

Travail en équipe de deux. Le professeur doit approuver les équipes.

Date de remise:

- *Au plus tard mercredi le 30 mars 2016 avant 11h00 au début du cours.*
- *Aucun retard permis*

Objectif

L'objectif de ce travail est de construire une **application de gestion classique** tout en mettant en pratique les nouveaux concepts appris dernièrement soient les applications *Windows* multi-formulaires, la gestion des exceptions, la validation, les listes, le filtrage (formulaire de recherche), le tri, etc. L'application graphique à construire devra permettre de faire des **recherches avancées de cartes** du jeu *Hearthstone* et de **construire des decks** personnalisés (*Deck Builder*).

Fonctionnalités offertes par l'application

- L'application doit utiliser un **minimum de deux formulaires** et il doit y avoir du **passage d'information entre ces formulaires**.
- **Au démarrage**, les données sur les **héros** et les **cartes de *Hearthstone*** doivent être chargées dans des tableaux d'objets, et ce, à partir du fichier XML fourni. Attention : aucun deck ne doit être chargé à ce moment.
- Le **formulaire principal** devra permettre d'effectuer une **recherche avancée de cartes** en fonction de plusieurs critères (des spécifications précises suivront plus tard). Les cartes correspondant aux critères de recherche doivent être **affichées** soit sous forme **tabulaire** (en utilisant un *DataGridView*) ou bien à l'aide d'**images**; l'utilisateur doit pouvoir choisir entre ces deux modes d'affichage.
- Fonctionnalités liées à la gestion des decks :
 - **Créer un nouveau deck** en respectant les règles de conception décrites plus loin :
 - **Choisir un héros** pour le *deck*.
 - **Donner un nom** au *deck*.
 - **Ajouter des cartes** dans le *deck* tout en précisant la quantité.
 - **Enregistrer un deck** dans un fichier au **format XML**. Les *decks* doivent être enregistrés dans le dossier « *decks* » du répertoire courant. Le nom du fichier doit être créé automatiquement en fonction du nom du *deck* (ne pas utiliser une boîte de dialogue de *Windows* pour l'enregistrement). Si le *deck* existait déjà (avec le même nom), l'ancienne version est écrasée.
 - **Charger un deck** en utilisant une boîte de dialogue standard de *Windows* (*OpenFileDialog*).
 - **Modifier un deck :**
 - **Modifier** son nom.

- **Ajouter une nouvelle carte** dans le *deck*.
- **Supprimer toutes les copies d'une carte** dans le *deck*.
- **Modifier la quantité d'une carte** dans le *deck*.
- Note : Il n'est possible de changer le héros du *deck*.
- **Visualiser un *deck*** soit sous forme **tabulaire** ou bien à l'aide d'**images** (de manière similaire au résultat de la recherche de cartes).
- **Faire une recherche de cartes** en limitant automatiquement celle-ci aux cartes pouvant être incluses dans le *deck* et, à partir du résultat de la recherche, permettre **l'ajout** d'une carte dans le *deck* si c'est encore possible (nombre maximal de copies non atteint).
- Note : À tout moment, il ne doit y avoir qu'un **seul *deck* chargé en mémoire**.

Règles de conception d'un *deck*

- Un *deck* est constitué d'un **nom** (composé d'un minimum de 3 caractères sans considérer les espaces superflus) déterminé par l'utilisateur, d'un **héros** et d'une **liste d'au plus 30 cartes**.
- Les cartes pouvant être incluses dans un *deck* doivent être soit de la **même classe** que celle du **héros**, soit de la classe **neutre**.
- Il peut y avoir un maximum de **deux copies** d'une même carte dans un *deck* sauf pour les cartes **légendaires** où le nombre maximal de copies est limité à **un**. Vous devez tenir compte du fait que les règles liées au nombre maximal de copies peuvent éventuellement changer; par contre, vous pouvez supposer que pour toutes les raretés sauf légendaire, le nombre maximal permis sera toujours le même.

Fichier texte

Toutes les **données** nécessaires pour ce travail proviennent du fichier « **cards-collectible.xml** ». Ce fichier contient l'information sur les cartes (serviteurs, sorts et armes) ainsi que celle sur les héros.

Champs à considérer pour les **cartes** :

- Type (*type*).
- Identifiant (*id*).
- Nom (*name*).
- Extension (*set*).
- Rareté (*rarity*).
- Coût (*cost*).
- Texte (*text*).
- Classe (*playerClass*).
- Mécaniques (*mechanics*) : Il peut y en avoir plusieurs.
- Points d'attaque (*attack*).
- Points de vie (*health*).
- Race (*race*).
- Durabilité (*durability*).

Champs à considérer pour les héros :

- Identifiant (*id*).
- Nom (*name*).
- Extension (*set*).
- Rareté (*rarity*).
- Classe (*playerClass*).
- Points de vie (*health*).

Diagramme de classe à produire

- Vous devez produire un **diagramme de classes** avec *Visual Studio*.
- Sur le diagramme, on doit voir uniquement les **classes** avec leurs **attributs** et leurs **propriétés qui sont de type objet ou énumération** (pas de constructeurs, méthodes et constantes) ainsi que les différentes **énumérations** avec les **valeurs** de ces énumérations. Pour les énumérations ayant beaucoup de valeurs, vous pouvez vous limiter à 5 valeurs pour cette remise.
- Il ne doit **pas** y avoir de **chevauchement** sur le diagramme.
- Vous devez ajouter les **associations** (**simples** et de **collection**) entre les éléments du diagramme.

À remettre

- **Remise papier uniquement.**
- **Page titre** selon les normes du département d'informatique. Voir :
<http://deptinfo.cegepgarneau.ca/sectionDepartement/index.php?pg=coffreOutils>
- **Diagramme de classes imprimé** et produit avec Visual Studio. La qualité d'impression doit être très bonne; si votre diagramme de classes est difficilement lisible, vous n'obtiendrez pas les points liés à cette partie du travail.
- **Remettez cette partie du travail au début du cours avant 11h00 mercredi le 30 mars.**