TW3B:

Cette application est un bestiaire permettant de visualiser les différentes créatures de l'univers The Witcher.

Sommaire :

* Fonctionnement de l’application
* Explication de l’agencement de l’application (aide à la compréhension des diagrammes)
* Fonctionnalité personnelle
* Idées supplémentaires pour l’application

Fonctionnement :

Utilisateur :

- Cliquez sur le bouton "Monstres" pour revenir à l'accueil de l'application

- Pour voir un monstre :

- Sélectionnez la race de votre monstre dans le volet gauche de la fenêtre

- Sélectionnez ensuite le nom du monstre dans le volet central

- Le détail du monstre s'affichera alors dans le troisième volet

Administrateur :

- Cliquez sur le bouton "Se connecter", puis entrez dans la fenêtre "Login" votre identifiant et mot de passe, puis cliquez sur le bouton "Se Connecter"

- Pour créer un monstre

- Cliquez sur le bouton "Création"

- Entrez les données de votre monstre

- Cliquez sur le bouton "Créer"

- Pour voir les monstres créés

- Cliquez sur le bouton "Mes Monstres"

- Voir la marche à suivre plus haut

- Pour se déconnecter, cliquez sur le bouton "Se Déconnecter"

Agencement de l'application et aide à la compréhension des diagrammes :

- Bestiaire Witcher contient :

- Le code des fenêtres en XAML

- Leur code-behind en C#

- Bestiaire.Monstres contient :

- Les Enums des monstres

- Les Classes pour les objets :

-Items

- Monstres

- Bestiaire.Services contient :

- Les fichiers binaires permettant la persistance des données

- La Classe InitItems, permettant la création des Items

- La Classe InitMonstres, qui permet la création des monstres de base

- La Classe SaveLoad, qui permet la sauvegarde et le chargement d'objets dans un fichier binaire

- La Classe ServicesConnexion, qui permet d'autoriser ou non la connexion et de vérifier les login et mot de passes, et permet également d'ajouter à un fichier binaire et à chaque connexion le login et l'heure de connexion

- La Classe ServicesMonstres, qui permet de créer les différentes listes et dictionnaires de monstres

- Bestiaire.Test contient :

- Des tests utilisant du Linq (hors projet)

- Des tests de chargement, sauvegarde afin de réinitialiser les fichiers binaires s’ils sont supprimés ou corrompus

Fonctionnalité personnelle :

Ajout d'un système de stockage des données de connexion, avec le login et la date de connexion

La liste n'est pas visible depuis l'intérieur de l'application, ces données ne doivent pas être accessibles -> on passe par la méthode test SeeIdSpy

Idées en plus qu'il aurait été possible d'ajouter au projet :

- Un placeholder pour le login

- Appeler le Spy directement dans ServicesConnexion, et passer la classe en private

- Améliorer le XAML pour le rendre plus agréable

- Améliorer le système de visibilité des composants XAML

- Remplacer le système de sauvegarde en fichiers binaires par un système de bases de données

- Gérer le cas où les fichiers binaires ne sont pas présents ou corrompus (actuellement, passer par une fonction test qui les réinitialise)

- Ajouter des données dans IDSpy (si l'utilisateur a créé ou supprimé des monstres)