3iLINGENIEURS </les sciences informatiques>	**Technologie .Net Maui** (3iL I3-I2FA - B. Chervy) **TP Maui N° 2**	**Année : 2023/2024** Temps : 2 sécances Nombre de page : 4

TP .Net Maui n° 2 (Suite) : Application à notes multiples

- AppShell.xaml :

```
<?xml version="1.0" encoding="UTF-8" ?>
<Shell
    x:Class="TP1.AppShell"
    xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:view="clr-namespace:TP1.Views"
    Shell.FlyoutBehavior="Disabled">

    <!--<ShellContent
        Title="Home"
        ContentTemplate="{DataTemplate local:MainPage}"
        Route="MainPage" />-->
    <TabBar>
        <!--<ShellContent
            Title="Notes"
            ContentTemplate="{DataTemplate view:Note}"
            Icon="icon_notes" />-->
        <ShellContent Title="Liste Notes"
            ContentTemplate="{DataTemplate view:ListeNotes}"
                      Icon="icon_notes" />

        <ShellContent
            Title="A Propos"
            ContentTemplate="{DataTemplate view:APropos}"
            Icon="icon_about" />
    </TabBar>


</Shell>
```

- Note.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="TP1.Views.Note"
             Title="Note">
    <VerticalStackLayout Spacing="10" Margin="5">
        <Editor x:Name="TextEditor"
                Placeholder="Taper votre note"
                HeightRequest="100"
                Text="{Binding Text}" />
        <Grid ColumnDefinitions="*,*" ColumnSpacing="4">
            <Button Text="Enregistrer"
                    Clicked="ButtonEnregistrer_Clicked" />
            <Button Grid.Column="1"
                    Text="Supprimer"
                    Clicked="ButtonSupprimer_Clicked" />
        </Grid>
    </VerticalStackLayout>
```

```
</ContentPage>
```

- Note.xaml.cs

```
namespace TP1.Views;
[QueryProperty(nameof(ItemId), nameof(ItemId))]
public partial class Note : ContentPage
{
    string _fileName = Path.Combine(FileSystem.AppDataDirectory, "notes.txt");

    public string ItemId
    {
        set { ChargerNote(value); }
    }

    public Note()
      {
             InitializeComponent();
        //if (File.Exists(_fileName))
        //    TextEditor.Text = File.ReadAllText(_fileName);

        string appDataPath = FileSystem.AppDataDirectory;
        //string sFileName = "notes.txt";
        //ChargerNote(Path.Combine(appDataPath, sFileName));
        string sFileName = $"{Path.GetRandomFileName()}.notes.txt";
        ChargerNote(Path.Combine(appDataPath, sFileName));
    }

    private void ChargerNote(string fileName)
    {
        Models.CNote noteModel = new Models.CNote();
        noteModel.Filename = fileName;
        if (File.Exists(fileName))
        {
            noteModel.Date = File.GetCreationTime(fileName);
            noteModel.Text = File.ReadAllText(fileName);
        }
        BindingContext = noteModel;
    }

    private async void ButtonEnregistrer_Clicked(object sender, EventArgs e)
      {
        //File.WriteAllText(_fileName, TextEditor.Text);
        if (BindingContext is Models.CNote note)
            File.WriteAllText(note.Filename, TextEditor.Text);
        await Shell.Current.GoToAsync("..");
    }

    private async void ButtonSupprimer_Clicked(object sender, EventArgs e)
      {
        //if (File.Exists(_fileName))
        //    File.Delete(_fileName);
        //TextEditor.Text = string.Empty;
        if (BindingContext is Models.CNote note)
        {
            if (File.Exists(note.Filename))
                File.Delete(note.Filename);
        }
        await Shell.Current.GoToAsync("..");

