

Projet SGBD: Monstropoches

2^e année informatique

L'objectif du projet est de mettre en œuvre, sur un cas pratique, les notions et les méthodes vues dans le module SGBD. Le projet démarre par une modélisation des données, et aboutit à la création d'une base de données relationnelle et à l'implémentation d'un certain nombre d'opérations (consultations, mises à jour, etc.). Si des informations sont manquantes ou ambiguës, il est possible de fixer des hypothèses à condition de les détailler dans le rapport. Ce projet est à réaliser par *trinômes*.

Énoncé

Ce projet consiste à réaliser une base de données gérant un ensemble de personnages (les *Monstropoches*) accessibles dans un jeu vidéo. On dispose des informations suivantes :

- Le jeu se déroule dans un monde découpé en zones ; chaque zone a un nom (“La forêt brumisante”, “Les landes joyeuses”, *etc.*).
- Chaque monstropoche appartient à une certaine espèce, il a un nom que lui donne le joueur, des points de vie (PV), des points d'expérience (PE) (entre 0 et 100), un genre (mâle, femelle, aucun ou les deux), un propriétaire (qui peut être un joueur, un PNJ, ou aucun si le monstropoche est en liberté) et un état (en mission, en repos, en phase de reproduction, *etc.*).
- Une espèce de monstropoche a un nom (par exemple, un “kangoupou”), un numéro, un ou plusieurs types (aquatique, magique, *etc.*) ; elle a aussi des zones de prédilection (où les monstropoches en liberté gambadent, volent, nagent, rampent ou restent sagement immobiles).
- Les monstropoches peuvent s'affronter ; il existe un certain nombre d'attaques, chaque attaque a un type (similaire aux types des monstropoches), une certaine puissance (indiquant le nombre de PV qu'elle peut ôter) et une précision (les chances de réussite).
- Chaque attaque n'est accessible qu'aux monstropoches de certains types possédant un certain nombre de PE. De plus, un monstropoche ne peut maîtriser que 4 attaques simultanément.
- Un coefficient multiplicateur (décimal entre 0 et 2) peut être appliqué à la puissance d'une attaque selon son type et le type du monstropoche attaqué.
- Il existe un certain nombre d'objets que les monstropoches peuvent récupérer au cours du jeu ; certains de ces objets sont uniques, d'autres non, et chaque objet ne peut être trouvé que dans certains lieux. Certains de ces objets permettent de rendre certaines attaques plus efficaces, ou au contraire, s'ils sont portés par le monstropoche attaqué, de rendre certaines attaques moins efficaces.
- Un monstropoche ne peut pas porter plus d'un objet en même temps.

- Au cours du jeu, un monstropoche peut muer ; il change alors d'espèce, et donc potentiellement de type (il conserve néanmoins les attaques qu'il maîtrisait avant sa mue, jusqu'à ce qu'il les remplace par des nouvelles). Chaque espèce est soit une espèce basique soit accessible à partir d'une seule autre espèce par mutation (deux monstropoches d'espèces différentes ne peuvent pas muer dans la même espèce) ; par contre, il est possible qu'à partir d'une même espèce on puisse muer vers plusieurs espèces. Pour pouvoir muer en une nouvelle espèce, le monstropoche doit donc être de l'espèce de départ requise, mais aussi avoir un certain nombre de PE et, pour certaines mues, porter un objet spécial.

On veut stocker toutes ces informations dans la base, et pouvoir disposer des requêtes de service suivantes :

Consultation

Informations sur les monstropoches, les espèces, les attaques, *etc.* On aimerait en particulier connaître :

- la liste des attaques accessibles par chaque type, classée par ordre croissant de puissance ;
- la liste des objets, avec pour chaque objet, des booléens indiquant respectivement s'il est nécessaire à une mue, si c'est un objet offensif et si c'est un objet défensif ;
- la liste des monstropoches pouvant effectuer une mue, avec leur espèce actuelle et l'espèce dans laquelle ils peuvent muer, leur PE et éventuellement l'objet qu'ils portent qui rend la mue possible (noter qu'éventuellement, un monstropoche peut avoir le choix entre plusieurs mues) ;
- les couples de monstropoches pouvant engendrer un enfant ; pour engendrer, un couple doit être formé de deux individus de même espèce ou ayant la même espèce souche, l'un doit posséder le genre male et l'autre le genre femelle ; les deux individus doivent aussi être libres ou appartenir au même propriétaire et avoir des PE supérieurs à 20.

