
M2107 PROJET TUTORÉ
Duel sur la toile : *Application de gestion de jeux en ligne*

1 Introduction

L'objectif de ce projet est de développer une application pour la société *Duel sur la toile* qui souhaite proposer une plateforme en ligne où des utilisateurs pourront se défier sur différents jeux qui se jouent en 1 contre 1. N'étant pas au fait des nouvelles technologies cette application sera disponible via des programmes JAVA qui communiqueront via une base de données.

La plateforme devra permettre à des gestionnaires de mettre en ligne des jeux qui seront proposés aux utilisateurs. Les jeux de la plateforme sont uniquement des jeux qui se jouent en un contre un respectant une des modalités suivantes :

- soit les joueurs jouent différentes manches pour lesquelles ils doivent faire un maximum de points (par exemple le Boggle <https://fr.wikipedia.org/wiki/Boggle> où les joueurs doivent trouver le plus de mots possibles en un certain temps) ;
- soit les joueurs jouent différentes manches pour lesquelles ils doivent être le plus rapide possible ;
- soit les joueurs font une partie où chacun joue après l'autre (par exemple, le Scrabble <https://fr.wikipedia.org/wiki/Scrabble> ou le puissance 4 https://fr.wikipedia.org/wiki/Puissance_4).

Les utilisateurs enregistrés peuvent accéder à la liste des jeux de la plateforme. Ils peuvent notamment défier leurs amis ou alors demander un adversaire sur la plateforme. Cet adversaire sera choisi parmi les utilisateurs qui ont fait la même demande pour le même jeu. La plateforme devra évidemment notifier aux joueurs l'état de leurs parties en cours mais aussi leur fournir des statistiques sur les parties passées. Enfin la plateforme propose un système de messagerie entre utilisateurs leur permettant de s'envoyer des commentaires.

L'application que vous aurez à développer devra gérer tous les aspects liés à cette plateforme et comprendra :

- une interface d'administration de la plateforme permettant notamment de mettre des jeux sur la plateforme mais aussi de les mettre à jour ou de les désactiver. La partie administration devra aussi permettre de gérer les utilisateurs et de faire des statistiques sur les différents jeux proposés (nombre de parties effectuées sur une certaine période, meilleurs joueurs etc.)
- une interface joueur qui devra permettre aux utilisateurs d'accéder aux services de la plateforme (jeux et messagerie) et lui fournir toutes les informations le concernant (partie en cours, invitations, statistiques etc.)

Ces deux modules seront détaillés dans la suite mais tout d'abord quelques précisions sur les données manipulées par cette application.

2 Description de *Duel sur la toile*

La plateforme de jeux de la société *Duel sur la toile* possède deux types d'utilisateurs :

- **Les administrateurs** qui gèrent la mise en ligne des jeux et les utilisateurs de la plateforme. Ces utilisateurs peuvent aussi produire des rapports sur l'activité de la plateforme.
- **Les utilisateurs** qui vont pouvoir défier des adversaires sur la plateforme, gérer leurs parties et communiquer avec leurs adversaires. Ils pourront aussi consulter des statistiques sur leurs performances ou sur celles de leurs adversaires.

Un utilisateur de la plateforme possède un pseudonyme (deux utilisateurs ne peuvent avoir le même pseudonyme), un email et un mot de passe. L'utilisateur s'identifie sur la plateforme avec son login et son mot de passe. Les administrateurs sont des utilisateurs avec un rôle particulier. Un utilisateur peut être activé ou désactivé. Un utilisateur désactivé ne pourra pas se connecter à la plateforme.

Un jeu proposé sur la plateforme possède un nom, un descriptif de quelques lignes, le fichier `jar` contenant l'exécutable JAVA du jeu (le fichier sera directement stocké dans la base de données sous la forme d'un objet *BLOB*). Les jeux sont d'un des trois types décrits section 1. Un jeu peut être activé ou non. Dans ce dernier cas le jeu n'est pas disponible pour les joueurs.

