# Java / WebDev CIR3

## Projet : premier semestre - Mine Sweeper

### Objectif :

L’objectif du projet est de valider les connaissances acquises en cours de WebDev concernant l’utilisation du Framework Java « Spring », et principalement les points suivants :

* Les bases d’une application Web (requêtes, verbes, html, …)
* Organisation des sources dans le framework (design pattern : DI)
* Utilisation du moteur de template « Thymeleaf » (design pattern : MVC/IoC)
* Persistance des informations en base de données (design pattern : ORM)

Il s’agit d’un projet individuel, qui devra être rendu au plus tard dimanche 12 Janvier.

Mode de rendu : code source sur un site d’upload, + lien par mail

(exemple : <https://send.firefox.com/>)

Les critères d’évaluations concerneront principalement :

* L’organisation du code (packages, …) qui doit respecter le design pattern MVC
* La conception du modèle ainsi que sa persistance (relations, …)
* L’application des bonnes pratiques vues en cours
* La simplicité et la lisibilité des algorithmes développés

Le projet doit obligatoirement être développée en Spring Boot (Spring Data, Thymeleaf, …) + Bootstrap.

L’utilisation de Javascript est **fortement déconseillée** pour ce projet.

En cas de question, vous pouvez me joindre à l’adresse suivante :

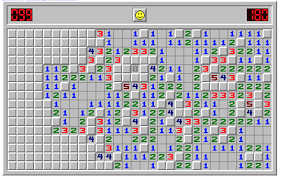
[sebastien.minet@soprasteria.com](mailto:sebastien.minet@soprasteria.com)

(joindre une copie de votre code, en **excluant** le répertoire target qui est volumineux)

### Présentation :

L’application à implémenter une version web du jeu « Minesweeper » dont la première version date des années 60.

<https://fr.wikipedia.org/wiki/D%C3%A9mineur_(genre_de_jeu_vid%C3%A9o)>



L’application devra permettre les grandes fonctionnalités suivantes :

* Créer des modes de jeu (configuration des paramètres : taille du champ, nombre de mines)
* Jouer une partie après avoir sélectionné un mode
* Garder une liste du top 5 des scores, par mode de jeu

Toutes les règles du jeu doivent être codé en Java (coté serveur).

### Détail des fonctionnalités :

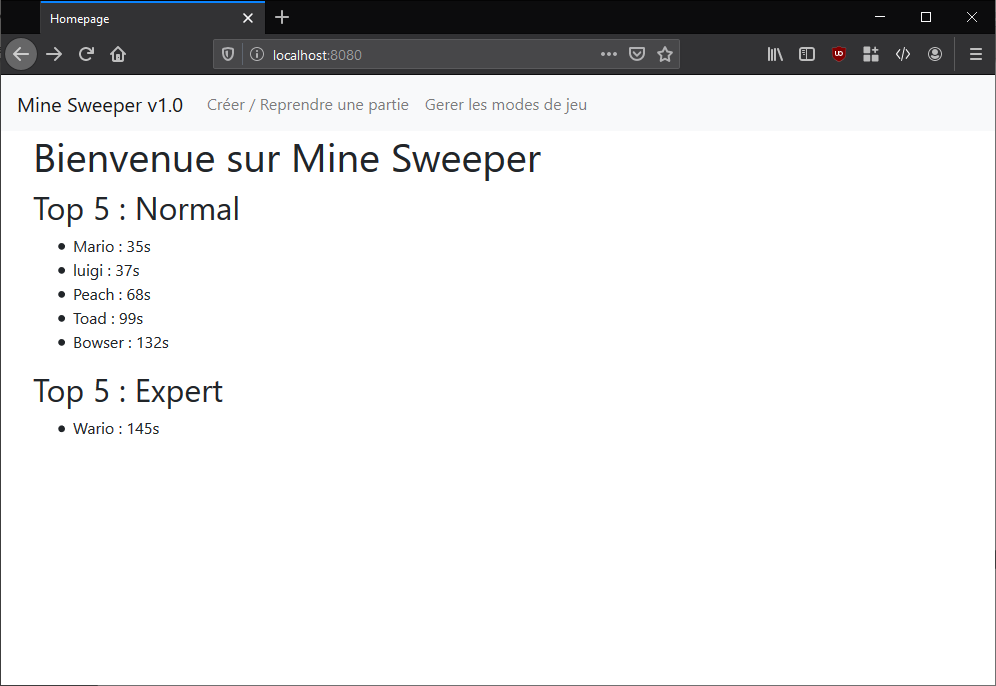
#### Page d’accueil :

La page d’accueil sert de point central d’accès aux fonctionnalités de l’application.

