

TP 1 – Pokemon Tournament

Nintendo souhaite créer un jeux vidéo d'un tout nouveau genre après le succès de Pokemon Go ! Ayant eu connaissance de votre talent, ils souhaitent faire appel à votre classe.

Ce jeu proposera deux fonctionnalités : un pokedex permettant de lister tous les pokémons du joueur connecté avec leurs caractéristiques. Puis un jeu de combat entre Pokemon qui se présentera sous la forme d'un tournoi entre deux dresseurs. Chaque Pokemon aura ses caractéristiques propres (Type, force, intelligence, vitesse etc.) ces caractéristiques devront influencer l'issue du combat.



Fonctionnement du jeu

Le jeu devra contenir les fonctionnalités suivantes :

- 1) La possibilité de créer des pokemons, permettant de définir pour chacun d'entre eux leurs caractéristiques (type, force, agilité, endurance, vitesse etc.). A vous d'inventer les caractéristiques que vous souhaitez.
- 2) La possibilité de créer les stades ou auront lieu les combats. Chaque stade à également ses caractéristiques propres qui peuvent influencer sur tel ou tel Pokemon. (Ex : le stade "Bubulle" donne +2 en force aux pokemons de type eau).
- 3) Lancement des matchs : les 8 Pokemons seront dispersés aléatoirement pour la première phase du tournoi. Puis chaque match sera joué par le moteur du jeu qui définira le vainqueur du match de manière presque arbitraire : c'est à dire en fonction des différents caractéristiques des pokemons et des stades. Il faudra tout de même inclure une part d'aléatoire.

- 4) Etant donné qu'il s'agit d'un jeu en mode deux joueurs, vous aurez à créer un vrai système de combat pour les matchs. Chaque dresseur dirigera alors un Pokemon : cela peut être avec une liste de commande à la pierre/feuille/ciseau mais vous pouvez inventer le système que vous souhaitez.

Enfin, vous pouvez ajouter les fonctionnalités que vous désirez pour rendre votre jeu le plus attractif possible.

Objectif du TP d'aujourd'hui

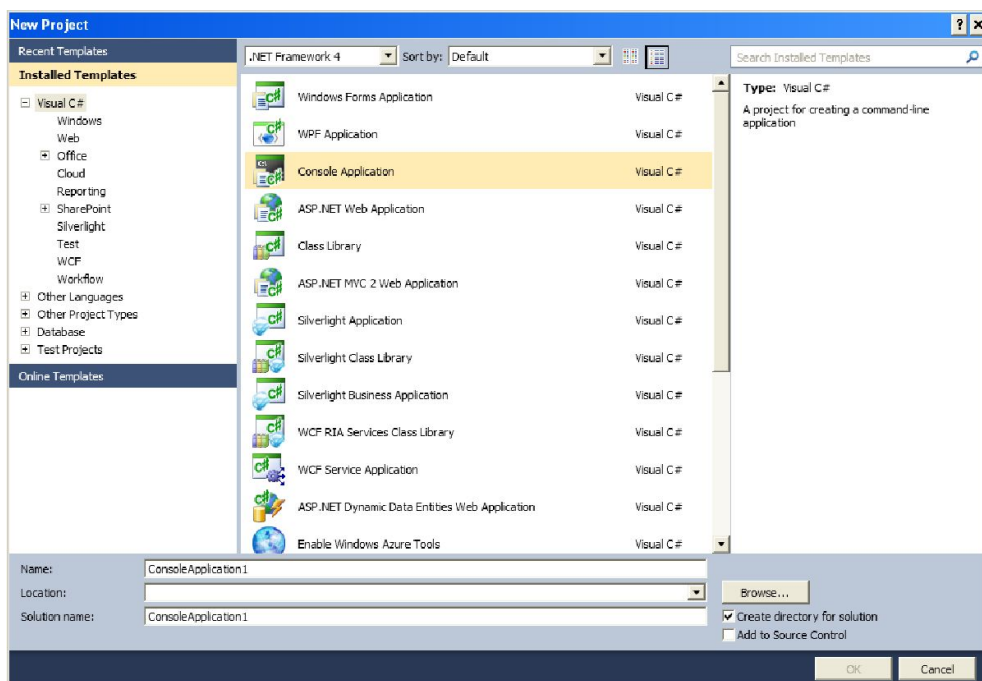
Créer la librairie de classes qui va être utilisé durant tous les Tps à suivre.