    }
}
```

- APropos.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
```

```
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="TP1.Views.APropos"
             xmlns:models="clr-namespace:TP1.Models"
             Title="APropos">
    <!-- BindingContext => Liaison entre la vue (cette page) et le model (classe CApropos) -->
    <ContentPage.BindingContext>
        <models:CApropos />
    </ContentPage.BindingContext>
    <VerticalStackLayout Spacing="10" Margin="10">
        <Image Source="logo_groupe3il.jpg"
                  SemanticProperties.Description="Cette application est une production 3iL"
                   HeightRequest="64" HorizontalOptions="CenterAndExpand"/>
        <!--<Label FontSize="22" FontAttributes="Bold" Text="Notes 3iL" VerticalOptions="End"
                                                       TextColor="{StaticResource Secondary}"/>
        <Label FontSize="22" Text="v1.0" VerticalOptions="End" TextColor="{StaticResource
                                                                                    Secondary}"/>
        <Label Text="Application XAML/C# écrite en .NET MAUI."
    TextColor="{StaticResource Primary}"/>-->
        <Label FontSize="22" FontAttributes="Bold" Text="{Binding Title}" VerticalOptions="End"
                                                    TextColor="{StaticResource Secondary}"/>
        <Label FontSize="22" Text="{Binding Version}" VerticalOptions="End"
                                                TextColor="{StaticResource Secondary}"/>
        <Label Text="{Binding Message}" TextColor="{StaticResource Primary}"/>
        <Button Text="En savoir plus sur 3iL" Clicked="APropos_Clicked" />
    </VerticalStackLayout>

</ContentPage>
```

- APropos.xaml.cs

```
namespace TP1.Views;

public partial class APropos : ContentPage
{
      public APropos()
      {
             InitializeComponent();
      }

      private async void APropos_Clicked(object sender, EventArgs e)
      {
        if (BindingContext is Models.CApropos apropos)
        {
            await Launcher.Default.OpenAsync(apropos.MoreInfoUrl);
        }

        //await Launcher.Default.OpenAsync("https://www.3il-ingenieurs.fr/");
    }
}
```

- CPropos.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace TP1.Models
{
    internal class CApropos
    {
        public string Title => "Notes 3iL - " + AppInfo.Name;
        public string Version => AppInfo.VersionString;
        public string MoreInfoUrl => "https://www.3il-ingenieurs.fr/";
        public string Message => "Application XAML/C# écrite en .NET MAUI.";
    }
}
```

- Note.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="TP1.Views.Note"
             Title="Note">
    <VerticalStackLayout Spacing="10" Margin="5">
        <Editor x:Name="TextEditor"
                Placeholder="Taper votre note"
                HeightRequest="100"
                Text="{Binding Text}" />
        <Grid ColumnDefinitions="*,*" ColumnSpacing="4">
            <Button Text="Enregistrer"
                    Clicked="ButtonEnregistrer_Clicked" />
            <Button Grid.Column="1"
                    Text="Supprimer"
                    Clicked="ButtonSupprimer_Clicked" />
        </Grid>
    </VerticalStackLayout>

</ContentPage>
```

- Note.xaml.cs

```
namespace TP1.Views;
[QueryProperty(nameof(ItemId), nameof(ItemId))]
public partial class Note : ContentPage
{
    string _fileName = Path.Combine(FileSystem.AppDataDirectory, "notes.txt");

    public string ItemId
    {
        set { ChargerNote(value); }
    }

    public Note()
      {
             InitializeComponent();
        //if (File.Exists(_fileName))
        //    TextEditor.Text = File.ReadAllText(_fileName);

        string appDataPath = FileSystem.AppDataDirectory;
        //string sFileName = "notes.txt";
        //ChargerNote(Path.Combine(appDataPath, sFileName));
        string sFileName = $"{Path.GetRandomFileName()}.notes.txt";
        ChargerNote(Path.Combine(appDataPath, sFileName));
    }

    private void ChargerNote(string fileName)
    {
        Models.CNote noteModel = new Models.CNote();
        noteModel.Filename = fileName;
        if (File.Exists(fileName))
        {
            noteModel.Date = File.GetCreationTime(fileName);
            noteModel.Text = File.ReadAllText(fileName);
        }
        BindingContext = noteModel;
    }

    private async void ButtonEnregistrer_Clicked(object sender, EventArgs e)
```

```
      {
        //File.WriteAllText(_fileName, TextEditor.Text);
        if (BindingContext is Models.CNote note)
            File.WriteAllText(note.Filename, TextEditor.Text);
        await Shell.Current.GoToAsync("..");
    }

    private async void ButtonSupprimer_Clicked(object sender, EventArgs e)
      {
        //if (File.Exists(_fileName))
        //    File.Delete(_fileName);
        //TextEditor.Text = string.Empty;
        if (BindingContext is Models.CNote note)
        {
            if (File.Exists(note.Filename))
                File.Delete(note.Filename);
        }
        await Shell.Current.GoToAsync("..");

    }
}
```

