



Rubik

Qu'est-ce que vous faites mercredi prochain ?

François Sechet fsechet@student.42.fr
42 Staff pedago@staff.42.fr

Résumé: Ce projet va vous faire écrire un programme qui résout des Rubik's Cubes en un minimum de mouvements.

Table des matières

I	Préambule	2
II	Introduction	3
III	Objectifs	4
IV	Consignes générales	5
V	Partie obligatoire	6
VI	Partie bonus	8
VII	Rendu et peer-évaluation	9

Chapitre I

Préambule

En soutenance, vous allez devoir faire mieux que ça :

<https://www.youtube.com/watch?v=eQH7MU0gUCQ>

A défaut, ceci reste une performance acceptable :

https://www.youtube.com/watch?v=_v85dcvw2vQ

Par contre, ça, c'est trop facile :

<https://www.youtube.com/watch?v=X0pFZG7j5cE>

Chapitre II

Introduction

Dans ce projet, on vous demandera de résoudre des Rubik's Cubes, selon les critères des compétitions officielles de Rubik's Cube et en particulier celles du Fewest Moves Challenge (FMC), la limite de temps habituelle d'une heure en moins (votre limite de temps se comptera en secondes).

Lors d'une épreuve de FMC, tous les compétiteurs sont enfermés dans une salle avec un papier, un stylo, un cube et c'est tout. Pendant soixante minutes, ils vont individuellement chercher la solution la plus courte au mélange donné, et la rendre sur papier. Les solutions seront ensuite individuellement vérifiées, et sera proclamé vainqueur celui avec le score le plus bas.

Bon courage !



Le record du monde actuel de FMC est de 20 mouvements, par Tomoaki Okayama. Plus de 200 personnes ont déjà trouvé des solutions en-dessous des 30 mouvements en compétition.

D'autre part, il est prouvé que dans le pire des cas, pour chaque cube, il existe au moins une solution en au plus 20 mouvements. Plus d'informations: <http://www.cube20.org/>



Il est vivement encouragé d'apprendre à résoudre le Rubik's Cube IRL pour ce projet.

Chapitre III

Objectifs

Tout le monde connaît le Rubik's Cube, et il y a de fortes chances pour que vous vous soyez déjà un peu esquiné ce qui vous sert de cerveau dessus à un moment dans votre vie.

Que vous ayez ou non déjà réussi à le résoudre, ce projet est à votre portée de jeune programmeur enthousiaste. Il s'agit en effet d'un simple projet d'algorithmie, faisant un peu appel à de la représentation dans l'espace, à quelques vagues notions de théorie des groupes, un peu aussi à votre cerveau, bref, des bricoles.

Chapitre IV

Consignes générales

Vous êtes libre d'utiliser le langage de votre choix (C, C++, Perl, Python, Java, Brainfuck si ça vous amuse?). Aucun segfault, fuite mémoire, double free, boucle infinie, interruption non gérée avec discernement ne seront tolérés, à plus forte raison parce que ce n'est clairement pas le plus difficile dans ce projet.

Le cas échéant, vous devrez fournir un Makefile avec les règles habituelles pour compiler votre programme.

Vous avez le droit d'utiliser toute librairie que vous pouvez justifier en soutenance. Bien entendu, toute utilisation d'une librairie ou ressource externe faisant vraiment le boulot à votre place n'est pas autorisée, vous devez justifier votre propre algorithme en correction.

Vous devez être en mesure d'expliquer votre algorithme avec des mots et des concepts visuels simples.

Chapitre V

Partie obligatoire

Votre programme doit accepter en paramètre un mélange.

Un mélange est une séquence de mouvements séparés par un ou plusieurs espaces à appliquer sur un cube de manière à le mélanger non pas aléatoirement mais selon un pattern prédéfini.

La notation utilisée est la notation internationale (F R U B L D pour Front / Right / Up / Back / Left / Down). Pour comprendre cette notation, je vous redirige vers <http://www.francocube.com/notation.php> qui me semble être un bon point de départ, le reste du site étant également très riche d'enseignements pour ce projet. Je vous invite également à bien regarder la vidéo de ce projet.



