|  |  |  |  |
| --- | --- | --- | --- |
| Développement d’application AVEC BASE DE DONNÉES  420-GED-HY  **Travail personnel #1**  **Partie serveur de l’application**  **(15 pts)** | | | |
| **Hugo St-Louis** | | **Bureau :** B 2344 | **Poste :** **2738** **hslouis@cegepsth.qc.ca** |
|  | | | |
| **Description** | | | |
| **But :** | Implémenter un jeu vidéo de type RPG avec l’Entité Framework. | | |
| **Objectifs** | 1. Utiliser d’un modèle relationnelle de base de données dans le but de créer une application. 2. Utiliser l’entité Framework pour gérer les éléments du CRUD d’un jeu. 3. Utiliser des procédures stockées avec l’entité Framework. | | |
| **Durée :** | 15 h | | |
| **Pondération :** | Travail évalué sur 100 points mais ramené sur 15 points. | | |
| **Remise :** | Le 21 septembre 2016 à 23h59. | | |
| **Note** | * Ce travail se fait en équipe maximum de trois(3). * Conserver une copie de sécurité. Il est de **votre responsabilité** de conserver une copie de sécurité dans l’éventualité où la lecture des données serait impossible. Cette copie doit être **disponible sur demande**. * Les règles de la PDEA et de la PIEA s’appliquent**.** * Le plagiat, sous toutes ses formes, est interdit. | | |

# Explication du travail pratique :

Le but de ce travail est de créer les éléments de base du jeu **Hugo Land.** Ce travail sera réutilisé tout au long de la session et permettra d’implémenter les éléments vues en classe.



Source : <http://lesdieuxvikings.xooit.fr/t20-dede57-grand-dieu-loup-viking-p.htm>

Dans un premier temps, nous allons implémenter la partie serveur qui permettra une interaction avec la base de données.

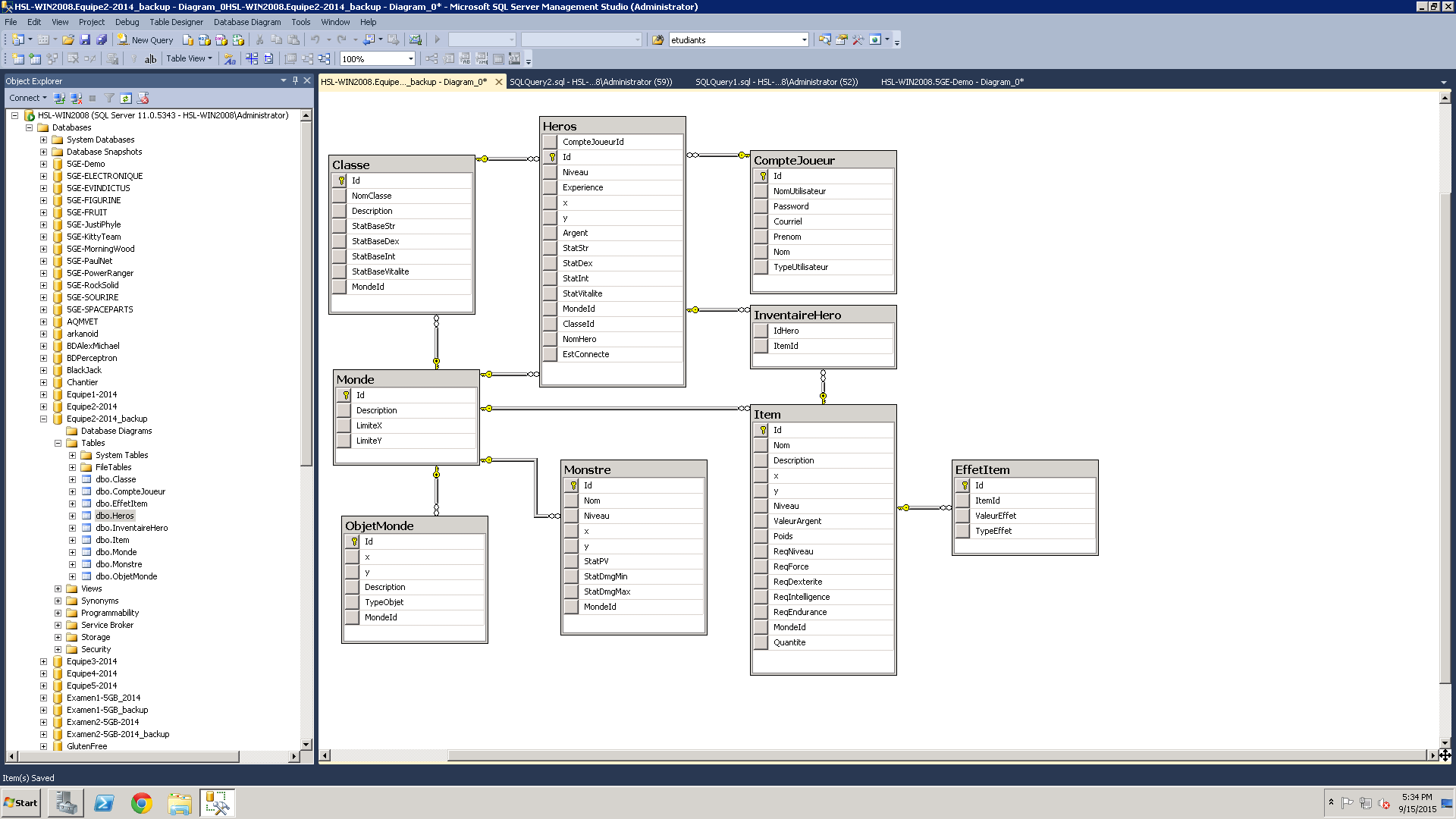
Ce jeu possèdera une partie serveur et une partie client. La partie serveur s’occupera de gérer :

