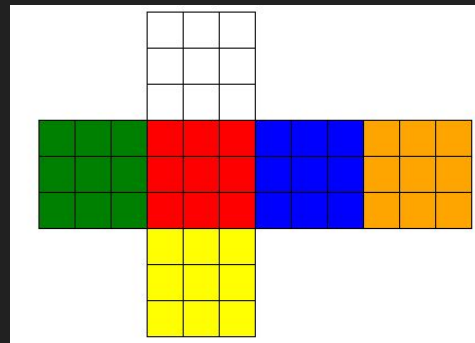


# Polycube

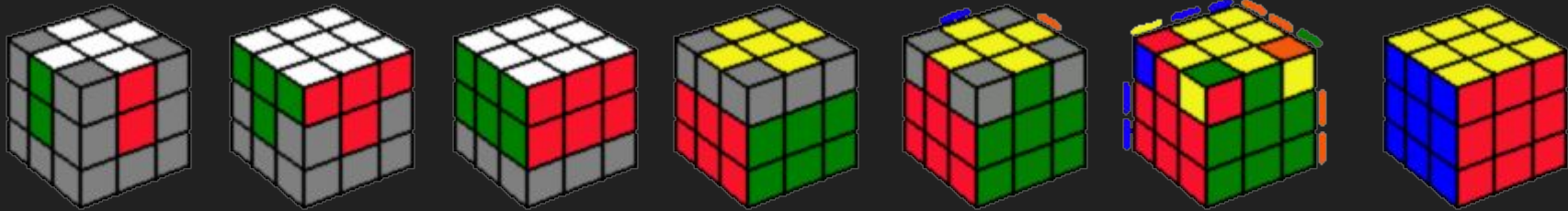
Valentin BOUCHEVREAU  
Kévin BRIAND  
Alexandre DORMEVAL  
Nicolas GAILLARD  
Pierre PETILLON  
Hugo PIGEON

# Encodage du cube

- Test des différents modèles
- Choix de la solution d'encodage : liste de 6 matrices 3\*3
- Orientation du cube constante :
  - Up : face blanche
  - Front : face rouge
  - Left : face verte
  - Right : face bleu
  - Back : face orange
  - Down : face jaune
- Mouvements : création d'une liste temporaire et suite d'affectations



