**Algorithmique Avancée – Solo Noble**

DEMANGE Alessi – NICOL Benoît

**1 – Explication Générale du code**

**Classe Grille :**

Attribut Grille :

* nbLigne, nbColonne : entier | Ces attributs correspondent à la taille de la grille en fonction de ses lignes et de ses colonnes.
* grille : String[][] | Cet attribut correspond à la grille en question celle ou va évoluer les billes

Constructeurs Grille :

* Avec paramètres File : Le constructeur prend un objet de type « File » en paramètre qui correspond au tablier que l’on veut résoudre. Il lit le fichier et il convertit les charactères en String dans un tableau à double dimension. De plus, il initialise les autres attributs.
* Avec paramètre String[][] : Le constructeur prend le tableau a double dimension en paramètre pour le donner à l’attribut grille.
* Sans paramètre : Même chose que pour le constructeur avec attributs mais la grille est initialisée sous la première forme (celle de base du tablier1).

Méthode calculerTailleGrille :

* Méthode permettant de donner une valeur de la taille de la grille (sa largeur).

Méthode calculerNbBille :

* Permet de calculer le nombre de bille dans la grille.

Méthode deplacementValide :

* Méthode qui renvoie un booléen, si on peut déplacer une bille à des coordonnées données et à une direction donnée alors on renvoie vrai sinon on renvoie faux

Méthode deplacerBille :

* Méthode qui change de valeurs les cases lors d’un déplacement d’une bille

Méthode retourArriere :

* Méthode qui fait la même chose que deplacerBille mais en arrière (lors du BackTracking)

+ Getteur + Setteur

**Classe soloNoble :**

Attribut soloNoble :

* tablier : Grille | Cet attribut correspond à l’objet Grille qui va être rempli car un fichier dans le constructeur.
* solutions : TreeMap<Integer, String[][] | Cette attribut correspond au stockage des solutions pour ensuite afficher le tout.
* nombreDeplacement : entier |

Méthode resoudreSoloNoble :

* Méthode principale qui utilise la méthode déplacer bille. Elle permet, en utilisant le BackTracking, de trouver la solution parfaite pour résoudre le tablier qui est inscrit.

Méthode ecrireSolution :

* Méthode qui ajoute, dans la TreeMap solutions, une solution.

Méthode ecrireToutesSolution :

* Méthode qui affiche toutes les solutions les unes après les autres en visant dans la TreeMap solutions.

Méthode Main :

* Méthode qui nous permet d’exécuter le programme principal.

+ Guetteur

**2 – Rapport sur le projet**

Au commencement du projet soloNoble, il a fallu préparer un algorithme papier à l’aide du BackTracking pour pouvoir résoudre un tablier donné. Nous avons fait, tous les deux, un algorithme différent pour pouvoir mettre en commun notre vision d’un algorithme permettant de résoudre le problème. Ce choix nous a beaucoup aidé car nous avons obtenu beaucoup d’éléments pour commencer notre algorithme java.

La programmation java est devenu plus compliquée que l’algorithme papier. Notre plus grand problème était l’affichage des grilles lors qu’une erreur et d’un retour en BackTracking. Nous avons stocké nos solutions dans une variable de type TreeMap pour pouvoir afficher nos solutions les unes après les autres et enfin d’aller en arrière lors d’une erreur. Après avoir réglé le problème, un autre arriva un plus tard. Certains mouvements de billes n’étaient en joie de se produire. Les mouvements de