Elle affiche :

* Un lien permettant de consulter la liste des modes de jeu
* Un lien permettant de reprendre une partie non finie, ou d’en créer une nouvelle (si la dernière partie a été achevée ou abandonnée)
* Le top 5 par mode de jeu

Exemple :



#### Ajout / suppression / modification des modes de jeu

Il s’agit d’un écran affichant la liste des modes de jeu. Comme dans un écran CRUD habituel, il est possible d’ajouter, modifier ou supprimer un mode de jeu.

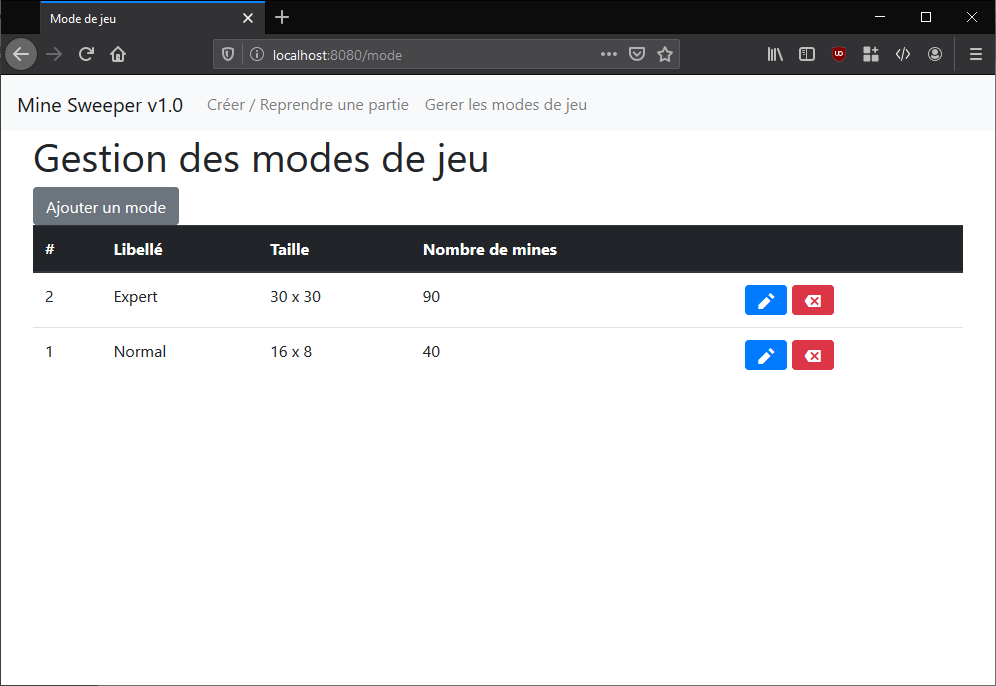
Dans le cas d’une modification ou d’une suppression, le top 5 des scores du mode en question sera réinitialisé.

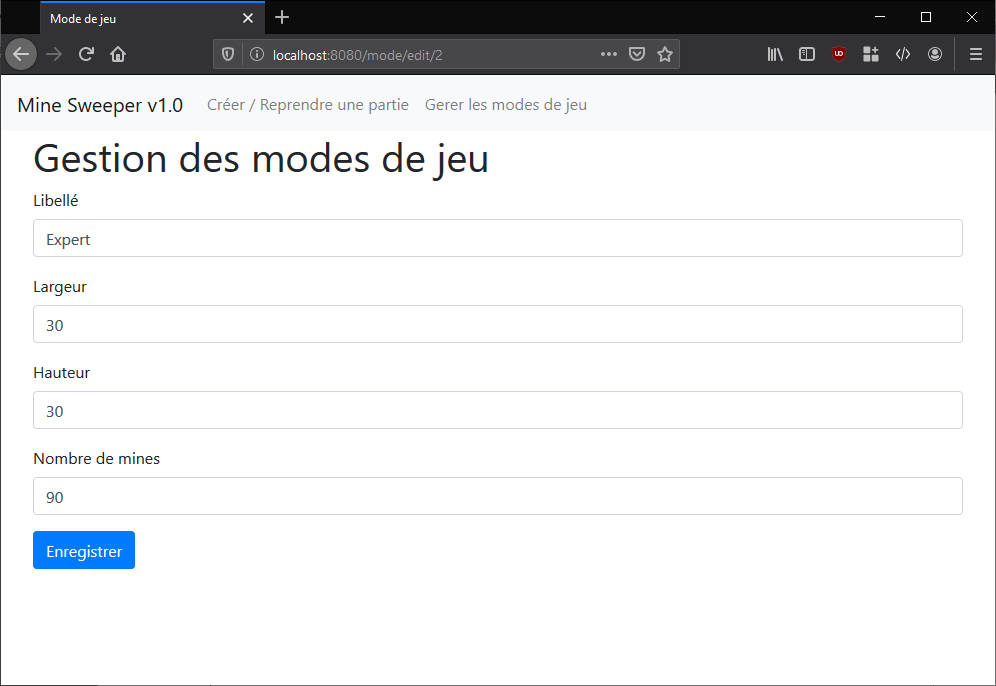
Les paramètres d’un mode de jeu sont les suivants :

