

VV - Code coverage

Antoine Leval, Erwan Iquel

February 22, 2019

Abstract

Le projet d'IDM consistait à développer un outil pour générer des vidéo selon une grammaire prédéfinie. Le code source du projet est téléchargeable à l'adresse suivante : https://github.com/Viinyard/IDM_PROJECT.

1 Développement des spécifications

Toutes les spécifications ont été développée chacune dans une classe différentes ayant une méthode main chacune, il est possible de les lancer chacune individuellement, ou d'appeler leur méthode `generate(VideoGen videogen);`

Cette méthode est utilisé pour appeler les features dans la classe de test notamment.

- CalculateAllSize.java :
`List<String> headers = new GenerateHeaderCSV().generate(videogen);`
- CalculateDuration.java :
`new CalculateDuration().generate(videogen);`
`helper.saveVideoGenerator(URI.createURI("exemple1_with_duration.videogen"), videogen);`
- CalculateMaxDuration.java :
`double maxDuration = new CalculateMaxDuration().generate(new CalculateDuration().generate(videogen));`
- CalculateNombreMedia.java :
`int nbMedia = new CalculateNombreMedia().generate(videogen);`
- CalculateNombreVariantes.java :
`int nbVariantes = new CalculateNombreVariantes().generate(videogen);`
- FFmpegGenerator.java :
`Utils.ffmpegGenerate(Utils.saveFile(new File("exemple1.videogenerated"),
new FFmpegGenerator().generate(videogen)), new File("output.mov"));`
- GenerateAllVariantes.java :
`List<Node> lVariantes = new GenerateAllVariantes().generate(videogen);`
`System.out.println(lVariantes.size());`
- GenerateCSVData.java :
`new GenerateCSVData().generate(videogen, new File("stats.csv"));`
- GenerateHeaderCSV.java :
`List<String> headers = new GenerateHeaderCSV().generate(videogen);`
- GenerateVignette.java :
`int nbVignettes = new GenerateVignette().generate(new CalculateDuration().generate(videogen)).size();`

- GenerateWebPage.java :

```
new GenerateWebPage().generate(videoGen);
Utils.saveFile(new File("index.html"), new GenerateWebPage().generate(videoGen));
```
- Generer10Alea.java :

```
Utils.ffmpegGenerate(Utils.saveFile(new File(args[0]+"erated"),
new FFmpegGenerator().generate(videoGen), new File("output.mov"));
```

2 test et classe utilitaire

- Utils.java
 - `public static File generateVignette(File f, double duration)`
 - `public static boolean ffmpegGenerate(File input, File output)`
 - `public static long getSize(File f)`
 - `public static double getDuration(File f)`
 - `public static File saveFile(File output, String content)`
- VideoGenTest1.java
 - `public void testNbVariantes()`
 - `public void testNombreLigneCSV()`
 - `public void testNombreVignette()`
 - `public void testNombreImagePageWeb()`

Pour la classe de test on utilise une classe de test paramétrée pour la lancer sur nos 10 différents fichier .videogen

```
@Parameters(name = "videogens")
public static Collection<Object[]> data()
return Arrays.asList(new Object[][]
"example1.videogen",
"example2.videogen",
"example3.videogen",
"example4.videogen",
"example5.videogen",
"example6.videogen",
"example7.videogen",
"example8.videogen",
"example9.videogen",
"example10.videogen"
);
```

3 Test sur les videogens des autres équipes

Pour les tests sur les .videogen des autres équipes nous avons généré un .jar qui appelle une classe générant une video à partir du premier argument du main, le fichier videogen.

Nous avons ensuite appelé le jar 10 fois pour chaque videogen proposé, comme ceci :

```
for i in 1..10
```

```
do
```

```
java -jar videoGen.jar /Users/VinYarD/Desktop/tests/LEBLANC_VANSTEEN/spec.videogen
done
```

Nous n'avons pas rencontré de problèmes particulier, sauf pour 3 videogens :

- Larzilliere-Forget
- LEBLANC_VANSTEEN
- LESAINTE_Arthur_LEBUHAN_Riwal

Pour LEBLANC_VANSTEEN et LESAINTE_Arthur_LEBUHAN_Riwal il a suffit de placer les video dans un dossier videos/

Pour Larzilliere-Forget nous avons dû éditer le fichier videogen car les paths des vidéos renseignées commençaient toutes par ./

Pour ce qui est des tests (5 x 10 fichiers), tous les tests passent sans problèmes, excepté 4 qui sont en faillure, les tests sur le nombre de variantes et le nombre de ligne CSV mènent à une erreur seulement pour les deux fichiers :

- example2.videogen
- example3.videogen

Le premier fichier ne comporte qu'une video optionnelles, et le second propose plusieurs video optionnelles.

Voici les erreurs générés :

[concat @ 0x7fb053000400] Unsafe file name '/v1.avi' /Users/VinYarD/Desktop/tests/Larzilliere-Forget/specs.videogenerated: Operation not permitted

[concat @ 0x7fecf9001200] Impossible to open '/Users/VinYarD/Desktop/tests/LEBLANC_VANSTEEN/videos/c /Users/VinYarD/Desktop/tests/LEBLANC_VANSTEEN/spec.videogenerated: No such file or directory

[concat @ 0x7fbdc7000400] Impossible to open '/Users/VinYarD/Desktop/tests/LESAINTE_Arthur_LEBUHAN_Riwal/2.mp4' /Users/VinYarD/Desktop/tests/LESAINTE_Arthur_LEBUHAN_Riwal/specs.videogenerated: No such file or directory