

T.I.P.E: Rubik's Cube

2x2x2

I) Définitions des constantes et opérations de bases.

Les constantes sont en principe écrites tout en majuscule, à l'exception des plus usitées qui sont alors tout en minuscule.

► *On charge tout d'abord le module utilisé:

A) Les couleurs

► *Tableau (C) des 6 couleurs utilisées dans le cube (version officielle):

► *Attributions des couleurs à chaque indice de C (initiales anglaises pour éviter les confusions bleu/blanc):

B) Le Rubik's Cube

Le Rubik's Cube est un tableau composé par chacun de ses 8 petits cubes.

Les petits cubes sont eux même des tableaux composé de leurs coordonnées et des couleurs de leurs 3 faces.

Les coordonnées sont eux aussi des tableaux donnée par 3 chiffres (d'après les notations adopté).

Les couleurs sont finalement toujours des tableaux donnée par les 3 couleurs des faces du petit cube concerné.

► *Definition d'un entier max (pour la fréquence max de BougerRubix plus loin):

► *Définitions des 8 cubes d'un Rubik's Cube fini:

► *Notations du premier tableau d'un cube (les coordonnées) et du second (les couleurs):

► *Définitions du Rubik's Cube fini:

► *Fonction de correspondance n° Petit Cube \Rightarrow Coordonnée:

► *Fonction de correspondance Coordonnée \Rightarrow n° Petit Cube:

► *Fonction de correspondance Couleurs \Rightarrow n° Petit Cube:

► *Fonction de correspondance Couleurs $\Leftarrow n^{\circ}$ Petit Cube:

- ▶ *Fonction de rafraîchissement du Rubik's cube, qui le place dans sa notation standard:

C) Les états du Rubik's Cube

α) La finitude

- ▶ *Fonction testant si le Rubik's Cube considéré est fini:

β) La faisabilité

Comme il existe une suite de mouvement permettant d'interchanger 2 petits cubes quelconques sans changer leur orientation, on ne considère que les orientations des petits cubes.

Il est alors nécessaire et suffisant d'avoir:

- $\forall \text{Cube1, Cube2} \in \text{Rubik}, \{\text{Cube1}[\text{coul}]\} \neq \{\text{Cube2}[\text{coul}]\}$
- $\forall \text{Cube} \in \text{Rubik}, \exists \text{CUBE.FINI} \in \text{RUBIK.FINI} \mid \exists n \in \mathbb{N} \mid \text{Cube}[\text{coul}] = c^n(\text{CUBE.FINI}[\text{coul}])$: où $c(E)$ est un cycle de l'ensemble E
- $\sum n \equiv 0 \pmod{3}$

- ▶ *Fonction qui vérifie l'axiome 1:
- ▶ *Fonction qui vérifie si un triplet de couleur est cycle du rubik fini:
- ▶ *Fonction qui vérifie l'axiome 2 et 3:
- ▶ *Fonction qui vérifie si un cube est faisable:

II) Les 12 mouvements possibles

A) Selon la face gauche

- ▶ *Fonction qui change les données d'un petit cube ayant subi une rotation du bloc L(left) (dans le sens direct):
- ▶ *Fonction qui applique celle qui précède à tous les petits cubes du rubik's cube et renvoie le nouveau:
- ▶ *Fonction qui fait cette fois-ci l'opération dans le sens indirect:
- ▶ *Notation:

B) Selon la face droite

- ▶ *Fonction qui change les données d'un petit cube ayant subi une rotation du bloc R(right) (dans le sens direct):
- ▶ *Fonction qui applique celle qui précède à tous les petits cubes du rubik's cube et renvoie le nouveau:
- ▶ *Fonction qui fait cette fois-ci l'opération dans le sens indirect:
- ▶ *Notation:

C)Selon la face basse

- ▶ *Fonction qui change les données d'un petit cube ayant subi une rotation du bloc B (basse) (dans le sens direct):
- ▶ *Fonction qui applique celle qui précède à tous les petit cube du rubik's cube et renvoie le nouveau:
- ▶ *Fonction qui fait cette fois ci l'opération dans le sens indirect:
- ▶ *Notation:

D)Selon la face haut

- ▶ *Fonction qui change les données d'un petit cube ayant subi une rotation du bloc H(haut) (dans le sens direct):
- ▶ *Fonction qui applique celle qui précède à tous les petit cube du rubik's cube et renvoie le nouveau:
- ▶ *Fonction qui fait cette fois ci l'opération dans le sens indirect:
- ▶ *Notation:

E)Selon la face posterieur

- ▶ *Fonction qui change les données d'un petit cube ayant subi une rotation du bloc P (posterieur) (dans le sens direct):
- ▶ *Fonction qui applique celle qui précède à tous les petit cube du rubik's cube et renvoie le nouveau:
- ▶ *Fonction qui fait cette fois ci l'opération dans le sens indirect:
- ▶ *Notation:

