



TP WPF

Les dataGrids, les JumpLists, les input bindings

Lemettre Arnaud

Version 1.0

10 Pages

16/08/2010

EPITA-MTI2014-WPF-TP-DataGrid-JumpList-Input Binding

```
<Button Content="Prof-MTI">
  <Button.Triggers>
    <EventTrigger RoutedEvent="Button.Click">
      <EventTrigger.Actions>
        <BeginStoryboard Storyboard="{StaticResource StudentLearning}"/>
      </EventTrigger.Actions>
    </EventTrigger>
  </Button.Triggers>
</Button>
```

Propriétés du document

Auteur	Lemettre Arnaud
Version	1.0
Nombre de pages	10
Références	EPITA-MTI2014-WPF-TP-DataGrid-JumpList-Input Binding

Historique du document

Date de vision	Version	Auteur	Changements
16/08/2010	0.1	Lemettre	création

Site de référence

Description	url
Site MTI	
Blog MTI	

Sommaire

Introduction	4
Partie 1 : DataGrid	5
Renseignements	5
Pré-requis.....	5
La vue.....	5
Binder la liste	5
Appliquer un style.....	5
Personnaliser l’affichage.....	5
Travail à faire	6
Partie 2 : JumpList.....	7
Renseignements	7
JumpList statique	7
Manipulation de la JumpList.....	7
Rajout de bouton	7
Pré-requis.....	7
Ajout des boutons d’actions	7
Travail à faire	8
Partie 3 : Input Binding	9
Renseignements	9
La vue	9
Les inputs	9
Travail à faire	9
Modalités de rendu	10

Introduction

Le but de ce TP est de se familiariser avec la technologie WPF de Microsoft.

Tout au long de ce TP, les notions suivantes seront abordées :

- Les dataGrids
- Les JumpLists
- Les inputs Binding

Ce travail est à faire individuellement, tout code similaire sur deux personnes sera considéré comme un travail non rendu et non négociable, il est recommandé de lire le sujet jusqu'à la fin.

Toute mention de login_l fait référence à votre login EPITA.

Ce TP ne demande pas plus de 4h de travail.

Bonne chance ;)

Partie 1 : DataGrid

Renseignements

Dans cette partie nous allons manipuler la dataGrid en parcourant un ensemble de fonctionnalités.

Pré-requis

Dans un premier temps, il vous faut créer 2 classes dans la couche DBO :

- 1 classe User
 - o Name
 - o Firstname
 - o Mail
 - o Gender
 - o Present
- 1 énumération EnumGender
 - o Male
 - o Female

La vue

Puis rajouter sur la vue une dataGrid.

Binder la liste

Créer une liste de User, que vous initialiserez dans une fonction avec quelques objets User. Le but étant de voir des différences d'affichage. Faites varier les valeurs à l'intérieur de ces objets.

Pour binder la liste utiliser la dataContext de la dataGrid.

A partir de ce moment sans rien changer, si vous lancez votre application vous devez avoir une liste de personne dans votre dataGrid au chargement de la fenêtre, si vous avez mis votre fonction d'initialisation dans le constructeur de la fenêtre.

Appliquer un style

Pour commencer, nous allons changer le style des headers de la grid. Vous allez donc créer un style de cette manière :

- Hauteur : 30
- Le fond : lightBlue
- Couleur du texte : blue
- Taille du texte : 18
- Au survol du header un tool tip disant : click to sort

De la même manière, appliquer une couleur alternative sur les lignes de la datagrid.

Personnaliser l'affichage

N'afficher maintenant que les colonnes nom et mail.

Travail à faire

Vous devez créer une solution visual studio, avec un projet de type WPF/C#. Vous pouvez créer cette solution avec le framework 4.5.

Le nom de la solution devra se nommer login_ITP4

Le nom du projet devra être WPF_DataGrid

Partie 2 : JumpList

Renseignements

Dans cette partie, nous allons manipuler une fonctionnalité de windows seven permettant d'améliorer l'expérience utilisateur.

JumpList statique

Créer une tâche permettant d'ouvrir un notepad, ainsi qu'une autre tâche permettant d'ouvrir un fichier readme.txt situé à la racine de votre disque dur (c:\).

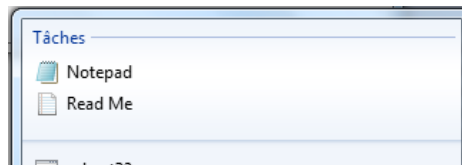


Figure 1 : Exemple de résultat

Manipulation de la JumpList

Comme vu dans le cours, inclure la possibilité de rajouter une tâche « calculatrice » au travers d'un bouton situé sur l'interface. Et un deuxième bouton ayant le pouvoir de supprimer l'ensemble de la jumpList.

Rajout de bouton

Pré-requis

Pour simuler un long traitement nous allons mettre en place un background worker.

Vous devrez initialiser le background worker de la manière suivante :

- Autoriser le rapport de progression
- Autoriser l'annulation
- S'abonner aux events suivants
 - DoWork
 - Implémenter une boucle for allant de 1 à 100, à chaque item faire un sleep de 25. Puis mettre à jour, la progression du worker au travers de la méthode ReportProgress.
 - ProgressChanged
 - Lors de cet événement vous devez calculer le pourcentage d'avancement pour mettre à jour la valeur de progression de la taskBarItemInfo
 - RunWorkerCompleted
 - Lorsque cet événement sera déclenché mettre l'état du worker dans le bon état (stop, pause, error, ...). Et mettre à jour l'image dans la barre des tâches

Ajout des boutons d'actions

Placer les 2 images pour le bouton stop et play dans les ressources de la page.

Implémenter les commandes stop et start en vous inspirant des exemples fournis dans le cours. La commande start devra lancer le worker et la commande stop lancer l'annulation asynchrone. Il ne vous reste plus qu'à ajouter la TaskBarItemInfo comme dans le cours.

Travail à faire

Vous devez créer un nouveau projet de type WPF/C#. Vous pouvez créer ce projet avec le framework 4.5 et vous aider de Blend pour réaliser les animations.

Le nom du projet devra être WPF_JumpList

Partie 3 : Input Binding

Renseignements

Cette partie va nous permettre d'implémenter un système d'input binding

La vue

Créer une vue composée d'une grid dans laquelle vous mettrez un stackpanel (avec un background transparent), et dans ce stackPanel y ajouter une liste représentant un panel de couleur :

- Rouge
- Vert
- Bleu
- Jaune
- Orange
- Violet

Les inputs

Implémenter deux inputs bindings : le premier sera le raccourci « ctrl+c » et le second le clic droit de la souris. Ces deux bindings implémenteront la même commande qui changera la couleur de fond de la forme selon la couleur présélectionnée dans la liste.

Travail à faire

Vous devez créer un nouveau projet de type WPF/C#. Vous pouvez créer ce projet avec le framework 4.5 et vous aider de Blend pour réaliser les animations.

Le nom du projet devra être WPF_Binding

Modalités de rendu

Les fichiers seront à rendre dans une tarball ayant pour nom :

login_.zip

Cette tarball devra comprendre à la racine:

- Un dossier contenant la solution Visual Studio qui devra compiler :

Nom : login_ITP4

Une fois décompressé, nous devrions avoir :

/Login_ITP4

*.sln

/WPF_DataGrid

/WPF_JumpList

/WPF_Binding

Le tout à envoyer sur l'adresse mti.rendu.dotnet@gmail.com avec les balises suivantes :

[MTI2014][WPF][login_I][TP4]