- CNote.cs

```
using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace TP1.Models
{
    internal class CListeNotes
    {
        public ObservableCollection<CNote> CollNotes { get; set; } = new
ObservableCollection<CNote>();
        public CListeNotes() => ChargerNotes();
        public void ChargerNotes()
        {
            CollNotes.Clear();
            string appDataPath = FileSystem.AppDataDirectory;
            IEnumerable<CNote> listenotes = Directory.EnumerateFiles(appDataPath, "*.notes.txt")
                                        .Select(filename => new CNote()
                                        {
                                            Filename = filename,
                                            Text = File.ReadAllText(filename),
                                            Date = File.GetCreationTime(filename)
                                        })
                                        .OrderBy(note => note.Date);
            foreach (CNote note in listenotes)
                CollNotes.Add(note);
        }
    }
}
```

- ListeNotes.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="TP1.Views.ListeNotes"
             Title="Liste Notes">
    <!--<VerticalStackLayout>
        <Label
            Text="Welcome to .NET MAUI!"
            VerticalOptions="Center"
            HorizontalOptions="Center" />
    </VerticalStackLayout>-->
    <ContentPage.ToolbarItems>
        <ToolbarItem Text="Ajouter" Clicked="Ajouter_Clicked" IconImageSource="{FontImage
                Glyph='+', Color=White, Size=22}" />
    </ContentPage.ToolbarItems>

    <CollectionView x:Name="CollectionDeNotes"
                        ItemsSource="{Binding CollNotes}"
                        Margin="20"
                        SelectionMode="Single"
                        SelectionChanged="CollectionDeNotes_SelectionChanged">
        <CollectionView.ItemsLayout>
            <LinearItemsLayout Orientation="Vertical" ItemSpacing="10" />
        </CollectionView.ItemsLayout>
        <CollectionView.ItemTemplate>
            <DataTemplate>
                <StackLayout>
                    <Label Text="{Binding Text}" FontSize="22"/>
                    <Label Text="{Binding Date}" FontSize="14" TextColor="Silver"/>
                </StackLayout>
            </DataTemplate>
        </CollectionView.ItemTemplate>

    </CollectionView>
</ContentPage>
```

- ListeNotes.xaml.cs

```
//using static Android.Provider.ContactsContract.CommonDataKinds;

namespace TP1.Views;

public partial class ListeNotes : ContentPage
{
      public ListeNotes()
      {
            InitializeComponent();
        Routing.RegisterRoute($"{nameof(Note)}", typeof(Note));
        BindingContext = new Models.CListeNotes();

    }

    protected override void OnAppearing()
    {
        ((Models.CListeNotes)BindingContext).ChargerNotes();
    }
    private async void CollectionDeNotes_SelectionChanged(object sender,
SelectionChangedEventArgs e)
    {
        //await Shell.Current.GoToAsync(nameof(Note));
        if (e.CurrentSelection.Count != 0)
        {
            var note = (Models.CNote)e.CurrentSelection[0];
            await
Shell.Current.GoToAsync($"{nameof(Note)}?{nameof(Note.ItemId)}={note.Filename}");
            CollectionDeNotes.SelectedItem = null;
        }
    }

      private async void Ajouter_Clicked(object sender, EventArgs e)
      {
        await Shell.Current.GoToAsync(nameof(Note));
    }
}
```

- CListeNotes.cs

```
using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace TP1.Models
{
    internal class CListeNotes
    {
        public ObservableCollection<CNote> CollNotes { get; set; } = new
ObservableCollection<CNote>();
        public CListeNotes() => ChargerNotes();
        public void ChargerNotes()
        {
            CollNotes.Clear();
            string appDataPath = FileSystem.AppDataDirectory;
            IEnumerable<CNote> listenotes = Directory.EnumerateFiles(appDataPath, "*.notes.txt")
                                        .Select(filename => new CNote()
                                        {
                                            Filename = filename,
                                            Text = File.ReadAllText(filename),
                                            Date = File.GetCreationTime(filename)
                                        })
                                        .OrderBy(note => note.Date);
            foreach (CNote note in listenotes)
                CollNotes.Add(note);
        }
    }
}
```