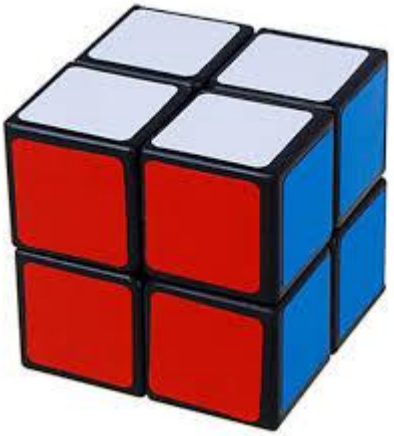
Projet Rubik’s Cube 2*2
- C avancé -

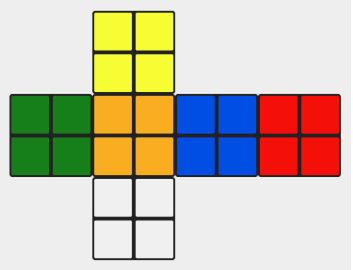


Réalisé par Nicolas Deronsart

I/ Description du projet

Dans l'objectif de créer un programme en langage C capable de résoudre n'importe quel Rubik's Cube 2*2, j'ai dû me lancer dans des recherches pour comprendre les techniques de résolutions utilisées par l'Homme.
Pour cela de nombreux sites m'ont été utiles, notamment les 2 suivants : How to Solve a 2x2 Rubik's Cube - The Pocket Cube, Le Rubik's Junior pas à pas, en une page.

Pour la réalisation, il a tout d'abord fallu créer les structures nécessaires à la modélisation du Rubik's Cube dans le module structures. Le cube est ainsi représenté par une matrice cube d'entiers de la forme int[6][2][2]. Le 6 pour les différentes faces du cube, et 2 pour le nombre de lignes et de colonnes par face. Cube qui sera initialement résolu. Une structure de liste chaînée est également codée pour sauvegarder les étapes de résolution du cube.
J'ai choisi un affichage sous forme de patron pour le représenter de la meilleure manière possible dans les limites de la 2ème dimension.



La face orange ici étant la face FRONT (devant), la verte étant la LEFT (gauche), la bleue est la RIGHT (droite), la rouge est la BACK (l'arrière), la jaune la UP (le dessus), et la blanche est la face DOWN (face du dessous).

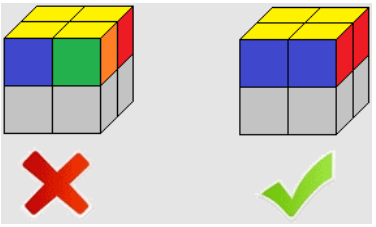
J'ai ensuite codé les différentes rotations suivantes possibles dans le module rotations, tout en faisant attention à bien respecter l'affichage de la face au dos où les colonnes sont inversées lors des rotations L et R. Ces rotations seront mémorisées en utilisant les lettres définies ci-dessous dans une liste chaînée contenant également l'état actuel du cube après la rotation en question.

II/ Les étapes de résolution

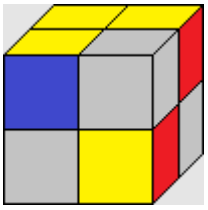
Pour résoudre un cube mélangé aléatoirement à l'aide des rotations, j'ai suivi une méthode en 3 étapes que je vais désormais expliquer.
Les programmes nécessaires à la résolution du Rubik's Cube 2*2 se trouvent dans le module rubiksCube.

Etape 1 : firstFace(cube, list);

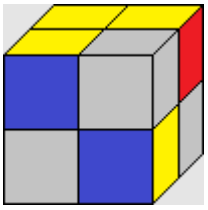
L'objectif de cette première étape est de résoudre la face du haut en respectant la correcte position des coins comme montré dans l'exemple ci-contre. Chaque coin doit être bien placé tel que à la fin de l'étape la couronne haute soit déjà résolue.



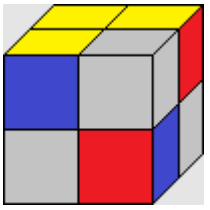
Pour cela la fonction cornerPosition nous indique la position du coin recherché dans le cube. Selon cette position le programme déplace chacun des 3 premiers coins un à un pour les mettre à leurs correctes positions. Le 4ème coin est ensuite placé en utilisant l'un des algorithmes suivants selon le cas.



FDF'



R'D'R

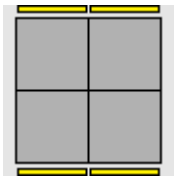


R'DDRDR'D'R

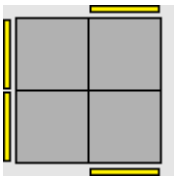
Etape 2 : secondFace(cube, list);

Cette seconde étape consiste à placer la face du dessous, la face jaune. À la sortie de l'étape précédente 7 configurations différentes sont possibles. Pour chaque cas, il a fallu appliquer un algorithme de résolution différent.

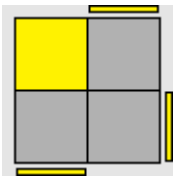
Cas 1:



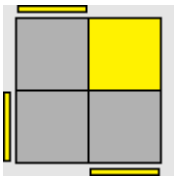
Cas 2:



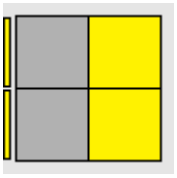
Cas 3:



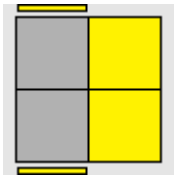
Cas 4:



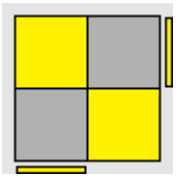
Cas 5:



Cas 6:



Cas 7:



Etape 3 : permutingLastPieces(cube, list);

Cette dernière étape permute les dernières pièces toujours mal placées dans la couronne basse du cube.

Pour cela il suffit de chercher 2 coins de même couleur côte à côte dans la couronne basse à l'aide de la fonction correct2Pieces. Si il n'y en a pas il faut appliquer l'algorithme ci-dessous et ils apparaîtront.
Une fois ces 2 coins trouvés, il faut faire des rotations D jusqu'à compléter notre 3ème face, puis il suffit d'orienter le cube de manière à ce que cette 3ème face se trouve à l'avant puis appliquer l'algorithme suivant : **R'BR'FFRB'R'FFRRD'**.

Le Rubik's Cube est désormais résolu !

III/ L’interface

Pour compiler le programme il suffit d’ouvrir le terminal dans le dossier src et d’exécuter la commande make. L’exécutable exec est désormais disponible dans le dossier père, il faut désormais ouvrir le terminal dans ce répertoire parent et utiliser ./exec pour l’exécuter.

L’interface apparaîtra alors et vous permettra d’accéder aux différentes possibilités du programme.

- Dans le 1er choix vous pourrez tester librement les rotations sur un Rubik’s Cube 2*2 puis tenter d’appliquer l’algorithme de résolution sur le cube créé.

- Dans la 2ème possibilité, vous pourrez mélanger aléatoirement le cube en choisissant le nombre de rotations puis observer les étapes de la résolution si vous le souhaitez.

- La dernière option permet de tester l’algorithme sur autant d’itérations que vous le souhaitez (dans la limite de 1000000) dans le but de connaître une approximation de la moyenne de coups pour la résolution

Pour les 2 premiers modes vous pourrez également enregistrer dans des fichiers se trouvant dans le sous-dossier txts les états du cube ainsi que les étapes de résolutions détaillées, ou simplement la succession des rotations effectuées.