

Développement d'un jeu 4X multijoueur en JEE

BEZET Camille, PAVOT Fiona, BAIVEL Alexandra, MITSAKIS David ING2
GSIA2

15 Janvier 2024

Base de données

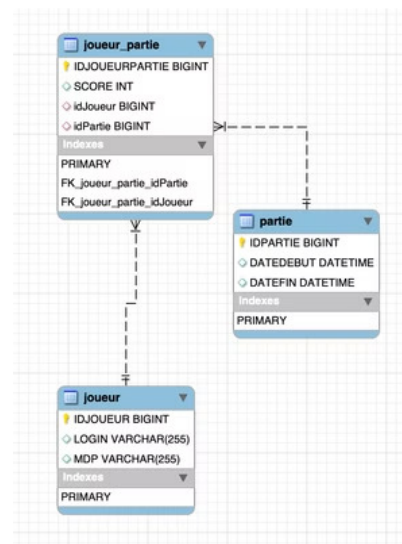
JPA et MySQL

L'architecture de la base de données repose sur JPA et MySQL.

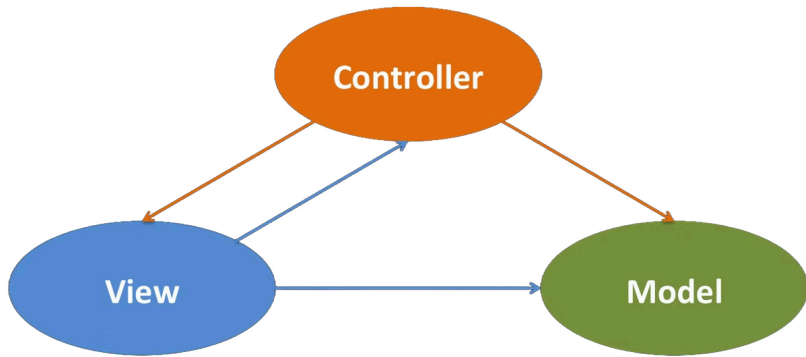
Les entités principales du projet sont *Joueur*, *Partie* et *JoueurPartie*.

Configuration

Le fichier *persistence.xml* configure la connexion à la base de données, incluant le driver JDBC, l'URL de connexion, les identifiants d'accès et la génération automatique des schémas.



Structure MVC



Modèle

Classes définissant les objets du jeu, tels que les soldats, les joueurs et le plateau de jeu.



Service

Assure la logique du jeu en appelant les classes du modèle et expose des fonctionnalités aux servlets.



Controller

Reçoit des requêtes envoyées par les pages JSP et appelle les fonctions des services concernés.



JSP

Assure l'interaction avec l'utilisateur et redirige vers les servlets correspondantes aux actions des joueurs.

Démarches et outils utilisés

1

Gestion de projet

Utilisation de Github pour la gestion de code et Discord pour la communication.

2

Inscription et connexion

Page login.jsp pour la création de compte et la connexion.

3

Multijoueurs

Le plateau de jeu est rafraîchi à chaque appel serveur pour le joueur en train de jouer et par polling toutes les 5 secondes pour les autres joueurs.

Bienvenue sur ce jeu

Inscription/Connexion

Login

Mot de passe

Se connecter

Le login est déjà utilisé ou le mot de passe est incorrect

Répartition des tâches au sein du groupe

Personne	Travail réalisé pour le projet
Fiona	<ul style="list-style-type: none">• Fonctionnalité “se soigner”• Fonctionnalité “passer au tour suivant”• Écriture et relecture du rapport final• Review du code
Camille	<ul style="list-style-type: none">• Page “Récapitulatif des scores” à la fin d’une partie• Page “Récapitulatif des scores” des anciennes parties d’un joueur• Review du code• Écriture et relecture du rapport final• Écriture des slides
David	<ul style="list-style-type: none">• Fonctionnalité “fourrage des ressources”• Fonctionnalité “recruter un soldat”• Gestion du tour par tour• Review du code• Écriture et relecture du rapport final
Alexandra	<ul style="list-style-type: none">• Connexion à la base de données• Architecture du projet• Page de connexion• Apparence de la grille• Fonctionnalité “Mouvement”• Fonctionnalité “Finir la partie”• Combat• Review du code• Relecture du rapport final

Difficultés rencontrées

1

Gestion de la base de données

Maîtrise de JPA et la bibliothèque Persistence.

2

Communication et gestion du code

Gérer les erreurs de code, gérer les erreurs lors des merges, communiquer sur ce qu'on fait.

Conclusion

1

Expérience formatrice

Apprentissage des technologies JEE
et de l'architecture MVC.

2

Fonctionnalités complexes

Gestion des scores, synchronisation
des joueurs et interactions
dynamiques avec la base de données.



Fonctionnalités développées

Soldats

Sélection, mouvements, soigner, fourrage des forêts.

Recrutement

Recruter un soldat en utilisant des points de production.

Finir le tour

Passer le tour au prochain joueur.

Combat

Système de combat entre soldats et villes.

Score

Le score est calculé selon la formule : **5*Nombre de parties gagnées - 6*Nombre de combats perdus + 10*Nombre de villes acquises**

Carte aléatoire

À chaque nouvelle partie une carte générée aléatoirement avec les éléments suivant : soldats, villes et forêts.

Affichage des scores

Après avoir cliqué sur "Finir la partie", les joueurs ont accès à un tableau récapitulatif des scores de chaque joueur.

Affiches des scores du joueur

Depuis la page d'accueil, le joueur peut voir les scores de ses anciennes parties.

