RAPPORT PROJET REVERSI

***Introduction :***

Le but de ce projet est de réaliser le jeu Reversi. Ce projet a pour objectif de réaliser ce jeu en utilisant les technologies Web telles que SpringBoot, Thymeleaf, et Heroku.

1. Tentative d’utilisation de Gradle

Nous avons perdu beaucoup de temps en cours à essayer d’utiliser Gradle sur nos PCs. Nous avons fini par choisir comme solution technique Maven car cette configuration était déjà en place sur nos PCs et nous avions certitude que cela fonctionne.

1. Analyse du jeu

Dans cette partie, nous avons étudié le fonctionnement du jeu pour comprendre les différents éléments dont nous aurions besoin pour créer le jeu. Nous avons aussi réalisé des pseudo-codes pour avoir les grandes lignes du fonctionnement du jeu.

1. Création des objets et des modèles

A partir de notre analyse, nous avons commencé à créer les entités nécessaires au fonctionnement du jeu. Nous avons donc mis en place les Cases, les Joueurs et la Partie dans des Controllers qui vont définir les méthodes et les attributs de chaque entité et les actions possibles sur chaque entité en fonction de l’URL tapée.

1. Création de la base H2

Il nous a fallu créer une base de données pour permettre la persistance des données. Nous avions le choix entre utiliser H2 ou PostGre. Nous avons choisi H2 au départ car elle s’intégrait facilement dans notre projet. Nous n’avons pas modifié ce choix car il répondait à nos attentes.

1. Affichage de l’othellier

Pour pouvoir tester les méthodes que nous avons implémentées lors de la création de nos modèles, nous avons commencé par afficher de façon sommaire les informations relatives aux entités dans la base de données. A partir de cela, nous avons commencé à mettre en forme ces informations pour obtenir l’aspect d’un othellier. Celui-ci contenait au départ l’état de la Case :

* -1 lorsque la Case était vide
* 0 lorsque le Joueur 0 avait un pion dans cette case
* 1 lorsque le Joueur 1 avait un pion dans cette case

1. Ajout des règles métiers du jeu

A partir du moment où notre affichage était fonctionnel, nous avons implémenté les règles métiers permettant de mettre en place le jeu ( Validité des coups, Nb de pions retournés, Ajustement des scores, … ). L’affichage nous permettait de tester la validité de notre implémentation en fonction du résultat attendu.

1. Amélioration de l’affichage graphique

Une fois que l’implémentation du jeu a été mise en place, il nous manquait plus qu’à améliorer le design de notre application.

***Problèmes rencontrés :***

* La mise en forme est une partie chronophage
* Les problèmes de build gradle nous ont beaucoup retardé et n’ont toujours pas trouvé de réponses
* La manipulation de thymeleaf nous a demandé un temps supplémentaire afin d’appréhender cet outil.

***Lancement du jeu :***

Pour lancer le jeu, il faut suivre les étapes suivantes :

* Lancer l’exécution du programme pour permettre la mise en place du serveur Spring
* Lancer le script permettant de créer les bases de données
* En allant sur localhost :8080/, une fenetre d’authentification apparait : le username attendu est *« player1 »* et le mot de passe est *« jeu »*
* Vous arrivez sur la page d’accueil du jeu permettant de commencer une nouvelle partie ou de reprendre une partie là où elle s’est arrêtée
* En cliquant sur *Commencer une nouvelle partie,* une partie est créée et vous permet de jouer à deux sur la même fenêtre.

***Pour aller plus loin :***

* Nous avons commencé la mise en place des sessions de jeu. En ayant plus de temps, nous aurions certainement réussi à mettre en place les authentifications nécessaires.
* Le design reste assez simple. L’utilisation de frameworks ( tel que Angular ) nous aurait peut-être permis d’obtenir un résultat plus user-friendly
* Nous avons travaillé sur l’intégration d’Heroku sur un petit projet annexe. Nous aurions voulu en faire de même pour ce projet.