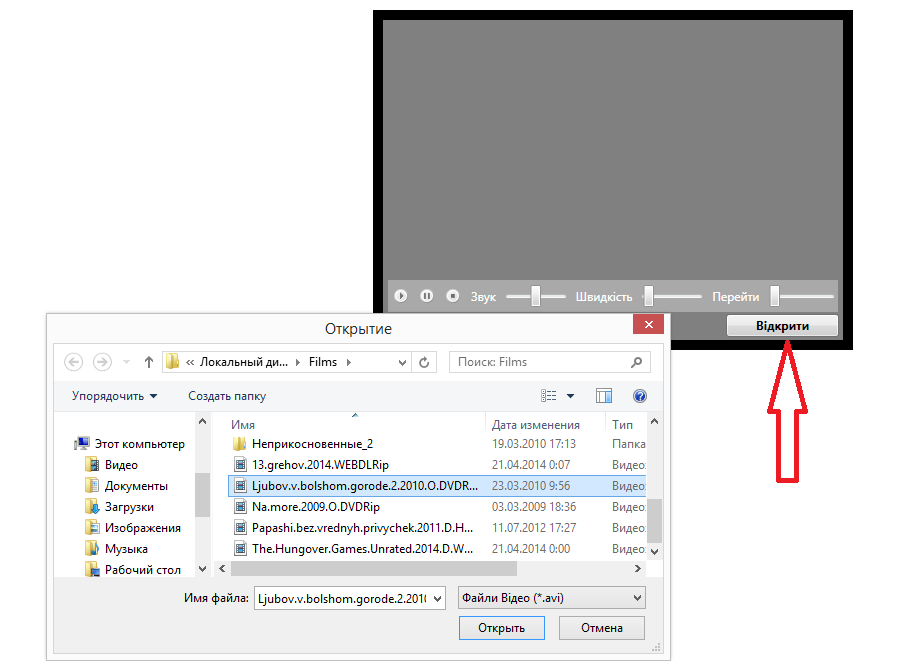


Рис. 6. Відкриття відео-файлу

Спробуємо перейти в довільну точку відео, а також протестуємо кнопки «Пауза» та «Стоп», зміну гучності – все працює коректно (рис. 7):

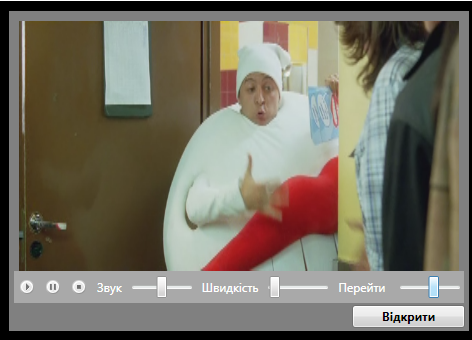


Рис. 7. Тестування функціоналу програвача

## ВИСНОВКИ

В ході даної лабораторної роботи я навчився на практиці створювати програмні засоби для програвання відео-файлів за допомогою технології WPF.