

Rapport de conception du jeu 'Atom' en C

Dornier Simon

Desormeaux Charles-Adèle

Ce projet est un projet de type J, on peut donc choisir la taille du plateau, la solution finale est aléatoire et le jeu peut être résolu par une IA.

Introduction :

Pour pouvoir créer un plateau nous allons avoir besoin de pièces, pièces qu'il faudra pouvoir modifier et déplacer à notre guise ; pour cela nous allons définir un type piece de type int, qui nous permettra de pouvoir modifier ses bits pour définir les 4 lettres de chaque pièce. Nous allons ensuite créer un tableau de pièces à 2 dimensions où on stockera les coordonnées des pièces. Pour finir nous avons créé une fonction pour dessiner les coordonnées, les bords et les pièces du plateau.

Création du plateau :

Maintenant que nous pouvons créer un plateau, nous devons rendre le jeu intéressant, pour cela nous avons créé une fonction qui va : créer un plateau avec des pièces aléatoires, puis modifier ses dernières pour être sûr qu'il y ait au moins une solution finale, et pour finir mélanger ces pièces en les échangeant et les tournants. Pour cela nous allons appeler '*genererAleatoireFinissable*' qui va appeler '*genererAleatoireFinis*' qui appellera '*generer*' ; cette suite de fonctions va allouer le place nécessaire, puis générer un plateau avec des pièces aléatoires et les corriger en changeant leurs lettres pour qu'il y ai au moins une solution, puis pour finir les échanger ou les pivoter un certain nombre de fois demandé au joueur avec la difficulté.

Déroulement du jeu :

Une fois que nous avons un plateau jouable, on va commencer par demander au joueur quelle taille de plateau (entre 1 et 9) et quelle difficulté, ensuite on a deux solutions : l'IA ou jouer nous-mêmes ; nous parlerons de l'IA plus bas. Nous allons donc utiliser une boucle dont on sortira quand le plateau sera résolu. Dans cette boucle 3 choix : le reste, la rotation d'une pièce ou l'échange de deux pièces. La première fonction sert à remettre le plateau à zéro si on sent que nos modifications nous ont éloignés de la solution. Les deux autres servent à jouer et à chercher une solution pour ce plateau. Quand cette boucle est terminée la fonction '*jeu*' renvoie le score du joueur au '*main*' qui l'affichera au joueur.

IA :

Quand on choisi d'utiliser l'IA pour terminer le plateau automatiquement, on appelle la fonction '*resolve*' qui va copier le plateau actuel un nombre de fois entré en paramètres, afin de le résoudre. L'IA essaie de trouver une solution en testant plusieurs possibilités avant de trier les plateaux obtenus par score et de recopier ceux ayant le moins de collision. L'algorithme se base sur l'idée que les modifications et sélections des meilleurs plateaux de manière successive converge généralement vers une solution. Pour se faire, la méthode '*rndMode*' est appelée afin d'effectuer des actions aléatoires sur les copies courantes et la fonction '*copieTab*' pour copier les meilleurs tableaux dans les tableaux courants. La méthode '*collisions*' est appelée à chaque copie de tableau afin de l'évaluer puis de le ranger via un tri par insertion. L'algorithme boucle en testant à chaque itération si le nombre de collisions a atteint 0 pour le meilleur tableau de la liste, et si oui, aucune itération de plus n'est nécessaire et on renvoie donc le dernier plateau trouvé, celui-ci étant une solution du plateau d'origine.

Conclusion :

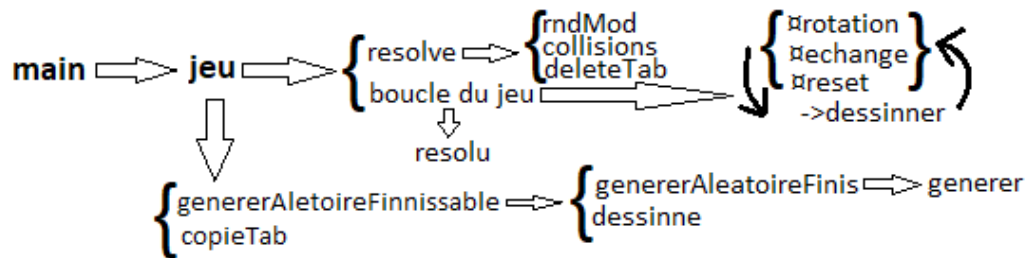
Nous avons donc un programme qui permet de générer un plateau aléatoire avec au moins une solution finale, sur lequel on va soit pouvoir faire tourner une IA qui résoudra à notre place, soit jouer nous-mêmes en déplaçant et en pivotant des pièces jusqu'à trouver une solution et obtenir un score.

Bugs connus : -Le programme permettant de déplacer les pièces aléatoirement ('genererAleatoireFinissable') peut rendre un plateau déjà fini et finir le jeu instantanément.

-L'IA ne fonctionne qu'avec des petits plateaux, si on dépasse 3x3 on risque de prendre beaucoup de temps et déclencher un 'segmentation fault'.

-Si on rentre un int de taille 2 ou 4 dans le choix entre l'IA et le jeu la boucle de jeu va tourner une fois en passant par la rotation mais avec des valeurs incorrects.

Schéma de fonctionnement :



-main : Appelle jeu et affiche le score.

-jeu : Initialise le plateau, fait une sauvegarde puis demande si le joueur veut jouer ou déclencher l'IA.

-genererAleatoireFinissable : Crée un plateau aléatoire finissable et l'affiche.

-genererAleatoireFinis : Crée un plateau aléatoire déjà fini.

-generer : Alloue la place nécessaire pour le plateau.

-copieTab : Crée une sauvegarde du plateau généré.

-resolve : Cherche une solution au plateau.

-rndMode : Effectue une rotation ou un échange aléatoire.

-collisions : Renvoie le nombre de collisions restantes.

-deleteTab : Supprime un plateau.

-boucle du jeu : Fait tourner le programme jusqu'à que le joueur trouve une solution.

-dessinner : Dessine le plateau actuel dans la console.

-reset : Récupère le plateau créé par copieTab pour recommencer le jeu.

-rotation : Tourne une pièce.

-echange : Échange deux pièces.