

CODING CLUB

EPITECH



Bsq

Objectifs

1. **Comprendre le problème** : Vous devez lire une maps depuis un fichier, trouver le plus grand carré possible sans obstacles, et marquer ce carré sur la maps.
2. **Implémenter l'algorithme** : Utilisez des structures de données appropriées et des algorithmes efficaces pour trouver le plus grand carré.
3. **Tester votre solution** : Assurez-vous que votre solution fonctionne correctement avec différents jeux de données.

Structure du Projet

Le projet est structuré comme suit :

- `main.py` : Fichier principal pour lancer l'algorithme.
- `utils.py` : Fichier contenant des fonctions utilitaires.
- `/maps` : Dossier contenant différentes maps.

2 binaires sont mis à votre disposition pour tester votre solution :

- `./tester`
- `./solver`

Lancer le projet

Pour lancer le projet, rien de plus simple :

```
python main.py <maps>
```

Pour utiliser les binaires, vous pouvez les lancer de la manière suivante :

```
./solver <maps>
```

```
./tester <maps> <main.py> [solver]
```

Comment tester ?

Il semblerait qu'aucune maps ne soit fournie pour tester votre programme !

Heureusement, un générateur vous est proposée afin de vous lancer dans le projet !

Pour lancer le générateur, rien de plus simple :

```
./generator <width> <height>
```

N'oubliez pas de tester des cas particuliers, comme des maps de 1 par 10, 10 par 1, une map vide, avec une seule case, et plein d'autres possibilités.

Bonus ?!?!

Déjà fini ? Vous pouvez essayer d'implémenter des fonctionnalités supplémentaires pour améliorer votre projet :

- Compléter le fichier `generator.py` afin de reproduire le comportement du binaire de génération
- Réécrire le fichier `./tester` en Python. (il a été écrit en python avant d'être compilé)
- Ajouter des options pour personnaliser le comportement de votre programme (e.g. `--verbose`, `--color`, `--output`, etc.)

```

      _____      .____ .____
 /   _____/  _____  _|_/_/_|_  |   _ _   _   |   |   |
/    \_   _/    \_   _/    \_   _/    |   |   |   |   |   |   |
\      \_ \    (  <_> |  <_> ) /_/_|   |   |   |   |   |   |
 \_____  \_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_
      \V          \V          \V          \V          \V          \V\

```