* Titre : champ texte contenant la description du mode de jeu
* Largeur : largeur du champ de mine (nombre de case)
* Hauteur : hauteur du champ de mine (nombre de case)
* Nombre de mines placées sur le champ (forcément inférieur à L x H)

(Pensez à la gestion des erreurs !)

Exemple :

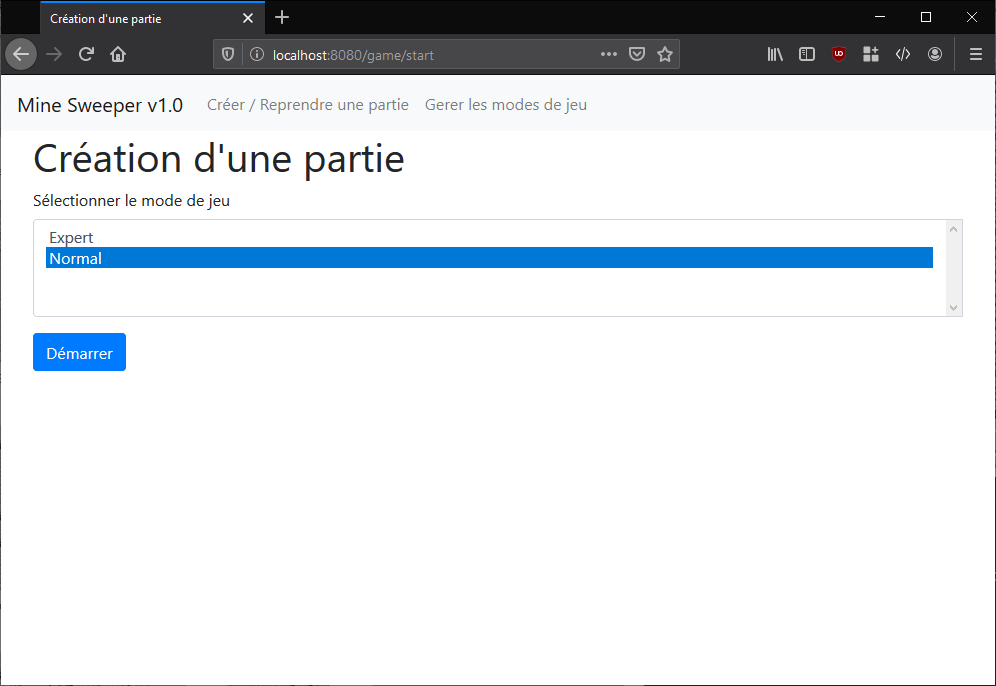




#### Création d’une partie

Il est à noter qu’il doit être impossible de lancer une nouvelle partie tant que la partie précédente n’est pas terminée.

1. Si ce n’est pas le cas, l’utilisateur est redirigé vers la dernière partie lancée pour qu’il puisse la finir.  
   (ceci est vrai, même si le navigateur a été fermé, ou que le serveur Java a été relancé)
2. Si c’est le cas, un formulaire permet de sélectionner un mode de jeu parmi les modes disponibles. Après sélection, la partie est créée, et l’utilisateur est redirigé vers l’écran de jeu



#### Déroulement de la partie

Lors de la création de la partie, une grille dépendant du mode de jeu est initialisée.

Puis les mines, dépendant également du mode de jeu, sont disposées de manière aléatoire sur le champ.

Le champ (tableau) est alors affiché « vide » à l’utilisateur, qui va devoir découvrir l’emplacement des mines.