E)Selon la face avant

- ▶ *Fonction qui change les données d'un petit cube ayant subi une rotation du bloc A (avant) (dans le sens direct):
- ▶ *Fonction qui applique celle qui précède à tous les petit cube du rubik's cube et renvoie le nouveau:
- ▶ *Fonction qui fait cette fois ci l'opération dans le sens indirect:
- ▶ *Notation:

III)Les outils

A)Le patron

- ▶ *Fonction qui dessine un petit carre:
- ▶ *Fonction qui trace le patron d'un Rubix:

B)Le volume

- ▶ *Fonction qui renvoie les plot d'un petit cube donnée:
- ▶ *Fonction qui dessine un rubick's cube donnée:

C)La création d'un rubix cube

- ▶ *Fonction qui creer un rubik's cube à partir de ses couleurs donnée dans l'ordre de référence:
- ▶ *Fonction qui creer aléatoirement un rubik's cube:

D)Changement d'origine

- ▶ *Fonction qui change le petit cube n°1 avec celui donné comme origine:

E)Deplacement du Rubick's Cube

- ▶ *Fonction qui execute une série de mouvement sur un rubick's cube:

IV)La résolution

A)Les mouvements clefs

- ▶ *Mouvement de base:
- ▶ *Mouvement d'échange:
- ▶ *Mouvement d'orientation:

B)Echange de 2 petits cubes

- ▶ *Mouvement pour échanger un seul petit cube avec le n°1:
- ▶ *Mouvement pour échanger n'importe quel cube entre eux:

C)Changement de l'orientation de 2 petits cubes

- ▶ *Mouvement pour changer l'orientation d'un petit cube et du n°1:
- ▶ *Mouvement pour échanger n'importe quel cube entre eux:

D)Résolution

- ▶ *Fonction qui place les petits cubes au bon endroit selon leur couleur:
- ▶ *Fonction qui oriente les petits cubes correctement:
- ▶ *Fonction tant attendu de résolution du rubick's cube:
- ▶ *Fonction finale de résolution d'un rubix crée:

E)Tentative (vaine) de simplification

Vu le nombre d'opération neccessaire pour résoudre le rubick's cube avec la méthode précédent, on tente de faire un programme qui les simplifierait.

- ▶ *Fonction de teste de l'égalité de deux rubick's cube:
- ▶ *Fonction de simplification d'une suite de mouvement:

V)Autre méthode de résolution

Cette nouvelle méthode se base sur un tri progressifs des principaux cas rencontré dans la résolution du rubick's cube (selon le guide officielle).Error, missing operator or `;`

A)Préliminaires

A nouvelles méthodes, nouveaux outils. Ceci est une suite d'amélioration apporté aux outils de la 1ère partie. En revanche, il faut en user avec parcimonie, nottament de bougerII qui a une facheuse tendance (depuis maple 13) à vite faire planter le système...

- ▶ *Déclarations des coordonnées références (en vue des les faires "roter"):
- ▶ *Fonction effectuant une rotation d'une face et d'un angle donné (renvoie des coordonnées):
- ▶ *Nouvelle version de DessineCube pour la rendre compatible avec les rotations:
- ▶ *Et donc de même une nouvelle version de DessineVolume:
- ▶ *Pour finalement amélioré BougerRubix afin d'afficher les rotations (pas en abuser):

B)Phase 1

Elle consiste a faire la première face du cube

- ▶ *Fonction qui donne la position d'un petit cube dans un rubick's cube donné à partir de ses couleurs

α)Phase 1-a

Elle consiste a placer côte à côte les deux premiers carrés

- ▶ *Fonction qui place le 1er cube voulu à l'endroit voulu:
- ▶ *Nouvelle fonction de cycle:

► *Fonction qui oriente le cube précédent:

► *Fonction qui execute la phase 1-a:

▼ ***β)Phase 1-b***

Elle consiste a placer côte à côte le troisième carré à coté des deux premiers carrés

► *Fonction qui execute la phase 1-b:

▼ ***γ)Phase 1-c***

Elle consiste a placer côte à côte le troisième carré à coté des deux premiers carrés

► *Fonction qui execute la phase 1-c:

► *Fonction qui execute la phase 1:

▼ **C)Phase 2**

Elle consiste à bien placé les autres petits cubes.

► *Fonction annexe de test d'orientation:

► *Fonction annexe de test de position:

► *Fonction annexe de test de position 2:

► *Fonction annexe de test de position 3:

► *Définition des mouvements clef:

► *Bijections utilisées dans l'étape 2:

► *Alignement après l'étape 2:

► *Fonction qui fait l'étape 2:

▼ **C)Phase 3**

► *Mouvement clef de l'étape 3:

► *Décalage préliminaire pour calculer le cycle:

► *Nouvelle fonction de cycle spécifique à l'étape 3:

► *Fonction de scan:

► *Fonction annexes d'éliminations:

► Fonction annexes d'éliminations:

► Fonction annexes d'éliminations

▼ **D) Résolution**

► *Fonction de résolution d'un rubick's cube:

ResoudreII(CreerRubixAleatoire());

Ce Rubick's cube a été résolue en 64 coups

Représentation 3D du cube
 $n = 0$

