Rapport

• Ma partie pour l’instant consiste à :

- trouver les conditions pour faire un sudoku avec solution,

- créer un programme qui génère un sudoku aléatoire,

- créer un sudoku qui génère un sudoku aléatoire avec une solution,

- créer un programme qui teste si un sudoku rempli est correct.

• La dernière partie m’a semblé tout de suite plus facilement réalisable.

J’ai donc créer des méthodes pour tester la validité des lignes, colonnes et blocs.

J’ai testé mes méthodes à partir de plusieurs grilles dont une valide.

• Du coup, je me suis dit que pour obtenir un sudoku avec une solution, je pouvais partir d’une grille valide et enlever des éléments.

J’ai complété mon programme en générant des sudokus avec plus ou moins de valeurs en pensant que cela générerait une difficulté plus ou moins élevée.

Mais en fait, je ne suis pas sure que cela soit lié. Ainsi que le fait de partir d’une grille valide n’assure peut être pas que le sudoku n’ait qu’une solution.

Puis pour que la grille de départ ne soit pas toujours identique, j’ai opté pour lui appliquer des permutations.

• Pour créer un sudoku aléatoire, mon test de validité ne pouvait pas m’aider puisque la grille ne serait que partiellement remplie.

J’ai donc créer une méthode qui teste si une valeur est déjà présente sur une ligne ou sur une colonne ou dans un bloc.

Grâce à ces méthodes, j’ai pu construire un sudoku aléatoire.

• J’ai alors pensé que je pourrai utiliser ces méthodes pour remplir une grille de façon aléatoire,

avec la méthode ci-dessous :

public void remplissage(){ // Méthode qui permet de remplir une grille valide de façon aléatoire

int i, j, k;

i=0;

j=0;

Random rand = new Random();

while (i<9){

while (j<9){

k = 1 + rand.nextInt(9);

this.grille[i][j]=k;

if (elementligne(i,j,k)&&elementcolonne(i,j,k)&&elementbloc(i,j,k)){

j = j + 1;

}

else {this.grille[i][j]=0;}

}

j=0;

i = i + 1;

}

}

Le programme prend trop de temps car les chances d’obtenir une grille valide en mettant des valeurs au hasard sont trop faibles.

• Pour la suite, j’ai pensé utiliser le résolveur d’Antoine afin de fabriquer des sudokus valides.

Le but étant de fabriquer un sudoku ayant une unique solution.