Manipuler des classes C#, des classes abstraites, des collections. Apprendre à séparer les objets dans les différentes couches.

Créer une application en mode console permettant la gestion « sommaire » du pokedex. Un utilisateur pourra ainsi lister ses pokemons et connaître le détail de chacun d'eux.

Etape 1 : Initialisation du projet

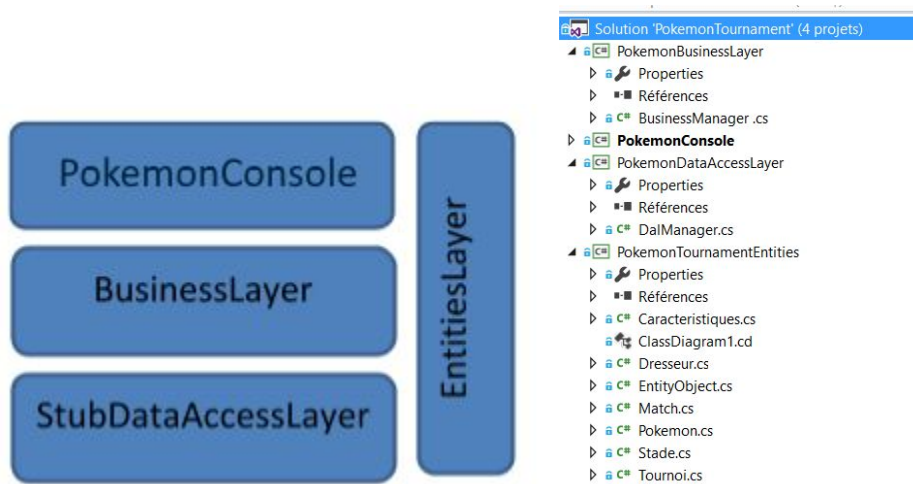
Ouvrir Visual Studio 2012. Fichier => nouveau projet, Choisir Visual C# => Application Console. La nommer « PokemonTournamentConsole ».



Dans la solution nouvellement créée, ajouter un nouveau projet de type bibliothèque de classes pour la couche business. Ouvrir Visual Studio 2012. Cliquer droit sur la solution => ajouter projet, Choisir Visual C# => Bibliothèque de classes. Nommer le projet « BusinessLayer ». Ajouter un 2^e projet Bibliothèque de classes. Nommer le 2^e projet Bibliothèque de classes. Nommer le « StubDataAccessLayer ». Ajouter un 3^e projet Bibliothèque de classes. Nommer le « EntitiesLayer ».

