SAE Pavage

Ce logiciel permet de transformer une image (matrice de pixels) en un plan de montage LEGO en utilisant différentes stratégies d'optimisation.

Fonctionnalités

Le programme propose 4 modes de pavage accessibles via des options en ligne de commande :

- **`stock`** : Priorise l'utilisation des pièces disponibles en stock.
- **`libre`** : Utilise toutes les formes du catalogue pour un pavage géométrique optimal.
- **`minimisation`** : Minimise l'erreur de couleur pour une qualité visuelle maximale.
- **`rentable`** : Optimise le prix total en favorisant les briques au meilleur rapport surface/prix.

Compilation

Le projet est structuré pour être compilé facilement via un **`Makefile`**.

make

Cela génère l'exécutable **`pavage`** dans le dossier **`C/code/exec/.`**

Exécution

Pour exécuter tout les pavages après compilation :

make run

Pour exécuter un pavage spécifique après compilation :

make run ARGO=libre
make run ARGO=stock
make run ARGO=minimisation
make run ARGO=rentabilite

Suppression

Pour nettoyer après avoir testé :

make clean

Rapport de Synthèse

Travail d'équipe et Algorithmes Chaque membre du groupe a développé un algorithme apportant une solution spécifique au problème de pavage :

1. **Stratégie de Minimisation de l'Erreur (Qualité Maximale)** : Cet algorithme privilégie la fidélité visuelle. Il ne remplace les briques de base (1x1) par des briques plus grandes que si la qualité des couleurs est strictement maintenue ou améliorée. Il ignore les contraintes de stock pour garantir l'image la plus nette possible.

2. **Stratégie de Gestion de Stock (Priorité Disponibilité)** : Cette approche est conçue pour respecter strictement l'inventaire fourni. L'algorithme vérifie en temps réel les quantités restantes dans le catalogue et adapte ses choix de briques pour éviter les ruptures de stock, quitte à utiliser des couleurs légèrement moins optimales.

3. **Stratégie de Rentabilité (Optimisation du Coût)** : L'objectif ici est de réduire la facture totale. L'algorithme calcule si l'utilisation d'une grande brique est financièrement plus avantageuse que l'assemblage de plusieurs petites briques pour une même zone, en acceptant une légère dégradation de la précision des couleurs.

4. **Stratégie à Formes Libres (Polyvalence Géométrique)** : Cet algorithme exploite l'intégralité du catalogue de pièces, y compris les formes complexes ou avec trous. Il utilise une méthode gloutonne pour placer les plus grandes formes disponibles sur des zones de couleurs homogènes, optimisant ainsi le nombre total de pièces utilisées.