Pavage de grille LEGO

SOMMAIRE :

Organisation du Projet

```
.
├── code
│   ├── brique.c
│   ├── dependance
│   │   ├── brique.h
│   │   ├── image.h
│   │   ├── main.h
│   │   ├── matching.h
│   │   ├── solution_1x1.h
│   │   ├── solution_2x2.h
│   │   ├── solution_forme_arbitraire_rentable.h
│   │   ├── solution_greedy_1x2.h
│   │   ├── solution_greedy_1x2_stock.h
│   │   ├── solution.h
│   │   ├── solution_rectfusion.h
│   │   ├── structure.h
│   │   └── util.h
│   ├── exec
│   ├── image.c
│   ├── main.c
│   ├── matching.c
│   ├── solution_1x1.c
│   ├── solution_2x2.c
│   ├── solution.c
│   ├── solution_forme_arbitraire_rentable.c
│   ├── solution_greedy_1x2.c
│   ├── solution_greedy_1x2_stock.c
│   ├── solution_rectfusion.c
│   └── util.c
```

```
├── input
│   ├── briques.txt
│   ├── image2.txt
│   ├── image3.txt
│   ├── image4.txt
│   └── image.txt
├── Makefile
└── output
```

Notice d’utilisation du projet

Ce projet utilise un Makefile pour compiler, déboguer, exécuter et nettoyer l'application de pavage basée sur différents algorithmes.
Les commandes suivantes permettent de compiler le code, produire l'exécutable pavage, lancer le programme avec un jeu de données, et nettoyer les fichiers intermédiaires.

1. **Compilation**

Pour compiler l'ensemble du projet, utiliser la commande :
make
ou :
make all

Cela compile tous les fichiers .c du dossier "code" et génère les fichiers objets dans "code/exec", puis crée l'exécutable "code/exec/pavage".

2. **Exécution**

Pour exécuter le programme avec les paramètres par défaut :
make run

Le programme s'exécute alors avec les fichiers du dossier "input" et traite tous les algorithmes.

Pour choisir un algorithme précis, utiliser la variable ARGO. Exemples :
make run ARGO=1x1
make run ARGO=2x2
make run ARGO=greedy
make run ARGO=greedy_stock
make run ARGO=rectfusion
make run ARGO=rentable
make run ARGO=all

On a la possibilité d'exécuter plusieurs algorithmes en une commande grâce à la variable ARGO. Exemples :

make run ARGO="1x1 greedy"
make run ARGO="2x2 1x1"

Les résultats sont enregistrés dans le dossier "output" sous les fichiers :
pavage_1x1.txt
pavage_greedy_1x2.txt
pavage_greedy_1x2_stock.txt
pavage_2x2.txt
pavage_rectfusion.txt
pavage_forme_arbitraire_rentable.txt

3. Débogage

Pour lancer le programme sous gdb :
make debug

4. Nettoyage

Pour supprimer l'exécutable, les fichiers objets et les fichiers de sortie générés :
make clean

5. Organisation du projet

code/ : sources .c et .h
code/exec/ : objets .o et exécutable
input/ : données d'entrée
output/ : fichiers de résultats