

Rapport de Projet – Développement d'une Application de Résolution de Puzzle

1. Introduction

Ce projet avait pour objectif le développement d'une application permettant la résolution automatique de puzzles à partir d'images numériques. Il s'agissait d'un défi à la fois algorithmique, graphique et organisationnel, mené par une équipe de cinq étudiants : Roman, Clément, Florian, Tom et Nassym.

2. Organisation de l'équipe et processus de développement

2.1 Répartition des rôles

L'équipe a d'abord adopté une approche collaborative sur l'ensemble du code et de l'algorithme, avant d'évoluer vers une répartition plus spécialisée des tâches pour optimiser l'efficacité :

- Nassym et Florian ont finalisé le cœur du code, notamment l'algorithme de résolution.
- Clément s'est chargé de l'implémentation de l'interface graphique via JavaFX, garantissant une interaction fluide avec l'utilisateur.
- Roman a rédigé l'ensemble du rapport de projet.
- Tom a produit les diagrammes techniques et de fonctionnement.
- Florian a joué un rôle central en coordonnant les travaux, en distribuant les tâches et en assurant le lien avec le professeur.

2.2 Communication et suivi

La communication interne a été assurée via WhatsApp, Discord et Microsoft Teams, permettant des échanges continus malgré les différences de disponibilité. Des points quotidiens étaient organisés tous les midis afin de faire le point sur l'avancement, résoudre les blocages et ajuster la charge de travail.

2.3 Gestion de projet

Le projet a été versionné via Git sur GitHub, avec une gestion rigoureuse des branches et des conflits. Cette approche a permis un travail simultané fluide et une traçabilité de l'évolution du projet.

3. Défis techniques et solutions

3.1 Gestion des bordures blanches

L'un des principaux défis visuels rencontrés fut la présence persistante de bordures blanches entre certaines pièces assemblées. Plusieurs tentatives ont été entreprises pour corriger ce défaut (notamment par post-traitement ou recalcul des marges), sans succès complet.

Impact : Une dégradation visuelle du rendu final, bien que cela n'affecte pas la validité de l'assemblage.

3.2 Gestion de la rotation

Pour certains puzzles, notamment ceux de petite taille, la prise en compte de la rotation des pièces s'est avérée cruciale. L'équipe a optimisé l'algorithme en intégrant un système de rotation permettant de tester automatiquement les quatre orientations possibles.

Résultat : Une solution efficace pour les puzzles simples, bien que certaines configurations complexes ne puissent toujours pas être résolues automatiquement.

3.3 Performance avec les puzzles de grande taille

L'approche algorithmique basée sur la récursivité s'est révélée inefficace pour les puzzles de grande dimension (>4000 pièces), en raison d'une profondeur d'appel excessive et de temps de calcul trop longs.

Compromis technique : L'algorithme a été optimisé autant que possible, mais une limite d'échelle a été acceptée pour garantir la stabilité du système.

4. Fonctionnalités et résultats

4.1 Algorithme de résolution

L'algorithme développé repose sur une gestion fine des signatures des côtés de chaque pièce, avec une comparaison RGB basée sur des pixel scores. Ce mécanisme a permis d'établir des correspondances robustes entre les pièces.

Exemple : Chaque bord est analysé via un échantillon de pixels ; un score de similarité est attribué selon les écarts RGB, assurant une fiabilité accrue dans les correspondances.

4.2 Interface graphique (JavaFX)

Grâce à JavaFX, l'application propose une interface simple et intuitive permettant à l'utilisateur de :

- Charger une image source
- Visualiser les pièces générées
- Observer la reconstitution automatique du puzzle

4.3 Système de rotation

Un module de rotation intégré permet d'adapter l'orientation des pièces, essentiel pour certains puzzles non uniformes.