Les joueurs peuvent se créer une liste d'amis. Ils peuvent envoyer des messages courts à des amis ou des adversaires. Ils peuvent défier une personne pour un jeu donné. Une telle invitation peut soit s'adresser à un joueur particulier, soit être lancée sans destinataire. Dans ce dernier cas c'est la plateforme qui se chargera de mettre en relation deux joueurs qui recherchent un adversaire pour le même jeu. On conserve la date et l'heure de l'invitation afin de pouvoir éliminer les invitations obsolètes (qui n'ont pas eu de réponse au bout d'un certain temps 24 heures par exemple).

Enfin les joueurs peuvent participer à une partie d'un jeu donné. Une partie concerne donc un jeu, deux adversaires et possède une date et heure de début, un numéro d'étape qui indique à quelle étape du jeu la partie se trouve. Cela peut être le numéro de la manche en cours pour les jeux qui se jouent en 3 manches, ou le numéro du coup pour les autres jeux. On prendra comme convention que -1 est une partie terminée et -2 une partie abandonnée. Enfin un champ texte permettra de conserver l'état de la partie sous la forme d'une chaîne de caractères JSON. Cet état est utilisé par le programme de jeu pour sa configuration. Pour chaque partie les deux joueurs ont un score.

3 Description de l'application

L'application est composée de deux modules à destination des deux types d'utilisateurs de la plateforme. Une troisième partie consistera aussi à implémenter deux jeux à destination de la plateforme. L'utilisation de ces deux modules et des jeux est très différente, ils devront donc avoir une interface Homme/Machine adaptée à leur utilisation.

3.1 Le module administration de la plateforme

Ce module doit servir à un administrateur pour effectuer les actions d'administration. Ce module devra lui permettre de gérer les jeux c-à-d ajouter de nouveaux jeux, modifier les jeux existants ou en désactiver (on évitera de supprimer un jeu pour ne pas supprimer les parties qui ont été effectuées pour ce jeu). Ce module permettra aussi de gérer les utilisateurs de la plateforme, notamment il pourra activer ou désactiver des joueurs. Lorsqu'un joueur est désactivé toutes ses invitations en cours sont supprimées et toutes ses parties en cours sont arrêtées en indiquant qu'il abandonne la partie. Enfin ce module devra permettre de consulter les statistiques de la plateforme. Par exemple afficher le nombre de parties en cours

ou pour chaque jeu le nombre de parties jouées sur une période donnée. À vous d'ajouter les statistiques qui vous semblent pertinentes.

3.2 Le module joueur

Ce module est à destination des utilisateurs de la plateforme. Lorsqu'un joueur lance l'application il a le choix entre se connecter ou créer un pseudonyme. Dans le deuxième cas il devra remplir un formulaire lui demandant les informations utiles. Une fois ces informations correctement saisies, l'utilisateur pourra se connecter avec le pseudonyme et le mot de passe choisis. Une fois le joueur connecté, il devra avoir accès aux informations suivantes :

- Le nombre de messages non lus
- Le nombre d'invitations qu'il a reçues ou qu'il a émises
- La liste de ses parties en cours ainsi que celles qui viennent de se terminer (par exemple les 10 dernières parties terminées)

Le joueur pourra effectuer les actions suivantes :

- Consulter ses messages et en envoyer
- Répondre aux invitations. Si il accepte une invitation une partie sera immédiatement créée.
- Consulter une partie en cours et éventuellement jouer une manche de cette partie. Lorsqu'un joueur consulte une partie il peut consulter les statistiques de son adversaire notamment sur les parties qu'ils ont jouées ensemble pour ce jeu.
- Consulter ses statistiques.

