**Roadmap za razvoj multimedijskega predvajalnika v WPF**

**Vaja 1: XAML in osnove C#**

**Cilj:** Oblikovanje osnovnega GUI-ja.

1. **Koraki:**
   * Razdelite glavno okno z upravljalci postavitve (Grid, StackPanel, DockPanel).
   * Dodajte:
     + **Menu** za aplikacijski meni (uporabite MenuItem za podmenije).
     + **ListView** za seznam predvajanj.
     + **TextBox** za ime trenutne datoteke.
     + **MediaElement** za predvajanje vsebine.
     + **S2lider** za napredek predvajanja.
     + **Label** za prikaz časa.
     + **Button** z ikonami za nadzor predvajanja.
   * Poskrbite za prilagodljivost gradnikov glede na velikost okna.
2. **Pomembne funkcije:**
   * MenuItem.Click za izhod (uporabite Application.Current.Shutdown()).
   * ListView.MouseDoubleClick za prikaz izbranega elementa v MessageBox.

**Vaja 2: Podatkovna struktura in vezava podatkov**

**Cilj:** Implementacija podatkovne strukture in povezava z vmesnikom.

1. **Koraki:**
   * Ustvarite razred, ki predstavlja multimedijsko datoteko, z lastnostmi:
     + Zvrst, pot do datoteke, slika, naslov, dolžina, avtor.
   * Implementirajte get in set metode z validacijo.
   * Povežite ListView z zbirko objektov (uporabite ObservableCollection).
   * Dodajte gumb za resetiranje imena v TextBox.
   * Ob dvokliku elementa predvajajte datoteko v MediaElement.
2. **Pomembne funkcije:**
   * Uporabite INotifyPropertyChanged za sinhronizacijo podatkov med modelom in UI.
   * Vezava podatkov z {Binding}.

**Vaja 3: Prikaz in shranjevanje podatkov**

**Cilj:** Izdelava kompleksnejšega prikaza seznama in shranjevanje v datoteke.

1. **Koraki:**
   * Oblikujte DataTemplate za prikaz slik in podatkov v ListView.
   * Označite trenutno predvajano datoteko z DataTrigger.
   * Implementirajte shranjevanje in nalaganje seznama (XML/JSON serializacija).
2. **Pomembne funkcije:**
   * SaveFileDialog in OpenFileDialog za uvoz/izvoz.
   * XmlSerializer ali JsonSerializer za delo s podatki.

**Vaja 4: MVVM vzorec**

**Cilj:** Dodajanje arhitekture MVVM za boljšo modularnost.

1. **Koraki:**
   * Ustvarite razred ViewModel in implementirajte:
     + Dodajanje (Add), odstranjevanje (Remove), urejanje (Edit).
     + Uporabite ICommand za povezavo dogodkov z logiko.
   * Onemogočite možnosti menija, če ni izbran element.
2. **Pomembne funkcije:**
   * Uporabite RelayCommand ali lasten ICommand razred.

**Vaja 5: Dodatna okna**

**Cilj:** Dodajanje modalnih in nemodalnih oken za nastavitve in urejanje.

1. **Koraki:**
   * **Okno Nastavitve:**
     + Dodajte TabControl za urejanje zvrsti (dodajanje, urejanje, brisanje).
     + Shranjevanje zvrsti v nastavitve aplikacije (uporabite ApplicationSettings).
   * **Okno Dodajanje/Urejanje:**
     + Omogočite vnos vseh podatkov iz razreda.
     + Podprite nalaganje datotek z OpenFileDialog.
2. **Pomembne funkcije:**
   * Owner lastnost za povezavo oken z glavnim oknom.

**Vaja 6: Uporabniški gradnik**

**Cilj:** Dodajanje lastnega gradnika za nadzor predvajanja.

1. **Koraki:**
   * Ustvarite UserControl za gumbe (predvajanje, pavza, naključni vrstni red).
   * Povežite dogodke s preostalim UI prek lastnih dogodkov (definirajte EventArgs).
   * Nadomestite obstoječe gumbe z novim gradnikom.
2. **Pomembne funkcije:**
   * Uporabite DependencyProperty za interakcijo z drugimi deli aplikacije.

**Vaja 7: Deljenje virov**

**Cilj:** Poenotitev izgleda aplikacije.

1. **Koraki:**
   * Ustvarite ResourceDictionary za stile.
   * Definirajte stile z lastnostjo BasedOn za dedovanje.
   * Dodajte Trigger za dinamične spremembe.
2. **Pomembne funkcije:**
   * Uporabite DynamicResource za prilagodljivost.

