

Alg-trainer brainstorming

Ce projet a été prévu il y a des mois, et je l'ai tjrs pas commencé, j'ai vrm tout plein de projets coding.

Ca t'aiderait à t'entraîner pour les algorithmes de OLL/PLL, et même F2L. (Je crois y avoir réfléchi il y a longtemps)

Grâce à kociemba, je générerais un algorithme qui me mènera à un des ces cas, la formule est dans l'historique de chatgpt 2025, et d'ailleurs il ferait pas le même algorithme à chaque fois, il faut que ca soit aléatoire, mais que cela mène à ca, exactement comme jperm.net

Ce serait bien aussi qu'on tire profit du rubik's cube connecté pour ca, j'ai essayé sur mac, mais pas sur windows,

Je me souviens encore l'époque où je testais le plus de projets de smart cubes pour ça, justement j'en ai trouvé aucun marchant avec python.

D'ailleurs non seulement ca t'entraînerait, mais ca enregistrerais vrm les statistiques pour chaque cas, ou chaque algorithme:

- on peut voir ao5, ao12, ... pour chaque cas (ce que j'entends par cas, c p.ex. J Perm, N perm, ...)
- Bien sûr, on donnerait le choix à l'utilisateur pour quel algorithme faire avec le cas
- On peut voir ao5, ao12 pour chaque algorithme d'un cas, pour t'aider à choisir le cas
- Si c connecté avec le rubik's cube connecté, le timer s'enclencherait et s'arrêterait automatiquement, et aussi on pourrait entrer nos propres algorithmes, mais ca c plus complexe, prob pas nécessaire pour l'instant
- Bien sûr, ca n'aurait pas de sens, de combiner les stats de tous les algorithmes pour un cas

Jsp si on va faire que ca marche aussi pour F2L

D'ailleurs ce serait aussi bien pour mémoriser, que pour s'entraîner avec fingertricks. (D'ailleurs les fingertricks peuvent nous aider à mémoriser également)

Nouvelle idée pour plus tard: je sais qu'on peut savoir les probabilités pour chaque cas de PLL, dcp on pourrait calculer l'espérance !

Voilà comment serait stocké les algorithmes: je sais pas si ca sera un fichier pour tous les PLL, ou un fichier par PLL, probablement un .csv, avec le nom du PLL, l'algorithme

Hiérarchie: étape de CFOP (F2L/OLL/PLL) > cas (j perm/oll 23/...) > algorithme

Ca sera comme rubik-stats, on pourrait générer des graphes pour:

- Pour une métrique (ao5, ao12, ...), l'évolution de tous les cas d'une étape
- Pour un cas (avec un seul algorithme!!), l'évolution d'une métrique

En réalité, cette fois, on pourrait combiner les 2, avoir ao5, ao12 pour 2 cas p.ex. mais je sens ca va être galère à voir, dcp on va pas rentrer dedans et bien garder 2 sous-classes: Metric, et Case

D'ailleurs nouvelle idée: on pourrait très bien mélanger les stats de 2 algs d'une même case, pour voir vrm à quel point ca a changé, et ... on peut très bien mélanger tous les algs pour tous les cas, mais d'une seule étape hein, on va pas s'amuser à mélanger depuis toutes les étapes, bien sûr qu'on peut très bien avoir ao5, ao12 de toute une étape, mais ... ca sera pas très réel, en fait

ce serait bien de faire ça mais avec les vraies probabilités.

Genre faudrait que qd on s'entraîne, on puisse choisir de faire avec les vraies stats ou non, et puis aussi si on passe notre temps à entraîner un seul alg, là il y aurait pas de sens d'inclure cela dans les statistiques de l'étape.

Mtn il reste encore une problématique, où va t on stocker tout ça, et comment surtout, il faut stocker:

- les algorithmes pour chaque cas pour chaque étape
- Les statistiques pour chaque alg, peut-être pas tout, mais au moins genre les 12 derniers, quoi que ... jsp trop, ce serait bien qd même d'en avoir plein comme ça on peut voir l'évolution

Dans les statistiques il faut aussi:

- la date
- Également le status? Jsp trop
- Aussi si c généré avec les bonnes probabilités, pour avoir l'évolution et les stats d'une étape, mais ça vrm ... te casse pas la tête, ça sera pour plus tard, et de tt façon on peut déjà voir cela avec les stats multi-phases.

Dcp comment on stockerait tout ça, je pense je peux pas faire un fichier par étape, genre même pas un fichier par cas, il me faudrait un fichier par alg, surtout pour les statistiques.

Aussi je me suis dit, il faut séparer nos propres statistiques de la librairie d'algorithmes ! Comme ça les autres peuvent aussi faire ça

Ca me rappelle l'idée de stocker des données dans le github repo, comme ça qd on PR, ça stocke dans notre compte github. En tout cas, on pourrait choisir nous-même où on veut stocker. Et library serait par contre dans le github repo bien sûr, en plus comme ça des autres pourraient le modifier pour s'entraîner.

Et dans les stats, il faut aussi mettre lequel l'utilisateur a select