

DEVOIR

Rapport Tp2 : JavaEE

Auteur :

Mohammed El Amine

SOUADJI

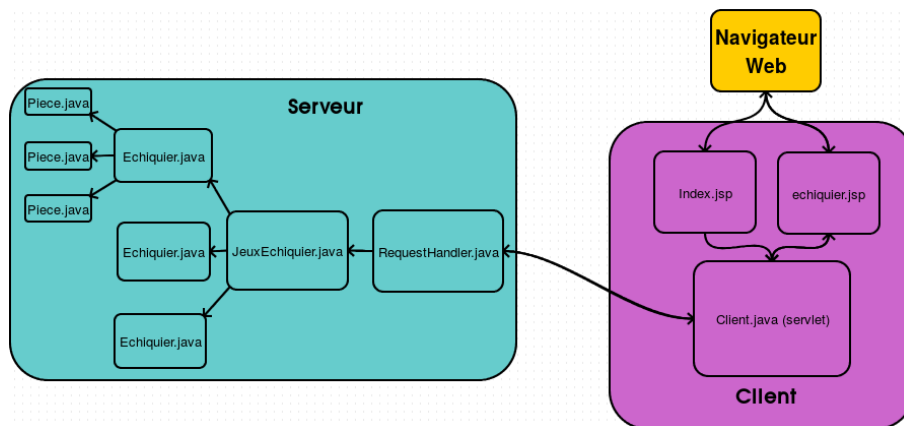


FIGURE 1 – Architecture du web service

1 Description des classes

1.1 Serveur

1.1.1 Piece.java

l'interface pièce.java est implémentée par toutes les pièces de l'échiquier, elle propose trois méthodes :

isPiece() : pour différencier les pièces et la case vide.

getJoueur() : pour récupérer la couleur de la pièce si elle appartient au joueur 0 elle sera blanche sinon elle sera noire.

getAvailableMove() : cette méthode retourne tous les déplacements possibles pour une pièce dans un échiquier vide, comme cela chaque pièce suit sa trajectoire par exemple un cavalier ne pourra pas se déplacer vers l'avant (par manque de temps, je n'ai pas finalisé cette fonction, car pour l'instant une pièce se déplace suivant sa trajectoire, mais elle peut sauter au-dessus d'autres pièces, ce qui n'est pas permis dans le jeu).

1.1.2 Echiquier.java

Cette classe représente un échiquier, elle offre plusieurs méthodes de gestion d'échiquier tel que le déplacement d'une pièce la récupération d'une case, la gestion des deux joueurs, et la gestion d'historique. chaque échiquier du jeu est représenté par cette instance.

1.1.3 JeuxEchec.java

Cette classe représente en quelque sorte le noyau de toute notre application (le modèle dans une architecture MVC), elle gère plusieurs parties et plusieurs joueurs à la

fois, elle permet la création de l'échiquier ainsi que son chargement et sa suppression, comme elle permet au joueur de joindre des parties déjà existantes. Dans cette classe, on trouve une hashmap qui contient tous les échiquiers, et chacun d'eux est représenté par un id qui sera fourni au joueur afin d'accéder à leur échiquier.

1.1.4 RequestHandler.java

cette classe représente le serveur de notre service web, elle offre toutes les fonctionnalités de la classe JeuxEchec.java, en plus de cela, elle permet l'authentification, l'enregistrement des nouveaux joueurs et les interactions avec la base de donnée derby.

1.2 Client

Le client se divise en deux parties le servlet qui joue le rôle de contrôleur et la page JSP qui joue le rôle de la vue. Toutes les opérations sont traitées par le servlet mis à part l'authentification et l'inscription des joueurs qui sont traitées par la Jsp.

1.2.1 Client.java

cette classe représente le client de notre web service, elle implémente tous les appels des méthodes offertes par le serveur. Toute opération sur l'échiquier passe par cette classe là, elle traite la requête et re-affiche la page echiquier.jsp si c'est un appel simple et dans le cas où c'est un appel ajax elle affiche une sortie.

1.2.2 Echiquier.jsp

cette page représente la vue de notre application, elle permet au joueur d'exécuter différentes commandes comme par exemple bouger une pièce ou sauvegarder la partie. Toutes les données sont envoyées au servlet client.java par la méthode GET du protocole HTTP, on ajoute, un paramètre op pour différencier les opérations.

1.2.3 index.jsp

La page index.jsp, elle représente la page d'accueil, elle gère l'authentification et l'inscription de nouveau joueur. Une fois le joueur authentifié elle lui permet soit de commencer une nouvelle partie ou de rejoindre une partie déjà existante.