4.4 Résultats globaux

- Algorithme de base robuste et performant pour les puzzles jusqu'à une certaine complexité
- Gestion efficace des correspondances entre pièces
- Interface utilisateur fonctionnelle et intuitive
- Satisfaction élevée de l'équipe sur la qualité de l'algorithme

5. Bilan et perspectives

5.1 Points forts du projet

- Organisation agile et adaptative
- Excellente coordination via des outils collaboratifs
- Solide implémentation algorithmique
- Travail d'équipe homogène et complémentaire

5.2 Limites et axes d'amélioration

- Absence de gestion optimisée pour les très grands puzzles
- Bordures blanches à corriger pour une meilleure expérience utilisateur
- Amélioration possible de l'interface graphique pour plus d'interactivité (ex. zoom, drag & drop)

6. Conclusion

Le projet a atteint la majorité de ses objectifs initiaux, en particulier en ce qui concerne le développement d'un algorithme de résolution fonctionnel et l'implémentation d'une interface graphique intuitive. Malgré certaines limites techniques liées à la performance et à l'affichage, l'équipe a su relever les défis de manière efficace, en maintenant un niveau de coordination et de communication exemplaire. Ce projet constitue une base solide pour d'éventuelles évolutions vers une version plus complète et optimisée de l'application.

Diagramme de classes :

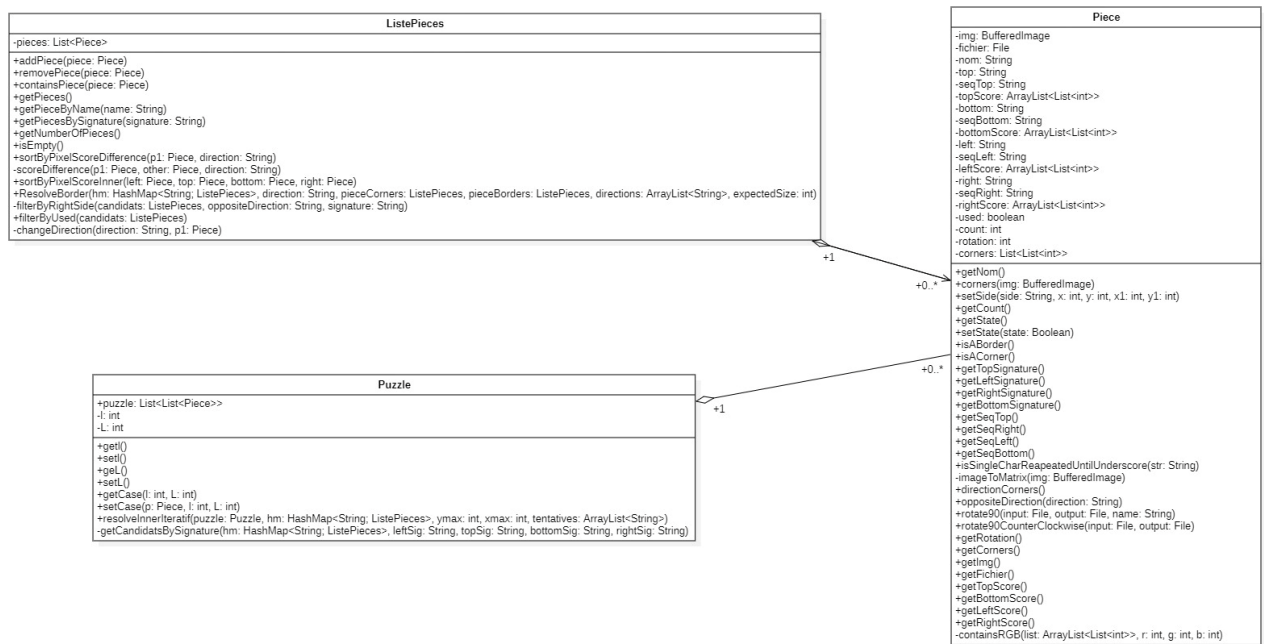


Diagramme des cas d'utilisations :

