

# VideoGenerator

Noureddine KADRI, Fahim MERZOUK

M2 ILa

ISTIC, Université de Rennes 1

Février 2019

Le but de ce document est de fournir des informations concernant la génération de vidéos, dans le cadre du module Ingénierie dirigée par les Modèles (IDM).

Un cahier de charges détaillé nous a été fourni au début du projet.

## 1. Introduction:

Dans ce projet nous développons un logiciel capable de générer plusieurs variantes de vidéos via le navigateur. L'utilisateur pourra interagir avec l'application en demandant une suite de vidéos.

Le générateur est conçu à partir d'une spécification textuelle (VideoGen) .

Une interface Web est fourni aux utilisateurs pour une meilleure expérience.

## 2. Solution:

Pour développer un outil qui génère plusieurs variantes différentes à partir d'une grammaire qu'on doit respecter, nous avons pris le temps de penser à un plan qui nous permet de couvrir toutes les variantes possibles.

Nous avons développé des fonctions qui, au final, arrivent à générer toutes les variantes possibles. Nous avons commencé par une fonction qui calcule toutes les combinaisons possibles des vidéos de type "Optional", ensuite on bifurque sur les vidéos de type "Alternative" et enfin nous ajoutons à chaque variante la liste des vidéos de type "Mandatory".

Une fois que nous avons généré la liste des variantes, nous créons une playlist dans conformément à ce que "FFMPEG" demande dans un dossier (variantes).

Exemple:

```
file 'playList/Videos/optional.mp4'  
file 'playList/Videos/mandatory1.mp4'  
file 'playList/Videos/alter1.mp4'
```

Ensuite, nous parcourons tous les fichiers du répertoire “variantes” et on exécute les commande “FFMPEG” qu’il faut pour la génération de:

- Vidéo: en concaténant toutes les vidéos de la variante, dans un répertoire propre aux vidéos générées (VideosGenerated).

commande :

```
"ffmpeg -safe 0 -f concat -segment_time_metadata 1 -i playlist1.txt -vf  
select=concatdec_select -af aselect=concatdec_select,aresample=async=1  
videoName.mp4"
```

- Image: Nous prenons une capture de la vidéo générée (variante) à la troisième seconde, dans un répertoire (Images).

commande :

```
"ffmpeg -y -i videoName.mp4 -r 1 -t 00:00:03 -ss 00:00:03 -f image2  
playList/Images/videoName.png";
```

- GIF: Nous le générons à partir de la troisième seconde avec un durée de sept secondes, dans un répertoire (GIF).

commande :

```
"ffmpeg -ss 3.0 -t 7.0 -i videoName.mp4 -f gif gifName.gif";
```

- Duration: Afin de calculer la durée totale des variante.

commande :

```
"ffprobe -v error -show_entries format=duration -of default=noprint_wrappers  
=1:nokey=1 videoName.mp4"
```

Nous avons créé un fichier CSV qui liste toutes les vidéos ainsi leurs présences ou pas dans une playlist.