Statistiques

- le nombre de monstropoches de chaque espèce ;
- le nombre d'attaques disponibles pour chaque type, avec le nombre de PE moyen nécessaire pour accéder à ces attaques, ainsi que leur puissance moyenne ;
- pour chaque intervalle de PE ($[0;9]$, $[10;19]$,...), le nombre de monstropoches dans cet intervalle ;
- la liste des joueurs (ou PNJ), avec, pour chaque joueur, le types de monstropoches qu'il possède en majorité (si un joueur possède plusieurs types en majorité, il apparaîtra sur plusieurs lignes), et le nombre de monstropoches de ce type qu'il possède.

Mise à jour

Ajout, suppression, toute modification souhaitable pour le bon fonctionnement de la base.

Toute autre requête que vous jugeriez utile pour la gestion de cette base. Vous êtes encouragé à faire évoluer les consignes de ce sujet afin de laisser libre court à votre créativité et apporter des idées novatrices ; bien entendu, ces évolutions ne doivent pas amoindrir l'intérêt pédagogique des requêtes demandées.

Travail demandé

1. Concevoir et implémenter la base de données en SQL.
2. Un rapport détaillé sur la conception et l'utilisation de la base qui contiendra :
 - Modélisation des données :
 - Description du contexte de l'application (entités, associations, règles de gestion).
 - Modèle entité-association.
 - Liste des opérations prévues sur la base (consultation, mise à jour, etc.)
 - Schéma relationnel :
 - Passage au relationnel.
 - Contraintes d'intégrité, dépendances fonctionnelles.
 - Schéma relationnel en 3^e forme normale.
 - Implantation (base SQL au choix : Oracle, MySQL, PostgreSQL) :
 - Création de la base de données, en prenant en compte les contraintes d'intégrité (scripts de création, suppression, insertion).
 - Implémentation des commandes SQL réalisant les opérations retenues.
 - Utilisation :
 - Description de l'environnement d'exécution.
 - Notice d'utilisation.
 - Description des interfaces éventuelles (JDBC, PHP, etc.¹).

Rapport et sources

Vous devez soumettre un premier diagramme entité-association à votre chargé de TD au cours de la semaine du **5 novembre**. Ce diagramme (pdf ou image scannée en jpg) devra être déposée sur la page moodle du cours avant le 12 novembre dans un fichier tgz ou zip contenant aussi un fichier texte avec les noms des membres du trinôme.

Le rapport final devra contenir ce premier schéma et les étapes qui ont conduit à un schéma final éventuellement différent devront être argumentées.

Le rapport et les sources (en .tgz) sont à déposer sur la page moodle du cours avant le **11 décembre** minuit sous la forme d'un fichier zip dont le nom est formé des noms du trinôme ayant effectué le projet : **NOM1_NOM2_NOM3_NOM4.zip** (les rapports papiers sont à remettre au secrétariat avant le 12 décembre midi).

Remarques

- Les commandes SQL doivent être le plus possible paramétrées. Les commandes de création et de suppression de la base sont indispensables.
- Le choix des données de test doit être en adéquation avec le modèle. Les données doivent être pertinentes et peu nombreuses (éviter les saisies inutiles, avec aucune valeur ajoutée).
- Vous pouvez vous mettre à plusieurs groupes pour créer et partager un jeu de données de test.
- Le soin apporté à la lisibilité des requêtes ainsi que des résultats sera évalué.
- Ne pas attendre la dernière semaine pour commencer le projet...
- Des soutenances seront organisées.

1. On proscriera tout framework (ajax, etc.).

- Vous souhaitez écrire un rapport conforme aux normes de typographie française ? Prenez 3h de détente avant de commencer à l'écrire et lisez ce résumé : <http://jacques-andre.fr/faqtypo/lessons.pdf>
- Voici une documentation de qualité pour vous aider dans la conception du diagramme entité-association et sa conversion en modèle relationnel : <ftp://ftp-developpez.com/cyril-gruau/ConceptionBD.pdf>