La page affiche également le nombre de mines à découvrir, ainsi qu’un bouton pour abandonner la partie.

Il y a ensuite 2 modes pour cliquer sur les cases :

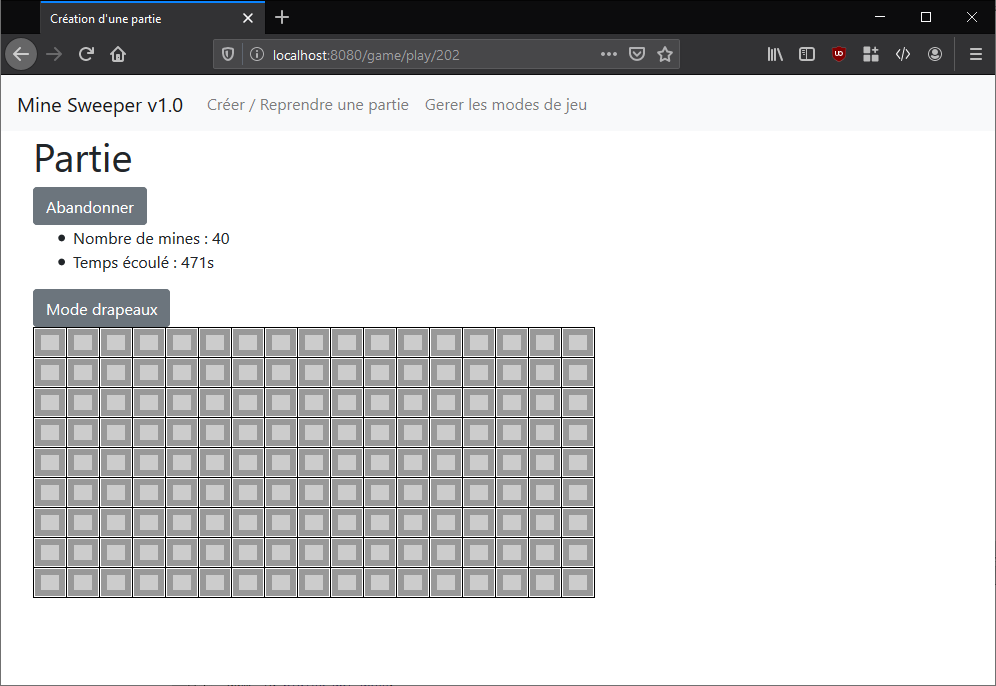
* Découvrir une case
* Ajouter un drapeau de déminage

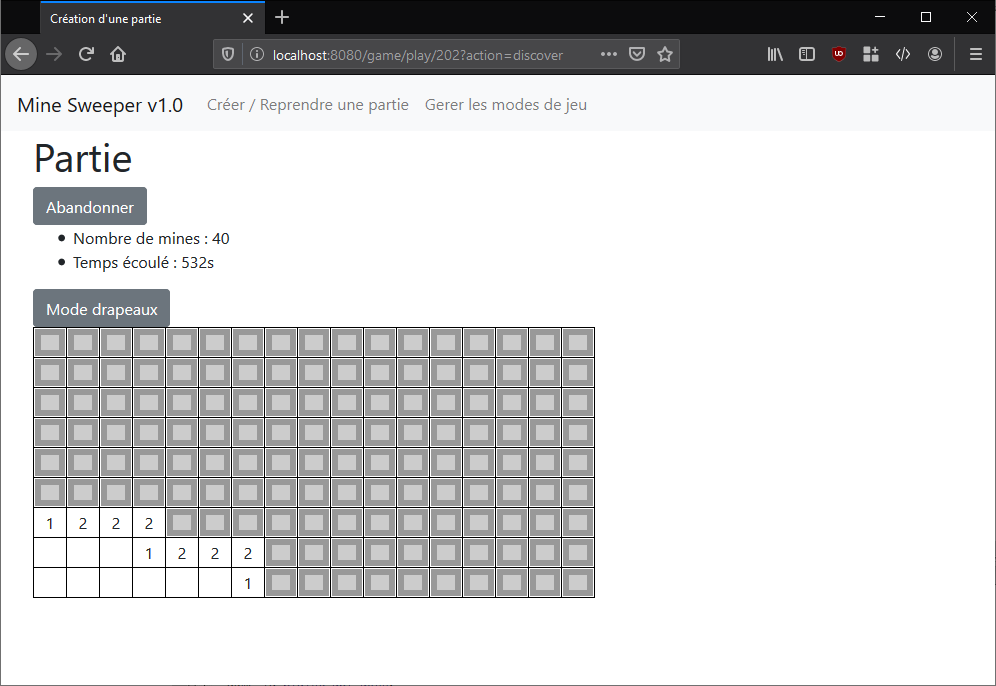
Découvrir une case :