3.3 Les jeux

On vous demande d'implémenter deux jeux pour la plateforme dont un en langue anglaise :

1. Le jeu du **puissance 4** qui consiste à essayer d'aligner 4 pions soit en vertical, soit en horizontal soit en diagonale dans une grille de 7 colonnes sur 6 lignes. Dans ce jeu, chaque joueur a une couleur de jeton. Ils jouent l'un après l'autre en insérant un jeton dans une colonne. Ce jeton descend dans la colonne pour s'empiler sur les jetons déjà présents. La partie est terminée dès qu'un des deux joueurs a réussi à faire un alignement de 4 jetons ou que la grille est remplie. Le nombre de points est par exemple le nombre de jetons restant du joueur gagnant multiplié par le nombre de jetons alignés (4 le plus souvent). Pour ce jeu, l'état d'une partie en cours devra contenir la grille avec les jetons déjà joués ainsi que le nom du joueur courant.

Attention ! *L'IHM ainsi que les informations concernant ce jeu doivent être rédigées en Anglais.*

2. Le jeu du **Mastermind** qui consiste à trouver une combinaison de 4 couleurs choisies aléatoirement. Le principe est de faire des parties en 3 manches. Dans chaque manche, les deux adversaires doivent trouver la même combinaison. C'est celui qui a mis le moins de temps qui remporte la manche. Notez bien que les deux adversaires ne sont pas au même endroit et qu'ils ne jouent pas obligatoirement leur manche au même moment. Pour ce jeu, l'état d'une partie consistera simplement en la combinaison à trouver et le temps mis par les joueurs pour chacune des manches.

Afin de permettre au module *joueur* de lancer un jeu, il faudra que les jeux implémentent l'interface Jeu suivante :

```
package projet;
import javafx.scene.layout.Pane;
public interface Jeu{
    /** permet à un joueur de lancer le jeu en retrouvant
        l'état actuel de la partie
        - idPartie est l'identifiant de la partie dans
            la base de données
        - numJoueur est le numéro du joueur de la partie (1 ou 2)
        - partage permet de donner au jeu l'accès à
            des variables du module joueur
        le retour de la méthode donne le composant graphique
        dans lequel s'affiche le jeu
    **/
    public Pane jouerCoup(int idPartie, int numJoueur, Object partage);

    /** permet de demander à créer une nouvelle partie et
        de la lancer
        - idJeu est l'identifiant du jeu dans la base de données
        - idJoueur1 et idJoueur2 sont les identifiants des
            deux participants
        - partage permet de donner au jeu l'accès à des variables
            du module joueur
        le retour de la méthode donne le composant graphique
        dans lequel s'affiche le jeu
    **/
    public Pane creerPartie(int idJeu, int idJoueur1, int idJoueur2,
        Object partage);
}
```

Cette interface permet de créer une partie ou de continuer une partie en cours. Les deux méthodes retournent le composant graphique dans lequel s'affiche le jeu. Le module joueur pourra ainsi intégrer la vue du jeu. Le paramètre partage peut vous permettre de définir une classe permettant au jeu d'accéder à des objets du module joueur. Notez bien que les jeux doivent être capable de créer une nouvelle partie dans la base de données, de récupérer l'état d'une partie en cours et de mettre à jour cet état après qu'un joueur ait joué.

4 Travail à faire

Vous aurez à analyser et développer cette application.

4.1 Organisation

Ce projet est à faire en groupe. Les groupes ont été constitués et vous sont imposés. Aucun changement n'est possible. Vous devez désigner un chef de projet par groupe avant la fin de la semaine. Chaque groupe aura un enseignant référent.

Au sein du groupe vous devrez partager vos développements via le dépôt GIT qui aura été créé à cet effet à l'URL `gitlab.com:iut-orleans/groupeXYZ.git` où XYZ est votre numéro de groupe.

4.2 Évaluation

Attention, l'évaluation du projet ne se fera pas uniquement sur les programmes que vous aurez rendus mais également sur la qualité de l'organisation de votre projet. La bonne répartition des tâches et la contribution de chaque membre du projet seront évaluées. L'activité de votre projet GIT sera notamment un bon indicateur de la manière dont vous aurez travaillé.