**Vaja 8: Časovniki in animacije**

**Cilj:** Dodajanje časovnikov za posodobitev UI in animacij za izboljšanje UX.

1. **Koraki:**
   * Uporabite DispatcherTimer za posodabljanje drsnika napredka.
   * Dodajte animacije za gumbe (Storyboard).
   * Implementirajte prehod med elementi.
2. **Pomembne funkcije:**
   * Uporabite BeginStoryboard in DoubleAnimation.

**Na splošno:**

* **Orodja:** Visual Studio, Blend for Visual Studio.
* **Viri:** Dokumentacija WPF, spletni vodiči (npr. Microsoft Learn).
* **Pomoč:** Debugging, uporaba Breakpoints.

Če potrebujete podrobno kodo za posamezen korak, mi sporočite. 😊

Tu je še za datoteke

Seveda, datoteke s slikami in videoposnetki lahko upravljate v WPF s pomočjo razreda OpenFileDialog in drugih funkcij za branje datotek. Tukaj je, kako lahko to vključite v vaš projekt:

### Postopek za delo s slikami in videoposnetki:

1. **Branje datotek iz datotečnega sistema**:
   * Uporabite OpenFileDialog za izbiro datotek.
   * Filtrirajte datoteke glede na vrsto (npr. slike: .jpeg, .png; videoposnetki: .mp4, .avi).
2. **Shranjevanje poti do datotek v seznam**:
   * Namesto da neposredno prikažete datoteke, shranite njihove poti v seznam (List<string>).
   * Uporabite ta seznam kot podatkovni vir za ListView.
3. **Prikaz podatkov v ListView**:
   * S pomočjo vezave podatkov (Data Binding) povežite seznam poti z ListView.
4. **Urejanje ali spreminjanje datotek**:
   * Iz seznama uporabite izbrane poti za zamenjavo slik ali videoposnetkov.

### Implementacija:

#### 1. **Uvoz datotek v** ListView**:**

private List<string> filePaths = new List<string>();

private void ImportFiles()

{

OpenFileDialog openFileDialog = new OpenFileDialog

{

Multiselect = true, // Omogoči izbiro več datotek

Filter = "Media Files|\*.mp4;\*.avi;\*.jpeg;\*.png|All Files|\*.\*"

};

if (openFileDialog.ShowDialog() == true)

{

foreach (string file in openFileDialog.FileNames)

{

filePaths.Add(file); // Shrani pot do datoteke v seznam

}

UpdateListView(); // Posodobi prikaz

}

}

private void UpdateListView()

{

listView.ItemsSource = null; // Ponastavi vir podatkov

listView.ItemsSource = filePaths; // Posodobi z novimi podatki

}

#### 2. **Zamenjava slike ali videa:**

private void ReplaceFile(int index)

{

OpenFileDialog openFileDialog = new OpenFileDialog

{

Filter = "Media Files|\*.mp4;\*.avi;\*.jpeg;\*.png|All Files|\*.\*"

};

if (openFileDialog.ShowDialog() == true)

{

filePaths[index] = openFileDialog.FileName; // Zamenja obstoječo datoteko

UpdateListView(); // Osveži prikaz

}

}

#### 3. **Izbriši iz seznama:**

private void RemoveFile(int index)

{

if (index >= 0 && index < filePaths.Count)

{

filePaths.RemoveAt(index); // Odstrani datoteko iz seznama

UpdateListView(); // Posodobi prikaz

}

}

#### 4. **Prikaz izbranega videa/slike:**

Za prikaz lahko uporabite MediaElement za videe in Image za slike.

private void DisplayMedia(string path)

{

if (path.EndsWith(".mp4") || path.EndsWith(".avi"))

{

mediaElement.Source = new Uri(path); // Predvajaj video

mediaElement.Play();

}

else if (path.EndsWith(".jpeg") || path.EndsWith(".png"))

{

BitmapImage bitmap = new BitmapImage(new Uri(path));

imageControl.Source = bitmap; // Prikaži sliko

}

}

#### 5. **Povežite z dogodki v** ListView**:**

V ListView lahko dodate dogodke, kot je dvoklik za izbiro datoteke.

<ListView x:Name="listView" MouseDoubleClick="ListView\_MouseDoubleClick" />

private void ListView\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{

if (listView.SelectedIndex >= 0)

{

DisplayMedia(filePaths[listView.SelectedIndex]); // Prikaže izbrano datoteko

}

}

### Kako to deluje:

1. **Dodajanje datotek**:
   * Gumb "Uvozi" omogoča izbiro datotek in njihovo dodajanje na seznam.
2. **Upravljanje seznama**:
   * Gumb "Zamenjaj" zamenja datoteko z novo, gumb "Odstrani" jo izbriše.
3. **Predvajanje/prikaz**:
   * Dvoklik na datoteko v ListView prikaže izbrano datoteko (sliko ali video).

S tem pristopom je vaša aplikacija modularna in omogoča enostavno upravljanje datotek.