# Charte Graphique

## Page d’arrivée :

<MauiSplashScreen Include="Resources\Splash\splash.svg" Color="White" BaseSize="128,128" /> avec logo INRAE à minima + éventuel logo projet



+ mettre en dessous le nom (et/ou le logo) de l’application utilisée

## Polices de caractères :



## Codes couleurs :



## Couleurs :

A placer dans Styles/Colors.xaml

<Color x:Key="InraePrimary">#00a3a6</Color>

<Color x:Key="InraeSecondary">#66c1bf</Color>

<Color x:Key="InraeTertiary">#275662</Color>

<Color x:Key="InraeGray">#CCCCCC</Color>

<Color x:Key="InraeDelete">#C70039</Color>

## Ajout de FontAwesome pour les icônes

* Ajouter dans MauiProgram.cs l’appel aux fichiers FontAwesome pour la génération des icônes

fonts.AddFont("fa-brands.ttf", "FABrands");

fonts.AddFont("fa-regular.ttf", "FARegular");

fonts.AddFont("fa-solid.ttf", "FASolid");

* Déplacer ces fichiers dans le répertoire Resources/Fonts
* Créer dans Model les classes correspondant à ces fonts : FaBrandIcons.cs, FaRegularIcons.cs, FaSolidIcons.cs et utiliser <https://andreinitescu.github.io/IconFont2Code/> pour générer le code C# correspondant

## Boutons :

* Créer une classe InraeButtons à placer dans Model utilisant les classes FontAwesome

namespace MonApplicationMaui.Model

{

 internal class InraeButtons

 {

 public const string Plus = FaSolidIcons.Plus + " Ajouter";

 public const string Delete = FaSolidIcons.Trash + " Supprimer";

 public const string Cancel = FaSolidIcons.Trash + " Annuler";

 …

 }

}

### Ajouter/Valider :

A placer dans Styles/Styles.xaml

<Style x:Key="btnValidate" TargetType="Button">

 <Setter Property="BackgroundColor" Value="{StaticResource InraePrimary}" />

 <Setter Property="TextColor" Value="White" />

 <Setter Property="CornerRadius" Value="15" />

 <Setter Property="Padding" Value="14,10" />

</Style>

A placer dans Views/…xaml

<Button

 Grid.Column="0"

 Margin="10"

 Command="{Binding …Command}"

 FontSize="20"

 Style="{StaticResource btnValidate}"

 Text="{x:Static model:InraeButtons.Plus}" />



### Annuler :

A placer dans Styles/Styles.xaml

<Style x:Key="btnCancel" TargetType="Button">

 <Setter Property="BackgroundColor" Value="{StaticResource InraePrimary}" />

 <Setter Property="TextColor" Value="White" />

 <Setter Property="CornerRadius" Value="15" />

 <Setter Property="Padding" Value="14,10" />

</Style>

A placer dans Views/…xaml

<Button

 Grid.Column="0"

 Margin="10"

 Command="{Binding …Command}"

 FontSize="20"

 Style="{StaticResource btnCancel}"

 Text="{x:Static model:InraeButtons.Cancel}" />

 

### Supprimer :

A placer dans Styles/Styles.xaml

<Style x:Key="btnDelete" TargetType="Button">

 <Setter Property="BackgroundColor" Value="{StaticResource InraeDelete}" />

 <Setter Property="TextColor" Value="White" />

 <Setter Property="CornerRadius" Value="15" />

 <Setter Property="Padding" Value="14,10" />

</Style>

A placer dans Views/…xaml

<Button

 Grid.Column="0"

 Margin="10"

 Command="{Binding …Command}"

 FontSize="20"

 Style="{StaticResource btnDelete}"

 Text="{x:Static model:InraeButtons.Delete}" />



## Menus et liens

Les menus seront utilisés avec le flyout dans la définition du fichier AppShell.xaml

<Shell

 x:Class="MauiApp1.AppShell"

 xmlns="http://schemas.microsoft.com/dotnet/2021/maui"

 xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

 xmlns:local="clr-namespace:MauiApp1.View"

 Shell.FlyoutBehavior="Flyout">

On utilisera des fichiers svg afin d’associer des images avec les liens principaux. Les fichiers svg seront à incorporer dans Resources/Images et à intégrer dans les attributs des FlyoutItem.ShellContent.

<FlyoutItem FlyoutDisplayOptions="AsMultipleItems">

 <ShellContent

 Title="Home"

 ContentTemplate="{DataTemplate local:AccueilView}"

 Icon="home" />

Ici on a un fichier home.svg installé dans notre répertoire. Il est listé en utilisant l’attribut Icon



