Projet C++

Contenu

Ce programme comporte les jeux Sokoban, Taquin, et 2048, ainsi que de nombreuses variantes de ces deux derniers. Pour chacun de ces jeux, l'utilisateur choisit :

- les dimensions du plateau (avec un minimum imposé selon le jeu auquel on joue).
- le mode de jeu : joueur humain ou robot.

Parties traitées

Taquin

L'utilsateur peut choisir de jouer avec n'importe quel type d'objets qui soient comparables entre eux et sur lequel il existe une relation d'ordre. Par exmple, on peut choisir de jouer avec des entiers, mais aussi avec des lettres majuscules/minuscules, et on pourrait même imaginer d'autres types d'objets, représentés chacun par une classe, à condition que cette classe définisse :

- les opérations ++, -- (préfixe et postfixe)
- un constructeur prenant un entier en argument, un constructeur sans argument, ainsi qu'un constructeur de copie
- les comparaisons (==, !=, <, <=, > et >=)
- l'opérateur d'affichage

Outre la possibilité de choisir le type d'objets, le programme a été conçu de telle sorte à ne jamais générer de grilles insolubles: il faut pour cela que la permutation de la grille (nombre d'échanges à effectuer pour la réordonner) soit de même parité que la case vide (nombre de déplacements à effectuer pour la remettre à sa place). Dans notre programme, nous considérons que la bonne place de la case vide est en bas à droite de la grille. La partie ne se termine donc que toutes les cases sont replacées dans l'ordre croissant et que la case vide est en bas à droite.

2048

En plus du jeu original, nous avons également implémenté 6 autres variantes:

1. Avoir à la fois des 2, des, 3, des 5 et des 7,
2. Choisir une base b et ne jouer ainsi qu'avec des nombres de la forme $b*2^n$,
3. Avoir des nombres négatifs pouvant détruire leurs opposés,
4. Avoir des cases "x2", "x4", etc, qui multiplient les valeurs des cases avec lesquelles elles fusionnent, quelles qu'elles soient,
5. Avoir des cases "destroy", qui détruisent les cases avec lesqueles elles fusionnent,
6. Combinaison des variantes 1, 3, 4 et 5.

Dans le jeu original, ainsi que dans les variantes 1 et 2, deux cases ne peuvent fusionner que si elles sont égales.

Sokoban

L'utilisateur peut choisir un nombre de caisses. Les murs (internes et externes) sont disposés de manière totalement aléatoire, ainsi que les buts. On vérifie que:

- le personnage, les caisses et les buts ne soient jamais placés initialement en dehors des murs externes, ni dans une zone entourée de murs, sinon le jeu serait irrésolvable.

La partie se termine lorsque toutes les caisses ont été placées sur les buts. Cependant, le jeu peut être bloqué si une caisse se retrouve avec un mur devant et sur l'un des côtés (gauche/droite), car dans ce cas-là la caisse

ne peut plus être déplacée.

Problèmes

Taquin
Les ensembles d'objets avec lesquels on joue peuvent être finis. S'il est tout de même possible d'utiliser un ensemble E fini, on ne peut en revanche pas jouer sur des grilles dont la surface est supérieure à #E +1 (en tenant compte du fait que la case vide puisse être représentée par une valeur en dehors de cet ensemble). Par exemple, si on joue avec des lettres de l'alphabet, sachant qu'il n'y en a que 26, la surface du plateau ne pourra pas être supérieure à 27.

Sokoban
Le nombre maximal de caisses possibles dépend de la disposition du plateau. Comme on choisit de placer les caisses et de manière totalement aléatoire en vérifiant à la fois qu'on n'est ni en dehors des murs externes, ni voisin d'un mur interne, le fait de mettre un trop grand nombre de caisses peut provoquer une boucle infinie. Pour éviter cela, le nombre de caisses sur le plateau ne pourra pas être supérieur à sqrt(l*h)/2 pour une grille de dimensions l*h.

Quelques pistes d'extensions

Le robot implémenté pour chaque jeu joue de manière totalement aléatoire. Cela peut poser problème pour certains jeux, comme par exemple dans le cas du taquin, où cela provoque des boucles infinies. Pour cette raison, on aurait pu implémenter un robot intelligent, jouant le meilleur coup possible à chaque fois.

Diagramme des classes

Le diagramme des classes se trouve dans le fichier modelisation.pdf joint avec l'archive.