

Java Introduction

Backlog

Projet `imagefilter`

Objectif fonctionnel

**Appliquer massivement une pipeline de
filtres sur des images**

Example

```
./imagefilter --filters "blur:3|grayscale" -i imgs -o output/
```

- Sur chaque image du dossier `imgs/`
- J'applique un flou d'une taille 3
- J'applique un niveau de gris
- J'enregistre l'image résultante dans `output/`

Démo

Versioning

Tout votre code doit être sur git

(La notation tient compte de la régularité de vos commits)

Stories

Story #01

EN TANT QUE développeur

JE VEUX convertir une image RGB en noir & blanc

AFIN DE avoir mon 1er filtre d'image

DoD L'image en N&B est enregistrée dans le
filesystem



- Utilisez la lib <https://github.com/bytedeco/javacv>
- Demandez moi pour les packages à importer ! (C'est pas simple du tout...)

Story #02

EN TANT QUE développeur

JE VEUX appliquer un flou à une image

AFIN DE avoir mon filtre d'image de flou

DoD L'image floutée est enregistrée dans le
filesystem

Conseil

- Utilisez un `GaussianBlur`
- La taille du flou doit toujours être impaire (ex: 3, 5, 7, 9, etc.)
- Commencez à réfléchir à une hiérarchie de classe pour vos filtres

Story #03

EN TANT QUE développeur

JE VEUX dilater une image

AFIN DE avoir mon filtre d'image de dilatation

DoD L'image dilatée est enregistrée dans le
filesystem



- Exemple de dilatation

Story #04

EN TANT QUE développeur

JE VEUX Centraliser les erreurs de mes filtres

AFIN DE simplifier mon dev

DoD Utiliser une checked exception custom utilisée
dans tous les filtres

Conseil

- Le contrat fort d'une hiérarchie de classe peut être utile ici
- Une exception possible : est-ce que la taille du flou est impaire ?

Story #05

EN TANT QUE développeur

JE VEUX Appliquer une pipeline de filtres sur un ensemble d'images

AFIN DE de traiter l'intégralité d'un dossier

DoD Un dossier `output/` contenant les images traitées

Story #06

EN TANT QUE développeur

JE VEUX proposer une CLI à mon app

AFIN DE rendre l'app paramétrable

DoD lancer l'app avec différentes sources /
destinations

Résultat attendu

```
$ ./image-filter -h
usage: imagefilter
  -h,----help
  -i,--input-dir <directory>
  -o,--output-dir <directory>
```

Conseil

- Découpez cette story en tâches => 1 tâche = 1 option du CLI
- Qui a dit le mot lib !?

Story #07

EN TANT QUE développeur

JE VEUX configurer ma pipeline depuis la CLI

AFIN DE faire varier mes filtres au runtime

DoD option `--filters "blur:3|dilate:5|grayscale"`

Story #08

EN TANT QUE développeur
JE VEUX logger les opérations
AFIN DE avoir de la traçabilité

DoD afficher chaque opération en console + en
fichier de log (+ option `--config-file image.log`)

Conseil

- Séparez les responsabilités !
 - Rappelez vous, 1 classe = 1 responsabilité
- Un filtre pourrait fournir une description pour son log (évite de dire à un filtre de devoir logger, ce n'est pas sa responsabilité)

Story #09

EN TANT QUE développeur

JE VEUX proposer un fichier de configuration

AFIN DE simplifier le partage de ma pipeline

DoD un fichier .ini contenant les options CLI

Conseil

- Il faudra ajouter une option CLI `--config-file image.ini`
- Si le fichier est présent, il remplace entièrement les options CLI
- Le fichier de conf pouvoir doit contenir :
 - `inputDir=...`
 - `outputDir=...`
 - `logFile=...`
 - `filters=blur:3|grayscale`

Story #10

EN TANT QUE développeur

JE VEUX fournir une JavaDoc

AFIN DE documenter le projet pour d'autres dev

DoD un site JavaDoc généré à partir du code du projet

Conseil

- Cette documentation doit être versionnée dans le repo git
- Toutes les classes doivent être documentées
- Toutes les fonctions `public` doivent être documentées
- BONUS : documentez les packages pour expliquer leur rôle

Story #11 (BONUS)

EN TANT QUE développeur

JE VEUX mixer le fichier de conf et la CLI

AFIN DE rendre mon programme plus flexible

DoD avec un fichier de conf, toute option CLI
présente va **override** la conf



- Plus facile à faire si une classe `Conf` est commune à la CLI et à la conf INI
- Si votre `Conf` est immutable, le merge est plus naturel

Story #12 (BONUS)

EN TANT QUE développeur

JE VEUX afficher les filtres disponibles

AFIN DE améliorer la discoverability de mon app

DoD ajouter une option `--list-filters`

Conseil

- La version du pauvre => une liste en dur
- La version bogoss => on regarde dynamiquement ceux qui hérient de `Filter`
 - <https://github.com/ronmamo/reflections>

Story #13 (BONUS)

EN TANT QUE développeur

JE VEUX afficher les membres de l'équipe dans
l'image

AFIN DE être fier de mon projet !

DoD ajouter un filtre `FilterZeTeam` qui écrit la liste
des membres dans l'image

Conseil

- `opencv_imgproc.putText()` 