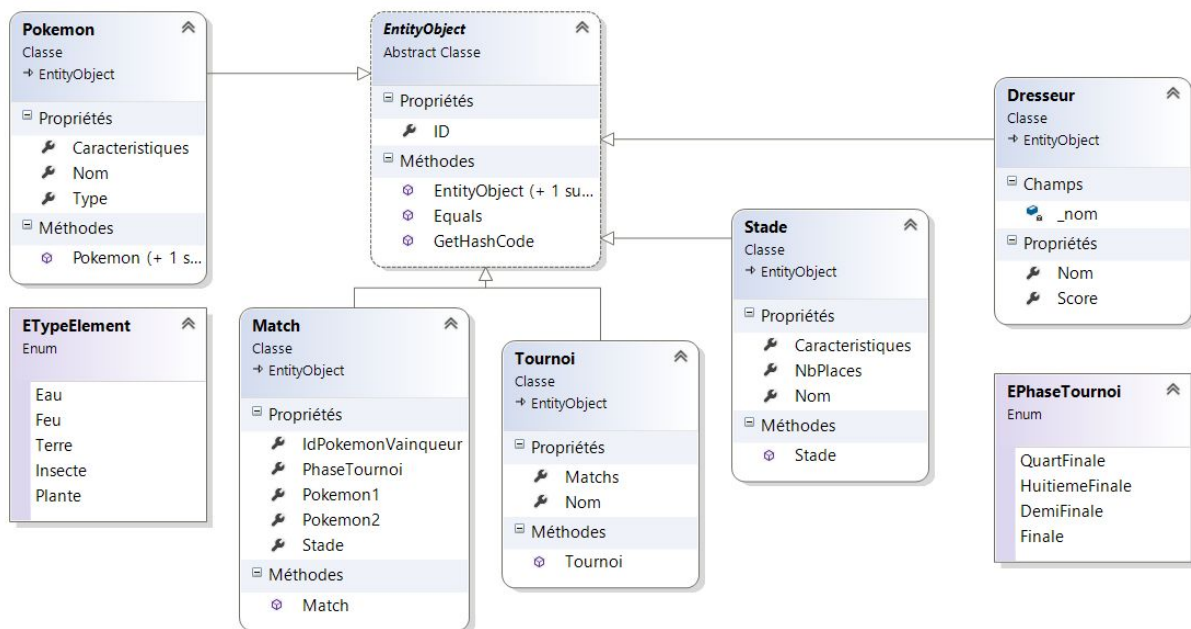
CO

Ajouter les références des différents projets de façon à ce qu'ils correspondent au schéma suivant :



Pour ajouter une référence à un projet, choisir le projet dans l'explorateur de solution. Puis cliquer droit => Ajout Reference et aller dans l'onglet projets.

Etape 2 : Couche Entities - EntitiesLayer



Pour commencer, il faut définir les objets de base. **Chaque objet devra contenir un numéro d'identifiant unique pour chaque enregistrement.** Afin de simplifier ce mécanisme on créera une classe abstraite « EntityObject » ayant pour unique propriété un ID de type « int ». Les autres

classes de la couche hériteront alors de cette dernière.

Créer les classes abstraites ainsi que les classes d'implémentation des éléments ci-dessus pour arriver au schéma présenté en début de chapitre, cependant vous pouvez faire preuve de créativité et d'originalité en ajoutant/modifiant celui-ci à votre convenance.

Etape 3 : Couche d'accès aux données - StubDataAccessLayer

Pour le moment, la couche d'accès aux données n'existe pas. Elle n'a pas encore été développée. On va donc « stuber » ou bouchonner la couche en question. Pour cela, il suffit de connaître les méthodes que la couche implémentera à terme et simuler l'ensemble de ces méthodes.

Créez une classe appelée DalManager. Cette classe doit contenir les méthodes suivantes :

- Retourner la liste de l'ensemble des pokemons existants
- Retourner la liste de l'ensemble des pokemons électriques existants
- Retourner la liste de l'ensemble des matchs existants
- Retourner la liste de l'ensemble des stades existants
- Retourner la liste de l'ensemble des caractéristiques existantes

« Stuber » la couche signifie retourner des données temporaires, inventées, afin de pouvoir travailler en parallèle sur plusieurs couches. Dans notre exemple, DalManager contient des listes de Pokemons, de matchs, de stades et de caractéristiques initialisées dans son constructeur.

Les méthodes se contentent de retourner les listes instanciées.

Etape 4 : Couche Métier - BusinessLayer

La partie métier de notre application est pour le moment très simple. Elle contient un manager qui permet à l'application de décentraliser et d'exécuter les demandes de la couche de présentation.

Créez une classe PokemonTournamentManager. Le BusinessManager demande les données au DalManager définit plus haut et les traite afin de mettre à disposition un ensemble de méthodes métiers destinées à notre application.

Le BusinessManager propose une méthode qui met à disposition de la couche Présentation l'ensemble des objets matchs existants.

Le BusinessManager propose également un ensemble de listings. Ces méthodes sont :

- Afficher la liste des stades
- Afficher la liste des Pokemons qui sont de type électriques
- Afficher la liste des Pokemons qui ont plus de 3 points de forces et plus de 50 points de vies.
- Afficher la liste des matchs qui ont eu lieu dans un stade de plus de 200 places.

Chacune de ses méthodes est destinée à de l'affichage de listing. **Elles revoient donc toutes des listes de string contenant le résultat attendu.** Utiliser Linq pour implémenter les méthodes

demandées.

Etape 5 : Couche de présentation -PokemonTournamentConsole

Il s'agit, pour commencer, de réaliser une application sommaire en Console.

Dans la partie présentation, proposer à l'utilisateur un menu à partir duquel il pourra accéder à l'ensemble des méthodes proposées par le BusinessManager.

Teaser (avec spoil)

Dans le prochain épisode de la saison 1 de "PokemonTournament" vous apprendrez à créer une véritable interface graphique en WPF :