2 Utilisation

2.1 Serveur

Pour faire marcher le serveur il faut d'abord modifier le chemin de la base de données dans la classe "m1.tp.javaee.webservice.server.RequestHandler" (ligne 41), car je n'ai pas pu utiliser un chemin relatif. Notre serveur se nomme RequestHandler donc le chemin vers le serveur est : `http://localhost:8080/wsEchiquierServer/RequestHandler`.

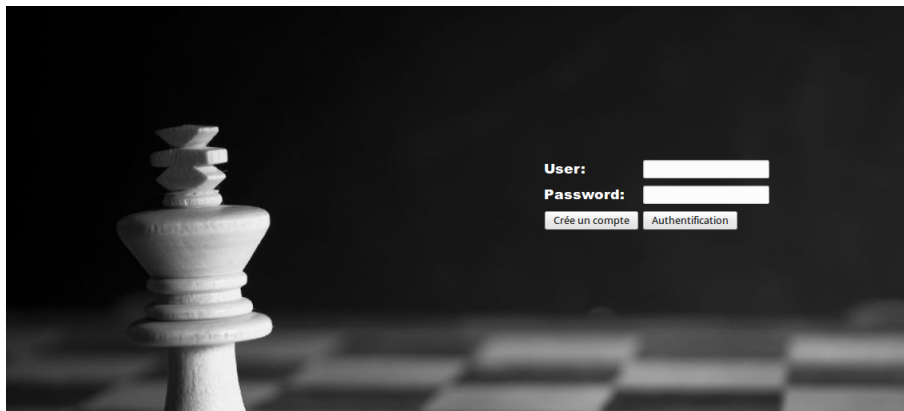


FIGURE 2 – Page d’authentification

2.2 Client

Le chemin vers le client : `http://localhost:8080/wsEchiquierClient/index.jsp`

Authentification Avant de commencer à jouer le joueur doit soit s’authentifier soit s’enregistrer Figure 2, cette partie est très basique pour s’authentifier il que l’utilisateur introduit un nom d’utilisateur et mot de passe valide. S’il n’a pas de compte, il suffit de mettre un nom d’utilisateur et mot de passe pour créer un compte.

Page d’accueil Cette partie offre à l’utilisateur deux choix soient créés une nouvelle partie ou rejoins une partie existante Figure 3. S’il crée une nouvelle partie, il pourra choisir entre crée un nouvel échiquier ou charger une ancienne partie, le joueur qui crée une partie a la couleur blanche. Et s’il choisit de rejoindre une partie existante, il aura la liste des parties en attente de joueur Figure 4, et dans ce cas-là, il aura la couleur noire.

Echiquier Cette page permet au joueur de jouer aux jeux d’échecs Figure 5. Pour pouvoir jouer, **on doit avoir un joueur blanc et un joueur noir chacun dans un navigateur**. pendant que le joueur attend son tour, la page jsp envoie des requêtes Ajax périodique pour voir, si c’est son adversaire a joué ou pas encore, si oui, elle recharge l’échiquier et donne la main au joueur.



FIGURE 3 – Page d'accueil

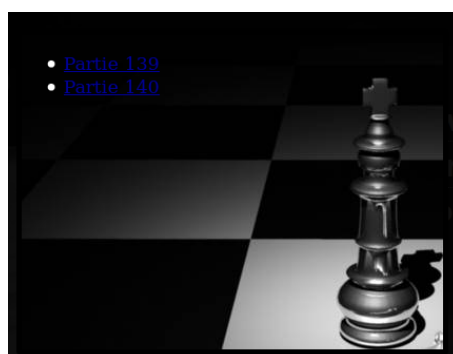


FIGURE 4 – Rejoindre une partie

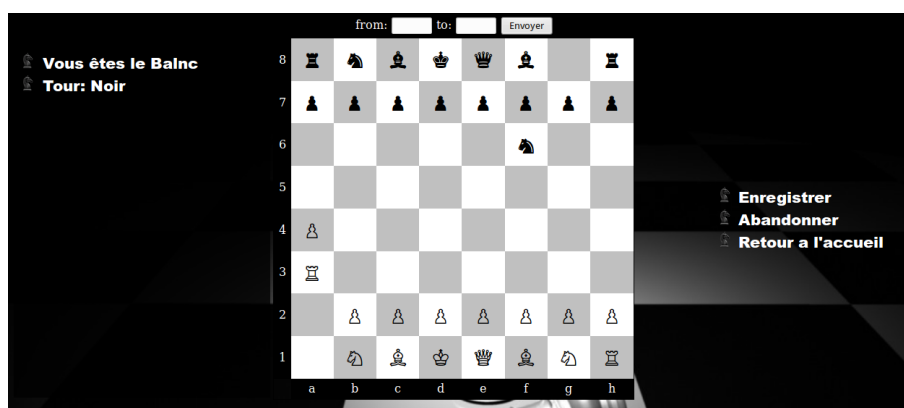


FIGURE 5 – Echiquier