Le rendu du projet se fera en quatre phases :

4.3 Phase 1 : Rendu le 30 mars

Ce rendu se fera sous la forme d'un rapport comprenant les éléments suivants :

1. Le dictionnaire des données
2. Le Modèle Conceptuel des Données (MCD)
3. Un script de création de la base de données correspondant à votre MCD comprenant les contraintes d'intégrité.
4. Les requêtes qui permettent de répondre aux questions suivantes
 - La liste des parties en cours de l'utilisateur *iuto*
 - La liste des messages non lus de l'utilisateur *iuto*.
 - Le nombre de parties gagnées par le joueur *iuto* contre le joueur *iutc* pour le jeu *Puissance 4*

Ces questions doivent vous aider à vérifier que votre MCD permet de retrouver les informations nécessaires à certains traitements de l'application.

5. Le diagramme des cas d'utilisation de l'application. Ce diagramme doit rester *grossier*.

4.4 Phase 2 : Rendu le 27 Avril

Cette deuxième phase va consister à faire l'analyse des modules de l'application en vue de leur développement. A l'issue de la première phase, il vous sera fourni la base de données avec laquelle vous devrez travailler pour la suite du projet.

Le rendu de cette deuxième phase sera un dossier comprenant :

1. pour les modules *Joueur* et *Administrateur*.
 - La description textuelle du cas d'utilisation
 - Le diagramme de cas d'utilisation détaillé
 - Le diagramme de séquence du scénario nominal
 - Le diagramme d'activité avec l'ensemble des scénarios
2. pour les deux jeux *Puissance 4* et *Mastermind*
 - La description textuelle du cas d'utilisation
 - Le diagramme de cas d'utilisation détaillé
 - Le diagramme de séquence du scénario nominal
 - Le diagramme d'activité avec l'ensemble des scénarios

Rappel ! L'IHM du *Puissance 4* doit être réalisée en Anglais.

3. un planning prévisionnel de votre projet sur la forme d'un diagramme de GANTT ainsi qu'une répartition des tâches entre les différents membres du projet (matrice RACI).
4. une fiche individuelle par membre du groupe où chaque membre du projet indique ce qu'il a réalisé au cours de cette phase et ce qu'il devra réaliser dans les phases suivantes.

4.5 Phase 3 : Rendu le 8 juin

Ce rendu sera dédié au maquetage. Il comprendra :

- L'analyse UML de votre projet
- Le modèle des tâches de votre projet
- Les IHM adaptées pour chacune des tâches en justifiant vos choix ergonomiques et fonctionnels. Ces IHM devront être fournies sous forme de maquettes avec des exemples d'enchaînements d'écrans. Dans cette analyse, ne vous censurez pas même si vous n'êtes pas sûr de pouvoir réaliser techniquement votre proposition.
- Un état d'avancement de votre projet par rapport à votre planning prévisionnel et un diagramme de GANTT éventuellement rectifié pour la fin du projet.
- Une fiche individuelle par membre du groupe où chacun indiquera ce qu'il fera, un bilan de ce qu'il a réalisé par rapport au prévisionnel en justifiant les différences et indiquera ce qu'il aura à réaliser pour la phase finale.

4.6 Phase 4 : Rendu final le 22 juin

Le détail du rendu final sera précisé par la suite mais il devra comprendre :

1. La finalisation des dossiers d'analyse précédemment demandés.
2. L'état d'avancement du projet. Ce qui a été réalisé, ce qui ne l'a pas été en expliquant pourquoi.
3. Un manuel utilisateur.
4. Une archive contenant tout ce qu'il faut pour exécuter votre projet (code sources, script de création de la base de données et jeu d'essai, fichiers annexes, etc.).
5. Une fiche individuelle où chaque membre du projet explique les tâches qu'il a réalisées au cours de cette phase et indique en les justifiant les différences entre le prévisionnel et le réalisé.

Chaque groupe présentera son travail lors d'une soutenance qui aura lieu le **22 juin**.