

# Projet Tableau LEGO

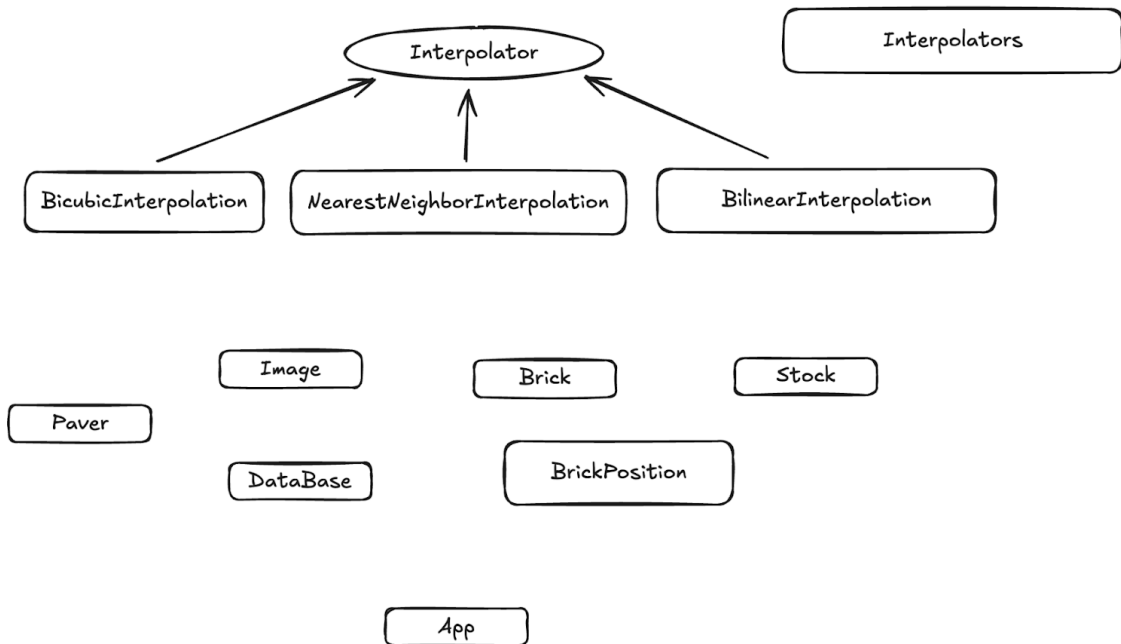
## Module Java

### Documentation



<b>Architecture du projet.....</b>	<b>2</b>
<b>Les choix d'implémentation.....</b>	<b>2</b>
<b>Difficultés rencontrées et bugs présents.....</b>	<b>3</b>
Difficultés.....	3
Bugs et Limitations.....	3

## Architecture du projet



L'approche de modularité du projet repose sur les Interpolators, en utilisant une interface. D'autres classes (telles que Image, Paver, DataBase et Stock) pourraient potentiellement être remplacées par des interfaces, mais cela n'a pas été réalisé, faute de temps.

La classe principale du module est App. C'est elle qui instancie tous les autres objets et gère la vérification des éléments passés en arguments.

Au niveau de la gestion des Exceptions, peu de blocs try-catch ont été utilisés, car en cas d'erreur, le reste du programme ne peut pas continuer son exécution.

## Les choix d'implémentation

Pour des raisons de temps, seuls les trois premiers algorithmes de réduction de dimension de l'image ont été implémentés :

- Bicubique
- Plus proches voisins
- Bilinéaire

Pour l'appel du programme C, un ProcessBuilder a été utilisé à la place de l'API FFM.

Concernant la gestion du stock, celle-ci est restée basique et s'appuie sur la base de données, conformément à l'indication selon laquelle ce n'était pas obligatoire pour les alternants. La Factory n'a pas été utilisée.

## **Difficultés rencontrées et bugs présents**

### **Difficultés**

Algorithmes d'interpolation : La principale difficulté rencontrée au début concernait la gestion des pixels avec des coordonnées non entières dans les algorithmes d'interpolation.

Appel au programme C : Un problème a été rencontré lors de l'appel du programme C. Il s'avère que ce problème provenait également du programme C lui-même, dû à un changement d'architecture de l'ordinateur personnel et du conteneur Docker.

### **Bugs et Limitations**

Décalage lors du redimensionnement : Un essai a été fait pour éviter la perte de pixels lors du changement de résolution en effectuant le redimensionnement en plusieurs étapes (division par 2 à chaque fois). Cependant, cette approche provoquait un décalage sur l'image compressée. Le résultat obtenu en un seul passage semblait tout de même convenable, et par manque de temps, il a été décidé de passer à une autre fonctionnalité.