* **L’interaction avec la base de données**
* **La persistance des données**
* **La gestion de mises à jour réalisées par les clients**

Vous devez créer une classe pour chaque objet. Dès le départ, vous devez créer un répertoire pour chacune des couches :

* La couche **Accès aux données** possèdera l’Entité Data Modèle.
* La couche M**étier** possèdera les méthodes de l’application
* La couche **Présentation** possèdera la partie Console qui permettra de visualiser les données. Dans ce travail pratique la couche de présentation servira exclusivement à tester votre code et ne servira plus pour les prochains travaux pratiques.

Pour vous économiser du temps, une base de données a été créée pour votre équipe. Voici le modèle relationnelle associé.



*Pour chacune des tables dans la base de données, voici l’objectif et les tâches qui y sont associés.*

## Monde :

**Objectif** : Cette table permet de stocker les informations relatives à un monde (carte) dans lequel évoluera le jeu. Cette carte spécifie les limites x, y du monde.

***Tâches :*** Vous devez implémenter les méthodes suivantes :

* Métier :
  + Créer un monde
  + Supprimer un monde
  + Modifier les limites et la description d’un monde
  + Retourner la liste des mondes
* Présentation
  + Pour chacune des méthodes ci-dessus, faites une méthode qui appelle une méthode et affiche les informations dans la console.

## ObjetMonde :

**Objectif** : Cette table permet de définir les éléments du décor dans le monde. Par exemple, cette table définira un rocher, de l’eau, etc. à une position x, y.

***Tâches :*** Vous devez implémenter les méthodes suivantes :

* Métier :
  + Créer un objet pour un monde donné.
  + Supprimer un objet pour un monde donné.
  + Modifier la description d’un objet pour un monde donné.
* Présentation
  + Pour chacune des méthodes ci-dessus, faites une méthode qui appelle une méthode et affiche les informations dans la console.

## Monstre :

**Objectif** : Cette table permet de définir les monstres dans le monde.

***Tâches :*** Vous devez implémenter les méthodes suivantes :

* Métier :
  + Créer un monstre avec des valeurs aléatoires.
  + Supprimer (tuer) un monstre.
  + Modifier les informations d’un monstre
* Présentation
  + Pour chacune des méthodes ci-dessus, faites une méthode qui appelle une méthode et affiche les informations dans la console.

## compte joueur :

**Objectif** : Cette table permet de créer les paramètres de connexion du joueur. Une fois connecté, le joueur se connectera et sélectionnera le héros avec lequel il désire jouer.

***Tâches :*** Vous devez implémenter les méthodes suivantes :

* Métier :
  + Créer un nouveau joueur
  + Supprimer un joueur
  + Modifier les paramètres d’un joueur
  + Valider la connexion d’un joueur
* Présentation
  + Pour chacune des méthodes ci-dessus, faites une méthode qui appelle une méthode et affiche les informations dans la console.

## item :

**Objectif** : Cette table permet les items que peuvent utiliser les héros. Lorsqu’un joueur met un item dans son inventaire, la position x, y de l’item est mise à zéro.

***Tâches :*** Vous devez implémenter les méthodes suivantes :

* Métier :
  + Créer un item
  + Supprimer un item
  + Modifier la quantité d’un item.
* Présentation
  + Pour chacune des méthodes ci-dessus, faites une méthode qui appelle une méthode et affiche les informations dans la console.

