

SAÉ 2.01 - GameAtlas

Description de l'Architecture

par Rémi LAVERGNE & Yvan CALATAYUD

I. Modèles (Models) :

- La classe `Jeu` représente un jeu avec ses propriétés comme son nom en chaîne de caractères (string) ou la note en chiffre à virgule (double).
- Les jeux sont stockés dans une collection observable `'ListJeux'` dans le vue-modèle Manager.
- La classe hérite implicitement des méthodes `Equals` et `GetHashCode`. Ces méthodes héritées proviennent de la classe de base `Object`, qui est la classe parente de toutes les autres classes dans C#. Dans notre cas, il était préférable de comparer les objets `Jeu` en fonction de leur contenu (nom+date) plutôt que leur référence, cela à cause d'un bug. Donc en substituant ces méthodes pour définir une comparaison personnalisée pour `Jeu`.
- Les classes `Admin` et `Utilisateur` représentent respectivement un administrateur et un utilisateur avec leurs propriétés spécifiques.
- La classe `Admin` hérite de la classe `Utilisateur`, ce qui signifie qu'un administrateur est également considéré comme un utilisateur avec des fonctionnalités supplémentaires spécifiques à son rôle (Ici, ce n'est que le booléen `EstAdmin` qui est mit à True).

II. Interfaces :

- L'interface `IPersistenceManager` définit les méthodes nécessaires pour charger et sauvegarder les données des jeux, des administrateurs et des utilisateurs au niveau de la persistance.

III. Persistance :

- La classe `Manager` utilise l'interface `IPersistenceManager` pour la persistance des données. On implémente cette interface dans la classe `DataContractPers`, qui utilise la sérialisation (=conversion des objets pour sauvegarder en XML ou autre) avec `DataContractSerializer` pour stocker les jeux, les administrateurs et les utilisateurs dans un fichier XML.
- La classe `DataToPersist` est une classe de données (car elle ne comporte pas de méthode ou constructeur sauf celui par défaut) et est utilisée pour stocker les jeux, les admins et les utilisateurs dans un format compatible avec pour convertir en XML.

IV. Contrôleur Vue-Modèle :

- `Manager` est la classe qui fait gestionnaire entre les Vues et le Modèle (Code-Behind).

Elle possède les propriétés suivantes:

- `'Admins'` : Une liste des administrateurs dans l'application (classe `Admin`).
- `'Utilisateurs'` : Liste des utilisateurs (classe `Utilisateur`).
- `'TopRatedGames'` : Une collection observable des jeux les mieux notés. Cette propriété utilise la composition d'une liste de jeux (`'ListJeux'`) en utilisant un ordre décroissant basé sur la note des jeux, puis en gardant les 4 premiers jeux. > En utilisant LINQ
- `'ListJeuxAffiches'` : Une collection observable des jeux à afficher (en Page d'Accueil). Toujours en prenant de 'ListJeux' les 5 premiers jeux. > En utilisant LINQ
- `'ListJeux'` : Collection observable de tous les jeux enregistrés.
- `'Persistence'` : Une référence à l'interface `IPersistenceManager` qui est utilisée pour gérer la persistance des données.
- `'ConnectedUser'` : L'utilisateur connecté actuellement (null si aucun).

La classe `Manager` implémente l'interface `INotifyPropertyChanged`, ce qui permet de déclencher des événements lorsque les propriétés de la classe changent. Pour actualiser les vues ou d'autres parties de l'application lors d'événements et d'être notifiés des changements afin de mettre à jour

Elle dépend de l'interface `IPersistenceManager` pour la gestion de la persistance des données. Pour séparer la persistance du reste de la classe et faciliter l'ajout d'autres implémentations de persistance, comme la persistance via XML (actuelle), base de données, etc.

Toutes les parties Code-Behind des pages XAML possèdent un `Manager`, pour permettre le binding ainsi que l'échange d'informations.