Une notation alternative existe, d'origine belge. Son utilisation est source de beaucoup de confusion, ce qui explique son manque de succès au niveau international. Vous ne devez pas l'utiliser pour ce projet. La voici : D D D D D D pour Dessus / Dessous / Devant / Derrière / Droite / De l'autre côté.

F R U B L D représente à la fois une face et un mouvement appliqué à cette face, mais la nuance est généralement assez explicite. Vos programmes doivent accepter les modificateurs de mouvements ' et 2 et renvoyer une solution utilisant cette même notation.

Voici un exemple de séquence valide : R2 D' B' D F2 R F2 R2 U L' F2 U' B' L2 R D B' R' B2 L2 F2 L2 R2 U2 D2



Les mouvements de tranches M, E et S ainsi que les rotations x, y et z ne sont pas acceptés.

Votre programme doit retourner sur la sortie standard les séquences de mouvements à appliquer sur un cube de type 3x3x3 préalablement mélangé avec la séquence donnée.

Par exemple :

```
$> ./rubik "F R U2 B' L' D'" | cat -e
```

Retourner l'inverse du mélange (ou variante directe avec insertion de séquences sans effet pour en modifier la longueur) s'apparente grandement à un cas de triche...

Vous devrez comme toujours gérer les erreurs de manière sensible.

La métrique utilisée pour compter les mouvements est une métrique par demi-tour (HTM), à savoir qu'un quart de tour ou demi-tour sur une même face comptent pour un seul mouvement, mais les mouvements de tranches comptent pour deux mouvements. Bref, votre score sera le résultat d'un wc -w sur votre solution.

```
$> ./rubik "F R U2 B' L' D'" | wc -w
```


Chapitre VI

Partie bonus

Tant que les contraintes de la partie obligatoire et des consignes sont respectées, vous pouvez ajouter tous les bonus que vous souhaitez, qui seront jugés à la discrétion de vos correcteurs.

Par exemple :

- Une vraie représentation graphique de l'état du cube à intervalles réguliers (ncurses, minilibx, openGL ? ce que vous voulez - une simple série de chiffres ou de lettres est considéré comme du debug, pas vraiment comme un réel bonus...), voire une représentation visuelle des faces qui tournent en temps réel (ça c'est vraiment classe !)
- Un algorithme qui descend aux frontières des solutions les plus optimales dans un délai qui reste raisonnable (au-delà de quelques secondes, ce n'est plus raisonnable)
- Un choix entre plusieurs algorithmes, voire une sélection de la meilleure solution entre plusieurs algorithmes
- Un générateur de mélanges intégré dont on peut spécifier la longueur et/ou le nombre
- Une subdivision de votre solution en sous-étapes "humainement compréhensibles"
- Un fonctionnement avec d'autres puzzles (4x4x4, 2x2x2, Megaminx, Square-1 ?)
- De réelles performances du corrigé à la résolution d'un rubik's cube en soutenance
- ...

Les bonus ne seront comptabilisés que si la partie obligatoire est parfaitement réussie.

Ces bonus ne doivent pas outrepasser le fonctionnement normal du programme (i.e. l'input des mélanges doit toujours être une séquence valide), d'éventuelles options préalables peuvent être indiquées précédées d'un '- '.

Chapitre VII

Rendu et peer-évaluation

Ce projet sera corrigé uniquement par des humains.

Les correcteurs seront particulièrement vigilants aux cas de triche qui entraînent systématiquement, dans les compétitions officielles de Rubik's Cube, le bannissement de toute compétition pendant plusieurs années. A 42, nous nous contenterons de simples séances d'humiliation publique et de la fourniture de sucreries pour tout le cluster.

Contrairement à vos collègues en compétition de FMC, vous avez le droit de vous aider de google, de vos voisins de gauche ET de droite ou du forum. Mais à l'instar de vos collègues en compétition de FMC, essayez de vous servir prioritairement de votre cerveau...