

# RAPPORT SAE32\_2025

## BUT Informatique - 2ème année

### SAE3.2-Application "Primitive Image Format"

Youness BOULALAM Algassimou DIALLO Ayoub ANHDIRE

07 janvier 2026

Développement d'une application de conversion d'une image au format PIF (1ère image) + Affichage dans une fenetre d'une image contenue dans un fichier PIF (2ème image).

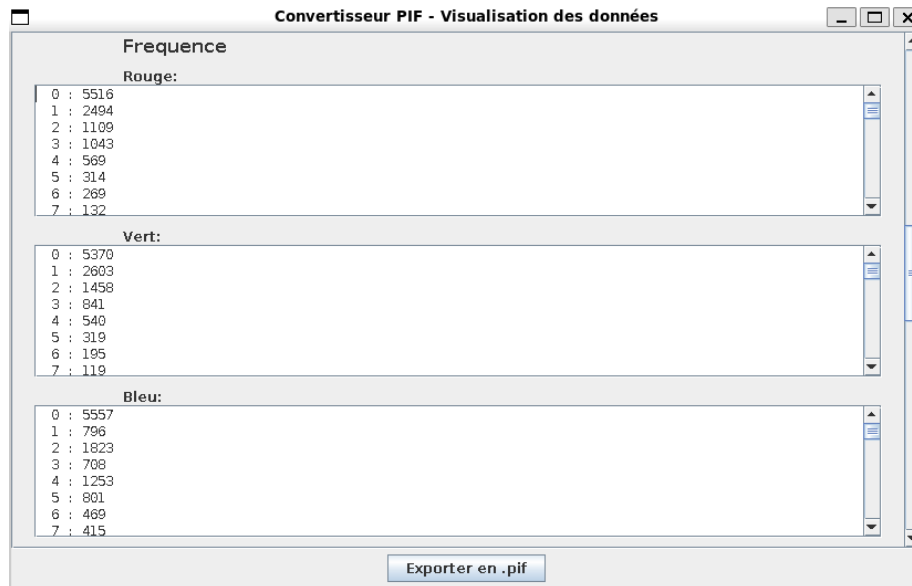


Figure 1: Interface principale du convertisseur , notamment avec les tables de fréquences

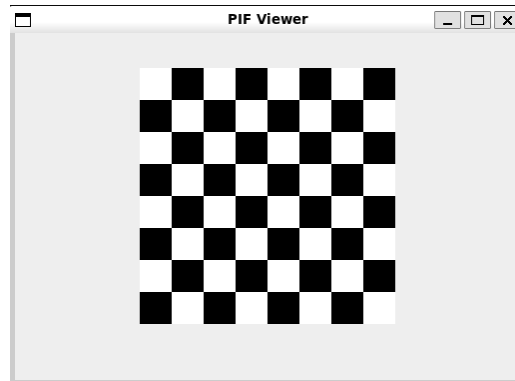


Figure 2: Interface principale du visualisateur ou l'utilisateur peut déplacer l'image avec la souris lorsqu'elle est trop grande

## Comptes GIT utilisés pour ce projet

- Youness BOULALAM (Groupe 4) : youness
- Algassimou DIALLO (Groupe 4) : Diallo VM fbleau
- Ayoub ANHDIRE (Groupe 4) : Ayoub ANHDIRE Anhdire Ayoub ANHDIRE

Réalisé en JAVA : Architecture MVC

Dépôt Gitea : [https://grond.iut-fbleau.fr/dialloa/SAE32\\_2025](https://grond.iut-fbleau.fr/dialloa/SAE32_2025) Professeur : M.Luc Hernandez

Date de rendu : 11 janvier 2026

## Sommaire

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Répartition des taches</b>	<b>5</b>
<b>3</b>	<b>Fonctionnalités principales</b>	<b>6</b>
3.1	Conversion au format PIF . . . . .	6
<b>4</b>	<b>Fonctionnalités principales</b>	<b>6</b>
4.1	Conversion au format PIF . . . . .	6
4.2	contribution d'Algassimou Pella Diallo . . . . .	6
4.3	Visualisateur au format PIF . . . . .	6
<b>5</b>	<b>MakeFile du Projet</b>	<b>6</b>