Lorsque l’utilisateur clique sur une case, il y a plusieurs possibilités :

* Il a cliqué sur une mine : L’utilisateur a perdu, la partie est terminée, et l’emplacement des mines est révélé
* Il a cliqué sur une case avoisinant une ou plusieurs mines : le nombre de mine dans les cases directement autour (les 8 cases) est alors affiché, donnant des informations précieuses à l’utilisateur
* Il a cliqué sur une case n’avoisinant pas de mine : les cases avoisinantes sont alors découvertes automatiquement de proche en proche, jusqu’à afficher la limite du territoire sans mine :

Exemple de la découverte :

Image result for mouse pointer



Pour référence, cette « découverte » peut s’inspirer de l’algorithme : « Flood fill »

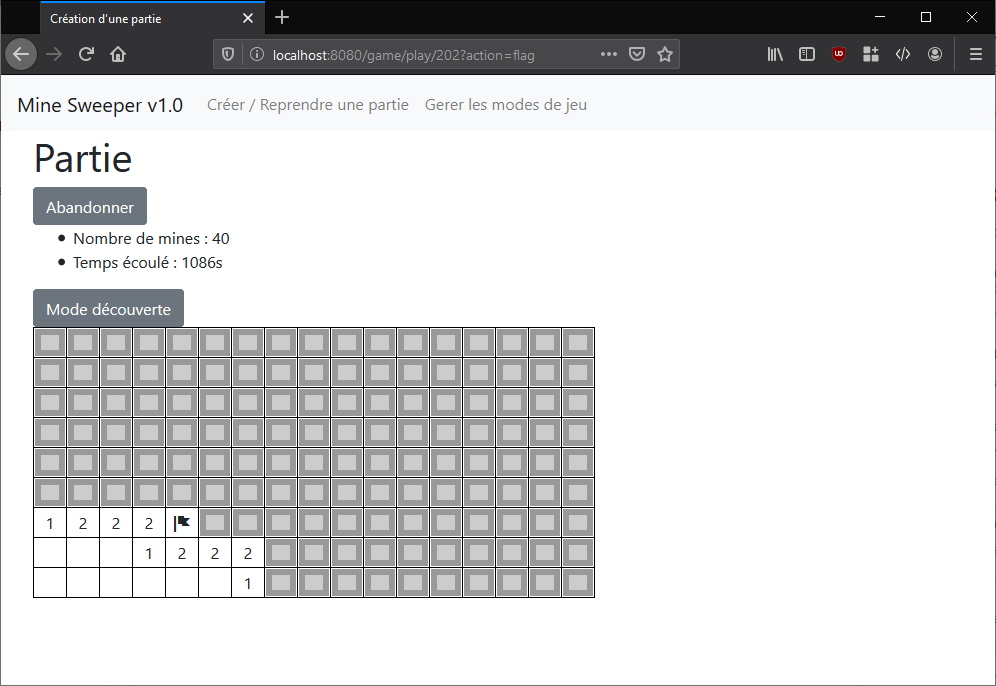
<https://fr.wikipedia.org/wiki/Algorithme_de_remplissage_par_diffusion>

Ajouter un drapeau de déminage :

Dans ce mode, quand l’utilisateur clique sur une case, elle affiche alors un drapeau de déminage ne permettant plus de cliquer par inadvertance sur la case. Ceci est utilisé par l’utilisateur pour marquer une case qu’il pense contenir une mine.

Si l’utilisateur clique sur une case possédant déjà un drapeau, celui-ci est supprimé, et il est de nouveau possible de découvrir la case.

Un clic sur une autre case n’a pas d’effet.

Image result for mouse pointer

Fin de la partie :

La partie est terminée quand toutes les cases ne possédant pas de mine sont découvertes.

L’écran de victoire est alors affiché, indiquant le temps écoulé entre la création de la partie et la victoire.

Si ce temps est dans le top 5 pour le mode de jeu, l’utilisateur peut saisir son nom afin de rentrer dans le classement.

Exemple d’une implémentation de « minesweeper » : <http://minesweeperonline.com/>