

RAPPORT DE PROJET DE LANGUAGES À OBJETS AVANCÉS

Marie LAPORTE

Patrick CHEN

Choix des jeux

La compilation génère un unique exécutable qui demandera le type de jeu auquel on souhaite jouer avant qu'il ne commence.

En particulier, on doit choisir :

- Le type de jeu, donc soit le Taquin, soit le 2048, soit Sokoban. Si on choisi 2048, il faudra aussi préciser le type de variante.
- La taille de la grille (x et y).
- Si l'on souhaite une interface graphique ou non.
- Si l'on souhaite jouer par nous même ou faire jouer un robot.

Les variantes 2048

Nous avons fait les variantes suivantes :

- Avec 3 et 5 : les chiffres 3 et 5 peuvent désormais apparaître et fusionner entre eux.
- Destroy : une case destroy peut apparaître et détruire la première case avec laquelle elle rentre en contact.
- x2 et /2 : deux nouvelles cases peuvent apparaître pour multiplier ou diviser le contenu d'une case.

Fusionner des variantes

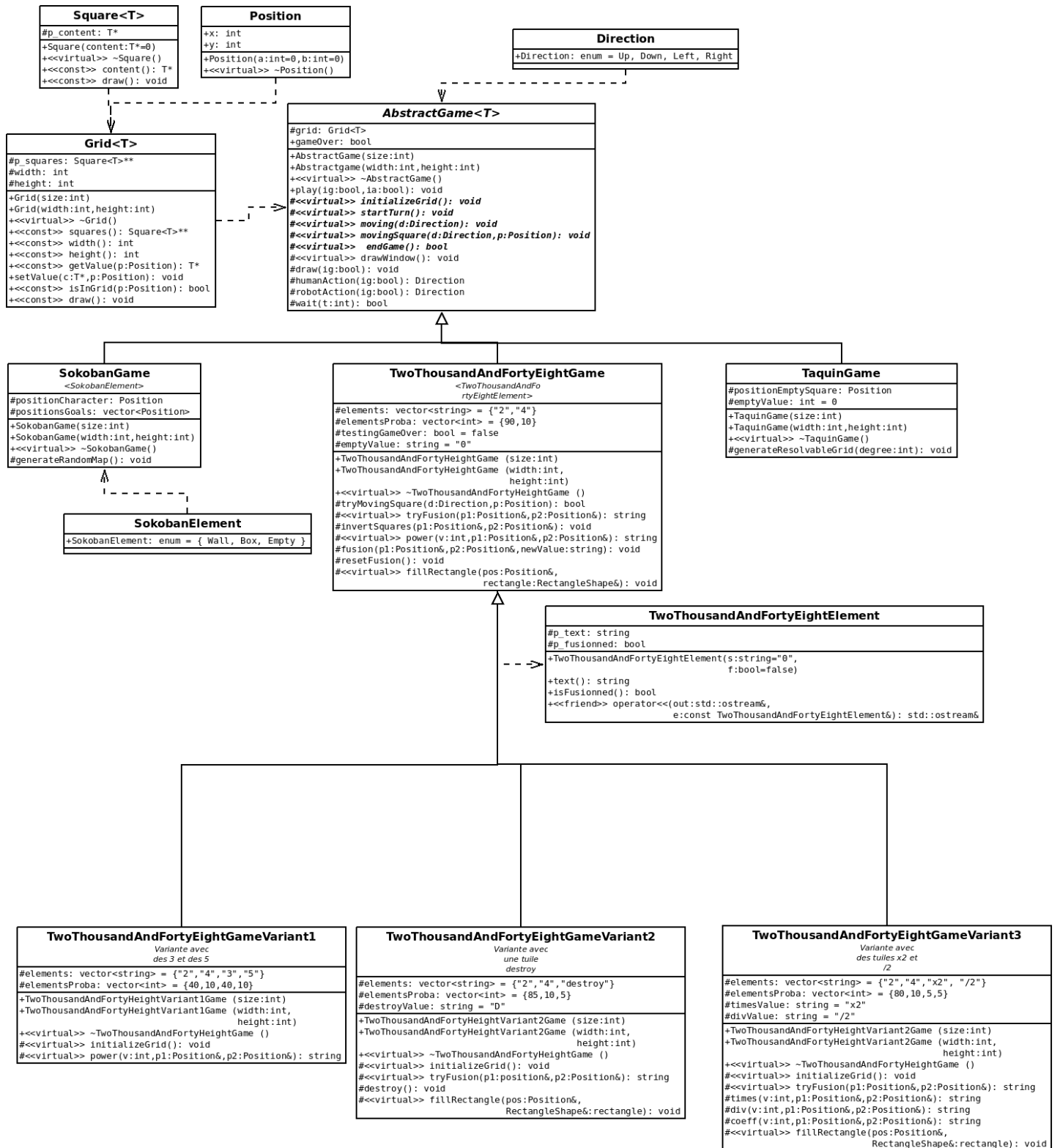
On a une classe pour chaque variante. Il est tout à fait envisageable d'avoir un 2048 avec plusieurs variantes en même temps, il suffirait d'utiliser l'héritage multiple et de bien gérer les conflits possibles.

Sokoban

Avec l'interface graphique, on assigne chaque élément à une couleur spécifique. Ainsi, on a

- Blanc -> Vide
- Bleu -> Murs
- Marron -> Boîtes
- Jaune -> Buts
- Vert -> Personnage

Diagramme de classes



Comment créer un jeu avec notre Framework

Un jeu sera une classe qui héritera de la classe abstraite ***AbstractGame<T>*** où ***T*** correspond au type de contenu qu'il y aura dans les cases de la grille du jeu. La seule méthode que l'on pourra utiliser après avoir instancié notre classe sera ***play()***, elle initialise la grille puis fait tourner le jeu, il ne faut pas la redéfinir. En revanche, ***AbstractGame<T>*** propose un grand nombre de **fonctions virtuelles pures** !

Il faudra impérativement compléter les méthodes suivantes :

- **initializeGrid()** : manière dont la grille doit être initialisée.
- **startTurn()** : actions à exécuter à chaque début de tour.
- **moving(Direction d)** : déplace l'ensemble des carreaux déplaçables.
- **movingSquare(Direction d, Position p)** : déplace un carreau isolé.
- **endGame()** : permet de déterminer si un jeu est terminé ou non.
- **drawWindow()** : uniquement pour l'interface graphique, permet de dessiner la grille.