## effet item :

**Objectif** : Cette table permet de définir les bonus/malus d’un item.

***Tâches :*** Vous devez implémenter les méthodes suivantes :

* Métier :
  + Ajouter un effet à un item.
  + Supprimer un effet à un item.
  + Modifier un effet à un item.
* Présentation
  + Pour chacune des méthodes ci-dessus, faites une méthode qui appelle une méthode et affiche les informations dans la console.

## hero :

Cette table permet de définir le héros d’un joueur.

***Tâches :*** Vous devez implémenter les méthodes suivantes :

* Métier
  + Créer un héros.
  + Supprimer un héros.
  + Modifier les valeurs d’un héros
  + Pour un héros, retourner tous les éléments (ObjetMonde, Monstre, Item, Héro) qui se trouvent **dans un rayon de 200 par 200**.
  + Lister tous les héros pour un joueur.
  + Déplacer un héros.
* Présentation
  + Pour chacune des méthodes ci-dessus, faites une méthode qui appelle une méthode et affiche les informations dans la console.

## inventaire HÉROS:

**Objectif** : Cette table permet de définir l’inventaire d’un héros. Lorsqu’un héros possède ramasse/achète un item, la position x, y de cet item est assigné à zéro.

Vous devez implémenter les méthodes suivantes :

* Métier
  + Ajouter un item dans l’inventaire d’un héros.
  + Supprimer un item de l’inventaire du héros. N’oubliez pas d’assigner une nouvelle position à l’item.
* Présentation
  + Pour chacune des méthodes ci-dessus, faites une méthode qui appelle une méthode et affiche les informations dans la console.

## classe :

**Objectif** : Cette table permet de définir les classes associées aux héros du jeu. Chaque classe attribue des bonus/malus à un héros.

Vous devez implémenter les méthodes suivantes :

* Contrôleur
  + Créer une classe.
  + Supprimer une classe.
  + Modifier une classe.
  + Recevoir toutes les classes pour un monde donné (clé).
  + Trouver la classe d’un héros.
* Vue
  + Pour chacune des méthodes ci-dessus, faites une méthode qui appelle une méthode et affiche les informations dans la console.

# Les procédures stockés

# Pour l’entité héro, vous devez définir des procédures stockés pour les éléments du CRUD. Ces procédure stockés doivent être appelés lorsque la commande SaveChanges(); est utilisé sur l’Entité Data Modèle.

Ce travail devra être remis électroniquement au plus tard le 21 septembre à 23h59.

## Évaluation du travail de vos collègues :

Dans le but d’être juste avec tous vos collègues, vous devrez évaluer la performance de vos collègues dans ce travail. Vous devrez décider, par consensus, la distribution du résultat du travail sachant que vous avez 300 % à distribuer entre trois étudiants.

1. Vous allez donc assigner un pourcentage de participation au travail à chaque membre de l’équipe. Par exemple, si les trois membres de l’équipe (Alice, Bob, Claude) ont travaillé équitablement, alors Alice aura 100 %, Bob 100 % et Claude 100 %. Par contre, si les trois membres de l’équipe (Alice, Bob, Claude) n’ont pas travaillé équitablement, alors on pourrait avoir Alice 100 %, Bob 130 % et Claude 70 %. Ce pourcentage servira à calculer la note associée à chaque étudiant de telle sorte que la note sera multipliée par ce facteur (Note du travail \* facteur). Ce pourcentage devra être décidé par consensus en équipe.
2. Si vous n’arrivez pas à obtenir un consensus, j’utiliserai un ratio entre le nombre de lignes de code utile développé (pas de commentaire, de vide ou de changement de ligne) par une personne divisée par le nombre total de lignes de code utile.

Dans tous les cas, vous devez me remettre un document (avec une page de présentation) qui précise la distribution et explique votre choix.

## Devis à remettre :

Ce travail se fait en équipe maximum de trois(3) personnes. Dans le cadre du travail pratique #1, il vous est demandé de :

1. Le code source de votre application.
2. La pondération associée à chacun des étudiants (Document Word avec une page de présentation et l’explication de votre choix).

# Voici la pondération :

1. *Création correcte du modèle (1* ***points****)* ;
2. *Application correct des méthodes CRUD (7* ***points****)* ;
3. *Utilisation et création des procédures stockés lié à l’Entité Data Modèle (2* ***points****);*

Les critères d’évaluation sont :

- la complétude;

- la qualité fonctionnelle (les fonctions livrées donnent les bonnes réponses et sont correctes);