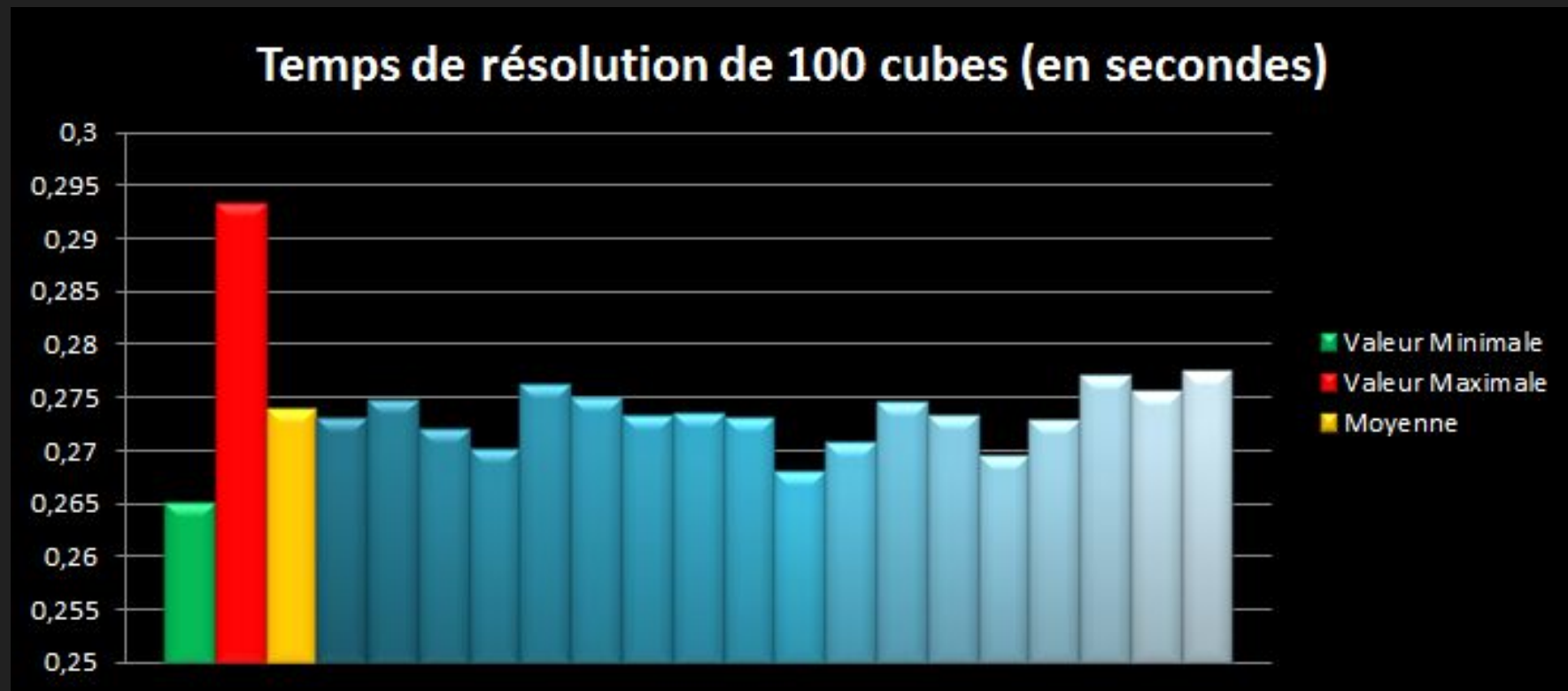
# Méthode de résolution



## Méthode de résolution basique

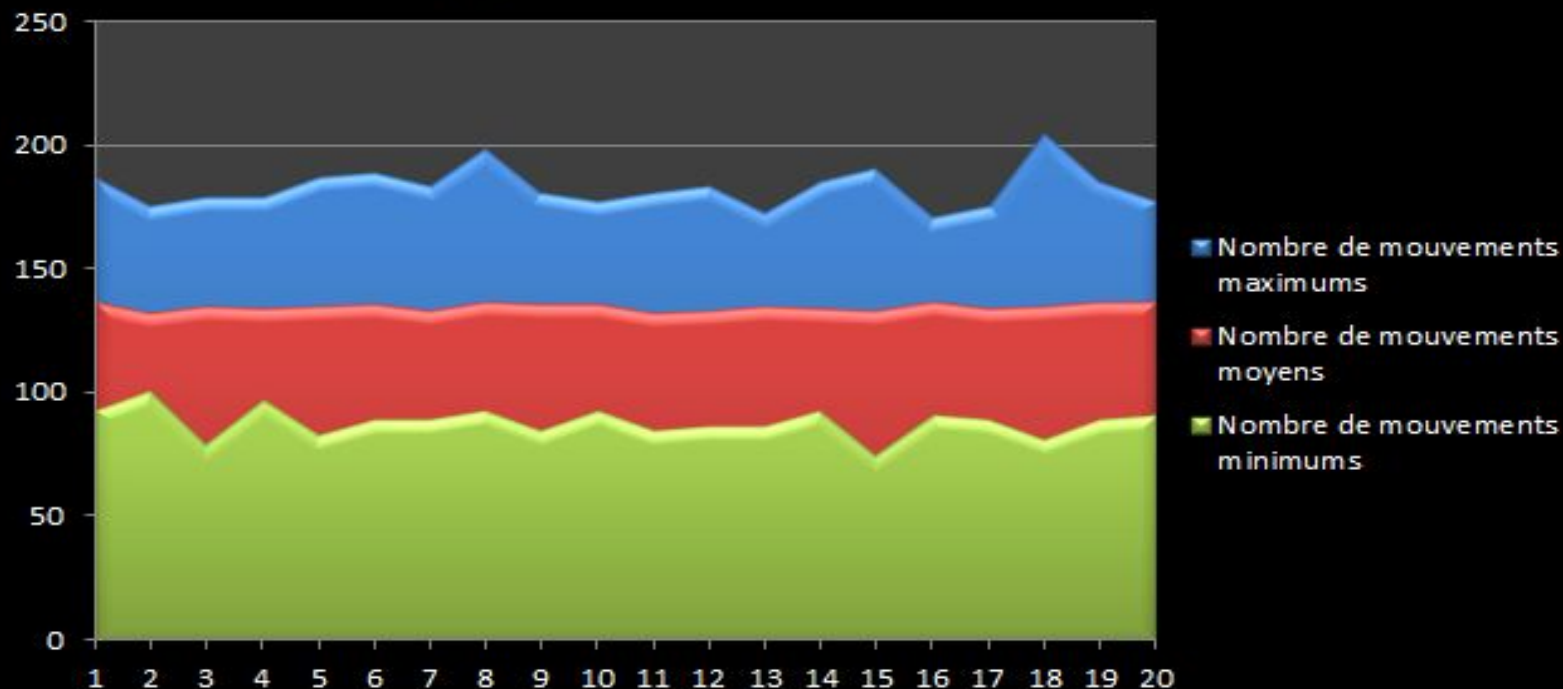
- + Facile à implémenter
- + Rapide d'exécution
- Résolution longue en terme de coups

