Sabina ASKEROVA, Ekaterina ZAITCEVA

Programmation d’un jeu en Java : Kakuro

Programmation Orientée Objet 2, 2022

Le TD4 nous a servi d'exemple pour la réalisation du Modèle-Vue-Contrôleur. En plus pendant la dernière séance de TP nous avons discuté avec le professeur à propos du projet, donc le sujet était assez clair pour nous.

La seule fonctionnalité avec laquelle on a eu des difficultés est le solveur.
Ça nous a pris beaucoup de temps d'élaborer un algorithme pour l'implémenter.
Et malheureusement, on n’a pas pu le finir par manque de temps.

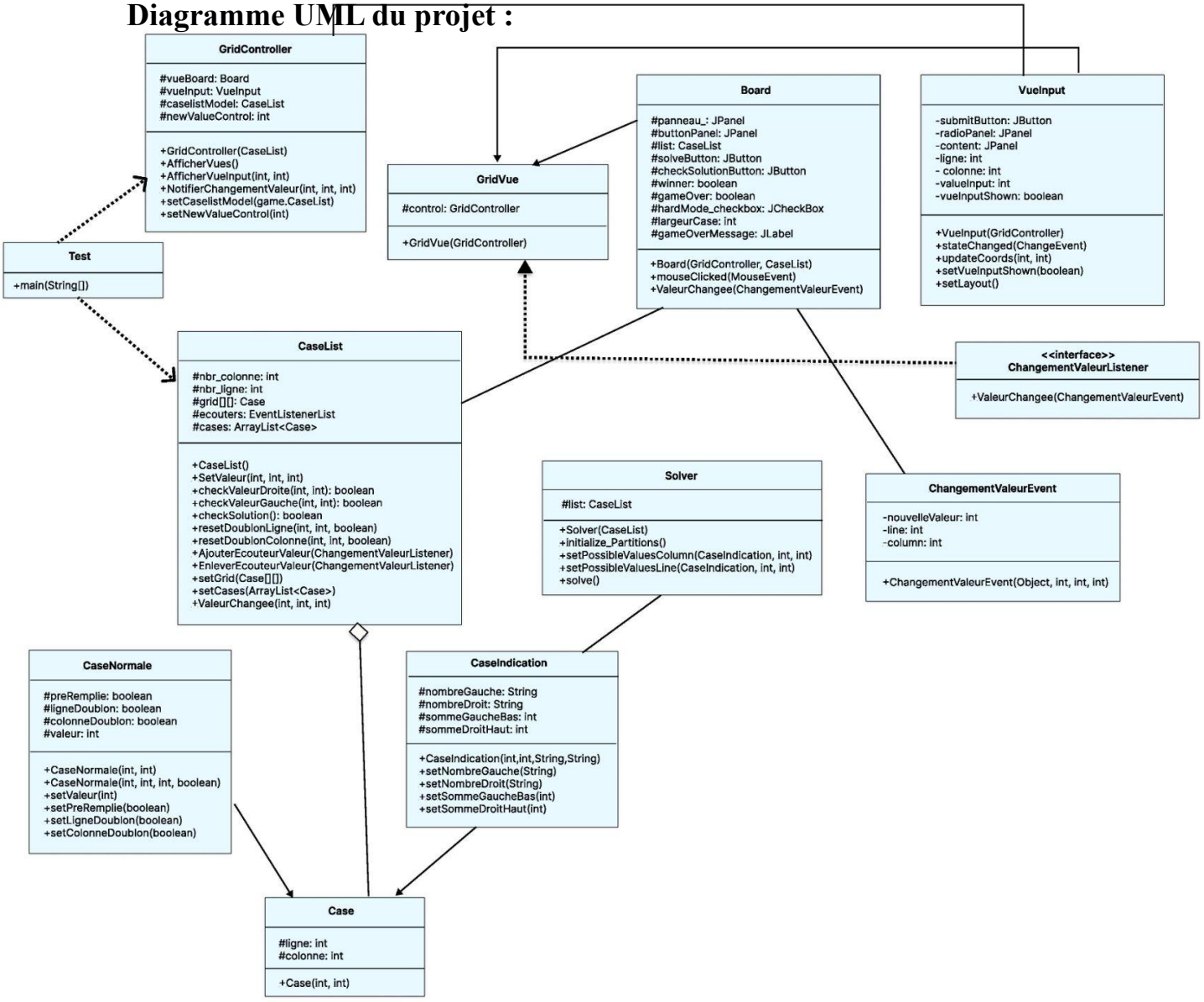
Voici est l’algorithme :

```
solveur:
tant que la grille CaseList n’est pas remplie
     pour chaque CaseNormale non remplie
          tab_l = obtenir valeurs possibles d'après la somme sur
          la ligne
          tab_c = obtenir valeurs possibles d'après la somme sur
          la colonne
          valeurs possibles d’une case = intersection(tab_l,
          tab_c)
          si il y a une seule valeur
               valeur de la case = valeur possible
          mettre à jour les tableaux de valeurs possibles de
          CaseNormale sur la même ligne est sur la même colonne
          que la case

mettre à jour les tableaux de valeurs possibles:
pour chaque combinaison de valeurs possibles sur la ligne
     si aucune de valeurs possibles d’une case de cette ligne
     n’est dans la combinaison
          supprimer le combinaison du tableau de combinaisons
     supprimer du tableau de valeurs possibles d’une case les
     valeurs qui ne sont présentes dans aucun combinaison
répéter pour chaque combinaison de valeurs possibles sur la
colonne
```

Quant à la répartition du travail, Sabina a commencé à travailler sur le projet plus tôt que Ekaterina. C'est donc elle qui a écrit la base de ce projet. Après Ekaterina a rejoint et nous avons corrigé les bugs ensemble, ajouté quelques nouvelles fonctionnalités et essayé d'améliorer la qualité du code en générale. Comme on l'a écrit plus haut, le solveur nous a pris beaucoup de temps, nous avons commencé son implémentation mais on n'a pas pu finir.

Diagramme UML du projet :



La grille du jeu est représentée par CaseList, une liste des CaseIndication et des CaseNormale.
CaseNormale est soit déjà remplie, soit à remplir. Elle est représentée par ses coordonnées dans la grille, sa valeur(0 si non-remplie) et le booléen qui indique si cette case est pré-remplie, donc il ne sera pas possible de modifier sa valeur.
CaseIndication contient ses coordonnées, sa somme gauche(sur la colonne) et sa somme droite(sur la ligne). Si il n'y a pas de somme gauche/droite, elle est notée par -1.

Fonctionnalités uniques :

- Marquage de doublons (Sabina)
  Si l'utilisateur saisit le chiffre qui existe déjà dans le même bloc(ligne/colonne), toutes les cases remplies de ce bloc deviennent rouges.
- Mode du jeu "difficile" (Ekaterina)
  A chaque vérification de solution si celui-ci est incorrect, le compteur des échecs incrément, après trois échecs la partie est finie(perdue).

Pour l'interface graphique nous avons opté pour la version assez minimaliste, donc l'utilisation de ce programme sera clair pour n'importe quel utilisateur.