<b>6 Conclusion</b>	<b>6</b>
6.1 Youness BOULALAM . . . . .	6
6.2 Algassimou DIALLO . . . . .	6
6.3 Ayoub ANHDIRE . . . . .	6

## 1 Introduction

Pour cette deuxième SAE du semestre 3, il nous a fallu réaliser **deux** programmes : un convertisseur d'une image au format png ou du moins supportable par la méthode *read* de la classe **ImageIO**. L'image sera donnée en argument ou sinon elle sera sélectionnée par un **JFileChooser** (à part s'il y a un deuxième argument et qu'il représente un chemin et un nom acceptable). Pour ce programme, on aura une interface simple avec la table des fréquences, la table des codes initiaux (Huffman) et la table des codes canoniques. À la suite de cela, l'utilisateur a la possibilité ou non de pouvoir sauvegarder son image au format PIF.

Le deuxième programme est un visualisateur ou celui-ci est censé afficher dans une fenêtre une image contenue dans un fichier au format PIF. Celui-ci est donnée en argument ou la sélection du fichier sera réalisée à l'aide d'un **JFileChooser**. La taille de la fenêtre sera la taille de l'image à afficher avec une taille qui ne dépasse pas l'écran : si l'image est plus petite que la fenêtre, elle sera centrée et si elle est plus grande que la fenêtre : elle sera partiellement visible et pourra être déplacée à la souris en maintenant le bouton gauche enfoncé.

Ce projet a été réalisé en trinôme, et nous avons développé l'application entièrement en Java, en respectant les bonnes pratiques vues en cours : architecture MVC, création de makefile, et une interface graphique codée sans bibliothèque externe. Un soin particulier a été apporté à l'ergonomie : l'interface est pensée pour être simple, claire, et agréable à utiliser.



## 2 Répartition des taches

Nom	Tâches effectuées
Algassimou DIALLO	<ul style="list-style-type: none"> <li>• Creation de la structure generale du projet (Dossier, UML)</li> <li>• Implémentation de BitOutputStream</li> <li>• Implémentation de BitInputStraam</li> <li>• Implémentation complète du contrôleur (ConverterController)</li> <li>• Génération et affichage des fréquences</li> <li>• J'ai aussi travailler sur le constructeur de HuffmanTree et Huffman node</li> <li>• Intégration de l'interface graphique (ConverterWindow)</li> <li>• Gestion du bouton Export et sauvegarde .pif dans un thread séparé</li> <li>• Creation et Test du MakeFile</li> <li>• Implémentation du PIFWriter (écriture des tables, pixels)</li> <li>• Implémentation du PIFReader (lecture, reconstruction, décodage)</li> <li>• Implémentation du contrôleur pour la vue</li> </ul>
Youness BOULALAM	<ul style="list-style-type: none"> <li>• Gestion des erreurs et messages utilisateur</li> <li>• Conversion RGBImage BufferedImage pour le Viewer</li> <li>• Conversion RGBImage BufferedImage pour le convertisseur</li> <li>• tache 1</li> </ul>
Ayoub ANHDIRE	<ul style="list-style-type: none"> <li>• Javadoc (Équipe)</li> <li>• Génération des codes Huffman</li> <li>• Génération des codes canoniques</li> <li>• Interface d'affichage des fréquences</li> <li>• Test de BitOutputStream</li> <li>• Lecture de l'en-tête</li> <li>• Diagramme de classe</li> </ul>

### **3 Fonctionnalités principales**

#### **3.1 Conversion au format PIF**

### **4 Fonctionnalités principales**

#### **4.1 Conversion au format PIF**

La conversion d'une image vers le format PIF repose sur plusieurs étapes : analyse des composantes RGB, calcul des fréquences, construction de l'arbre de Huffman, création des codes canoniques, puis écriture finale dans un fichier binaire structuré. Cette section présente les mécanismes mis en place et la contribution de chaque membre du groupe.

#### **4.2 contribution d'Algassimou Pella Diallo**

#### **4.3 Visualisateur au format PIF**

### **5 MakeFile du Projet**

### **6 Conclusion**

Nous avons pensé que ce projet ... (à compléter)

#### **6.1 Youness BOULALAM**

#### **6.2 Algassimou DIALLO**

#### **6.3 Ayoub ANHDIRE**