

Projet de Programmation objets orrientée avancée C++

TAQUIN

1	2	3
4		5
7	8	6

1	2	3
4	4	5
7	8	6

1	2	3
4		5
7	8	6

1	2	3
4	5	
7	8	6

1	2	3
4	5	
7	8	6

Felicitation !!! Robot		
vous avez réussi à faire le taquin		

2048

				16	4
				256	2
		8	64	1024	4
2	4	64	1024	16	16
	32	1024	2	16	
8	8	128	4	2	

				16	4
				256	2
		8	64	1024	4
2	2	4	64	1024	16
		32	1024	2	16
		16	128	4	2

				16	4
				256	2
		8	64	1024	4
	4	32	1024	2	32
2	2	16	128	4	2

16	2				
8	4				
4	256	2			
4	128	1024	4		
4	32	1024	2	32	
4	16	128	4	2	

	2		4		
	4				
16	256				
8	128	2	4		
4	32	2048	2	32	
8	16	128	4	2	

INTRODUCTION

Ce projet a pour but la construction d'une plateforme, ou framework, permettant d'implémenter plusieurs jeux proches : il s'agit de jeux de déplacement de blocs avec différents buts et modalités selon le jeu.

L'objectif de ce projet était de mettre l'accent sur l'utilisation de framework. Ainsi une bonne modélisation était la clef du développement de ces différents jeux, qui ne diffèrent enfin que de très peu.

Nous avons donc implémenter les jeux Taquin et 2048 avec deux variantes de ce dernier, avec un joueur humain et un joueur robot n'ayant aucune intelligence artificielle.

FONCTIONALITÉ

Le joueur choisit le jeu pour la partie (Taquin ?, 2048 ?), le joueur de la partie (humain ?, robot ?) puis la taille de la grille au lancement du programme ($n \geq 2$) et éventuellement la variante pour 2048 (Normale, Négative, 2048 avec des 2, 3, 4, 5)

TAQUIN

Le jeu fonctionne avec les touches numériques :

- 1 Pour déplacer la case vide vers la droite.
 - 2 Pour déplacer la case vide vers le haut
 - 3 Pour déplacer la case vide vers la droite
 - 5 Pour déplacer la case vide vers le bas
- Toute autre touche numérique est invalide

Si le joueur de la partie est un joueur humain 'intelligent' c'est à ce dernier de donner une direction de son choix, si cependant le joueur est robot, ce dernier donne une direction aléatoire.

Pour réussir ce jeu il faut que la case vide soit à la dernière position et que toutes les cases soient à la bonne position i.e la grille soit triée. Donc pour tester si le jeu est terminé on ne testera que si la case vide est en dernière position si oui on teste sinon on continue à jouer.

En cas de réussite les résultats sont stockés dans le fichier Score/Score_Taquin avec le nom du joueur, la taille de la grille, son score et la date et l'heure de fin du jeu.

NB: Une direction invalide ne fait rien donc le coup n'est pas compté, si par exemple la case vide est haut à gauche on ne peut pas aller plus haut ou plus à gauche.

2048

Le jeu fonctionne avec les touches numériques :

- 1 Pour incliner la grille vers la gauche
 - 2 Pour incliner la grille vers le bas
 - 3 Pour incliner la grille vers la droite
 - 5 Pour incliner la grille vers le haut
- Toute autre touche numérique est invalide

Les cases non vide de même valeur et n'y a pas de cases non vides entre eux se combinent.

Notre jeu n'implémente qu'une forme de déplacement qui est un déplacement vers la droite, pour tout autre direction on effectue une voire plusieurs rotations de la grille puis on fait un déplacement vers la droite et on rétablie le tableau, ce procédé est plus efficace est moins long a écrire que de faire un déplacement vers la droite , la gauche, le haut, le bas....

Pour réussir ce jeu il faut atteindre l'objectif étant 2048 .

Donc pour tester si le jeu est terminé on ne testera si l'objectif est atteint. En cas de réussite les résultats sont stockés dans le fichier Score/Score_2048 avec le nom du joueur, la taille de la grille, son score et la date et l'heure de fin du jeu. Un coup est toujours s'il ya au moins une case non occupée, si toutes les cases sont occupées le jeu ne se termine que si on n'a pas deux cases voisines de même valeur. Car si deux cases « voisines » de même valeur sur la même ligne ou la même colonne pourront toujours se combiner pour libérer une des deux cases.

Les variantes :

Variante Négative :

La variante négative est juste une particularité du jeu générant en plus des 2 ou des 4, des -2 ou des -4 avec une probabilité moins importante de générer des nombres négatives, donc deux cases de même valeur (positives ou négatives) se combinent et deux cases de même valeur en absolu et de signe contraire se 'détruisent' mutuellement en se combinant. Juste une redéfinition de deux méthodes de la classe mère est suffisante pour se faire.

2048 avec des 2 3 4 5

Cette variante génère en plus des 2 et des 4, des 3 et 5 la fusion reste possible entre deux cases de même valeurs engendré par la même valeur mais on ne se soucie pas de ce problème ici car il n y a aucun multiple de 3 ou 5 <2048 et étant puissance de 2 !!!

NB: Une direction invalide ne fait rien donc le coup n'est pas compté, si par exemple la case vide est haut a gauche on ne peut pas aller plus haut ou plus a gauche.

CONCLUSION

Le secret de ce projet reposait sur la bonne modélisation, la généricité et l'héritage , ainsi nous avons appris que l'utilisation du framework nous a permis de mieux 'factoriser le code' et de programmer de façon efficace et rapide.