# Performance



# Performance

**Nombre de mouvements minimums, maximums  
et moyens de résolution sur 100 cubes**

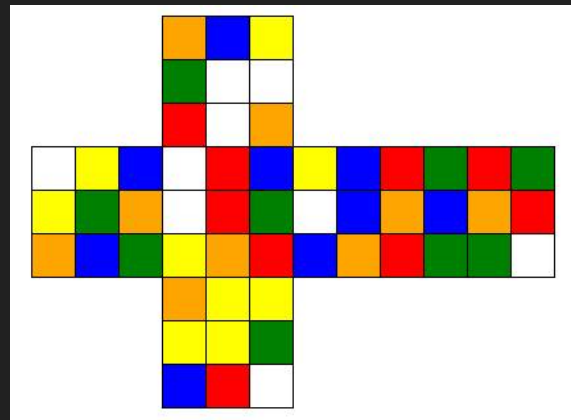


# Performance

- Longueur des solutions pour 100 000 cubes
  - Moyenne = 135,3
  - Meilleure = 57
  - Pire = 221
- Temps de calcul pour 100 000 cubes
  - Moyen = 154,8 s (soit 0,0015 s par cube)
- Occupation mémoire
  - 12 - 15 Mo (avec Python)
- Taille des fichiers
  - Nombre de fichiers = 12
  - Espace total = 103,7 ko

# Fonctionnalités supplémentaires

- Affichage du cube avec tkinter
- Générateur aléatoire de cubes
- Visualisation de la résolution sur [alg.cubing.net](http://alg.cubing.net)
- Menu textuel dans le terminal



```
polycube
-----
- 1 - Résolution d'un cube aléatoire
- 2 - Résolution d'un cube précis
- 3 - Afficher un cube précis
- 4 - Résolution via alg.cubing.net
- 5 - README
- 6 - PERFORMANCES
- 7 - Résolution 100 cubes et performances
- 0 - Quitter
-----
Que faire ? 
```

# Démonstration



# Problèmes rencontrés

- Apprendre à faire un Rubik's cube
- Appréhender Git
- Quelle représentation pour le cube (données)
- Minimiser le code et le nombre de mouvements
- S'organiser et définir les tâches pour pouvoir travailler en équipe