**Description des compétences**

**Conception et Programmation Orientées Objets**

**Documents :**

***-Je sais concevoir un diagramme de classes qui représente mon application :***

Le diagramme de classe est dans le dossier : Documentation

**-*Je sais réaliser un diagramme de paquetages qui illustre bien l’isolation entre les parties de mon application :***

Le diagramme de paquetage est dans le dossier : Documentation.

**- *Je sais décrire mes deux diagrammes en mettant en valeur et en justifiant les éléments essentiels :***

Les descriptions des diagrammes de classes sont dans le dossier : Documentation.

**Programmation :**

***-Je maîtrise les bases de la programmation C# (classes, structures, instances…) :***

Le langage C# est le seul langage de programmation utilisée dans ce projet.

***-Je sais utiliser l’abstraction à bon escient (héritage, interfaces, polymorphisme) :***

L’héritage est utilisé dans les cas suivants :

-Les classes Race et Competence héritent de la classe Element la variable car elles ont **3** attributs en commun.

-La classe Efle hérite de la classe Race car elle a tous les attributs de celle-ci en plus d’un attribut : NomElfique.

-La classe Race implémente l’interface IVivant pour devoir implémenter son attribut Stats. Une autre classe (Boss) était censée implémenter cette interface mais n’a pas été réalisée.

***-* *Je sais gérer des collections simples (tableaux, listes…) :***

Deux principales collections sont utilisées :

-Une ObservableCollection de Race.

- Une ObservableCollection de Race.

***-* *Je sais gérer des collections avancées (dictionnaires) :***

Un dictionnaire a été utilisé au début, puis enlevé car pas pertinent dans ce projet.

***-* *Je sais contrôler l’encapsulation au sein de mon application :***

Tous les attributs ont des getter et setter.

**- *Je sais tester mon application :***

Nous avons une application console dans notre projet qui a été utilisée pour tester certaines classes et la sérialisation.

***-* *Je sais utiliser LINQ :***

X

***- Je sais gérer les évènements :***

Des événements sont déclenchés lors du clic sur des boutons (changement d’onglet, ouverture de fenêtre…).

**IHM : Interface Homme-Machine**

## **Documents :**

***- Je sais décrire le contexte de mon application pour qu’il soit compréhensible par tout le monde :***

La description du contexte est présente dans le dossier : Documentation

***- Je sais dessiner des sketchs pour concevoir les fenêtres de mon application :***

Un fichier balsamiq contenant les sketchs du projet est dans le dossier Documentation. L’application finale ressemble à la conception de base.

***- Je sais enchaîner mes sketchs au sein d’un story-board :***

La dernière page du fichier balsamiq met en lien les différentes parties du projet.

***- Je sais concevoir un diagramme de cas d’utilisation qui représente les fonctionnalités de mon application :***

Le diagramme de cas d’utilisation est présent dans le dossier : Documentation.

***- Je sais concevoir une application ergonomique :***

L’application est intuitive et facile à prendre en main.

***- Je sais concevoir une application avec une prise en compte de l’accessibilité :***

Dans le MainWindow à la ligne 15, on a utilsé WindowState= ‘’maximised’’ afin que l’affichage de l’application s’adapte à la taille de l’écran.

**Programmation :**

***- Je sais choisir mes layouts à bon escient :***

On utilise Un Grid dans la fenêtre de création de race car c’est le plus adapté.

Un DockPanel est utilisé dans la MainWindow car l’ordre d’affichage est important dans ce cas (menu en haut et contenu en dessous)

On utilise un StackPanel pour afficher dans les onglets du detail des Races pour afficher de gauche à droite le nom de la race puis la croix qui permet de la supprimer.

***-* *Je sais choisir mes composants à bon escient :***

Plusieurs composants sont utilisés dans la partie WPF, pour changer le contenu de la MainWindow en cliquant sur un bouton.

***- Je sais créer mon propre composant :***

Dans la fenêtre de création de race, on peut ajouter une image en naviguant sur l’ordinateur et en choisissant les types de fichiers à afficher : JPEG,PNG ou JPG.

***- Je sais personnaliser mon application en utilisant des ressources et des styles :***

Il y a un dossier qui contient les images et icones utilisées dans l’application, dans SkyrimProjet.

***- Je sais utiliser les DataTemplate (locaux et globaux) :***

Nous avons un DataTemplate dans le TabControl du detail des races.

***- Je sais intercepter les évènements de la vue :***

Des événements sont déclenchés lors du clic sur des boutons (changement d’onglet, ouverture de fenêtre…).

***- Je sais notifier la vue depuis des évènements métier :***

On implémente l’interface INotifyPropertyChanged dans la classe ImageNotify.cs ce qui permet pour l’utilisateur de choisir une image pour la race qu’il crée.

***- Je sais gérer le DataBinding sur mon master :***

Le nom des races est affiché sur l’onglet des TabControls grâce au binding.

***- Je sais gérer le DataBinding sur mon détail :***

Lorsqu’on clique sur un onglet dans le TabControl dans UCRace, ça change les informations (compétences, statistiques…) dans le détail.

***- Je sais gérer le DataBinding et les Dependency Property sur mes UserControl :***

Nous arrivons à lier une propriété d’une classe à un contenu d’un label exemple dans UCDetailRace.

***- Je sais développer un Master/Detail :***

Le UserControl UCRace contient un TabControl qui constitué d’onglets(master) qui changent le contenu à droite(detail) lorsqu’on clique dessus.

# Projet Tuteuré S2

# Documents :

# *- Je sais mettre en avant dans mon diagramme de classes la persistance de mon application :*

# Dans le diagramme de classes, les classes avec des contours rouges sont les classes sérialisables, la classe Serialiser.cs a une dépendance sur le Modèle.

***- Je sais mettre en avant dans mon diagramme de classes ma partie personnelle :***

Dans le diagramme de classes, la classe avec un contour bleu est la classe personnelle (CreationRace)

***- Je sais mettre en avant dans mon diagramme de paquetages la persistance de mon application :***

Les paquetages entourés de rouge contiennent des classes

***- Je sais réaliser une vidéo de 1 à 3 minutes qui montre la démo de mon application :***

Le lien de la vidéo est dans le fichier PDF ProjetTuteures.

## **Programmation :**

***-Je sais coder la persistance au sein de mon application :***

Lorsqu’on ajoute une Race (ou on la supprime), et qu’on ferme l’application et qu’on la réouvre, le personnage reste (ou n’est plus la) si l’utilisateur a choisi de sauvegarder.

***-Je sais coder une fonctionnalité qui m’est personnelle :***

CreationRace permet d’ajouter une race à la collection avec contrôle de saisie, et choix de l’image.

***-Je sais documenter mon code :***

Toutes les classes du modèle sont commentées.

***-Je sais utiliser SVN. :***

Tortoise SVN est le seul outil utilisé pour partager les fichiers du projet, sans les .bin et .obj.

***-Je sais développer une application qui compile :***

L’application compile.

***-Je sais développer une application fonctionnelle :***

Toutes les fonctionnalités de l’application marchent.

***-Je sais mettre à disposition un outil pour déployer mon application :***