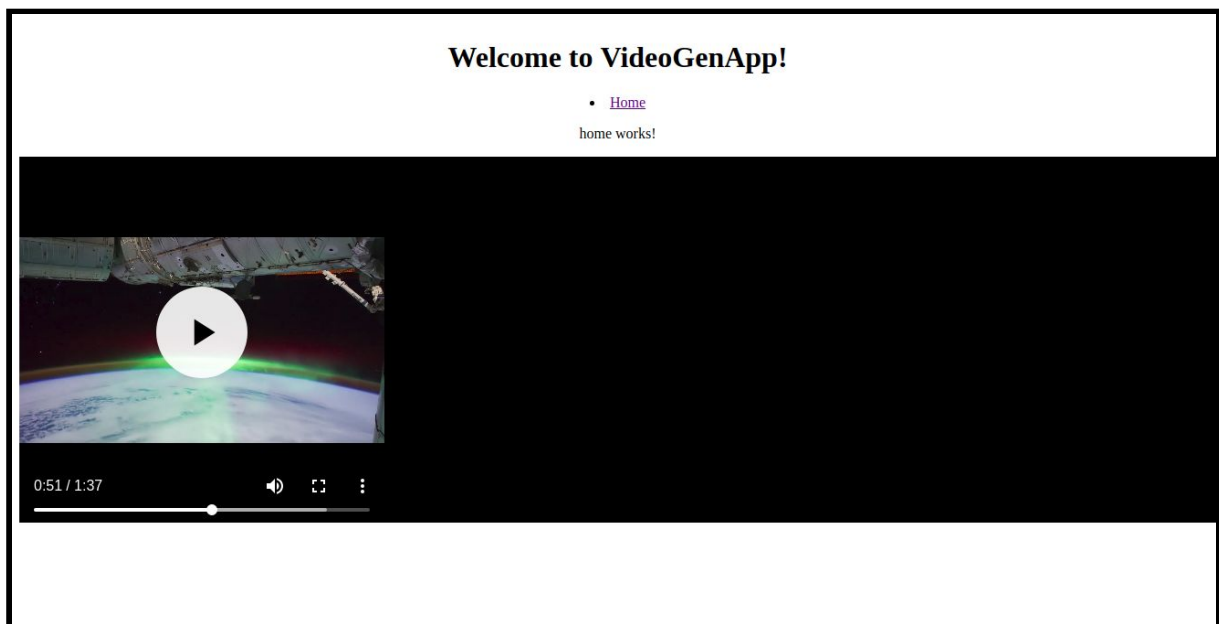
	A	B	C	D	E	F	G
1		mandatory2.mp4	mandatory1.mp4	alter3.mp4	optional.mp4	alter1.mp4	alter2.mp4
2	variante3	True	True	True	False	False	False
3	variante2	True	True	False	False	False	True
4	variante1	True	True	False	False	True	False
5	variante6	True	True	True	True	False	False
6	variante5	True	True	False	True	False	True
7	variante4	True	True	False	True	True	False

Afin de faire communiquer le générateur et la partie Web, nous avons commencé par convertir notre projet en projet Maven, et implanté nos services via l'API REST.

Pour visualiser les vidéos sur le terminal de l'utilisateur, nous avons créé un projet en utilisant Angular5 ainsi que l'API Videogular.

Exemple de composant utiliser pour la lecture de vidéo:

```
<vg-player>
  <video [vgMedia]="media" #media id="singleVideo" preload="auto" controls width="400"
height="400">
    <source src="http://static.videogular.com/assets/videos/videogular.mp4"
type="video/mp4">
  </video>
</vg-player>
```



### 3. Evaluation:

Pour tester notre projet, nous avons procédé comme suite:

- Téléchargement des vidéos des autres binômes ainsi que les spécifications VideoGen se trouvant sur le Drive.
- Chargement des vidéos dans notre projet et configuration des paths.
- lancement du main "[VideoGenTest1.java](#)" du projet "VideoGenTransformer".

A la fin de l'exécution, nous remarquons ce qui suit dans les dossiers qui correspondent (playList/VideosGenerated/ pour les vidéos générées ..):

- Des vidéos générées à partir les vidéos de départ et qui correspondent aux variantes.
- Des images capturées des vidéos générées.
- Des GIFs à partir des vidéos générées.

Nous avons vérifié que les vidéos générées correspondent aux variantes qu'il faut, GIF et images qui correspondent à chaque variante.

#### **4. Discussion:**

- Nous avons réalisé l'ensemble des spécifications essentielles demandées dans le cahier des charges.
- La prise en main et la gestion des modèles était une tâche complexe, nous avons pris un certain temps pour arriver à manipuler la grammaire Xtext et pouvoir réaliser les fonctionnalités du générateur.
- Le calcul des différentes variations de séquences nous a pris un peu plus d'effort vu le nombre important de possibilités.
- La qualité des vidéos générées était un challenge pour nous, car nous avons pas mal perdu du temps dans cette étape afin d'avoir des vidéos fluides.
- Nous n'avons pas pu aller plus loin en implémentation du service Web, par manque du temps vu les différents projets que nous réalisons en parallèle.
- Nous avons eu des difficultés pour la mise en place du notre service car nous avons besoin de le convertir en projet maven afin d'implémenter notre API.
- La mise en place du site Web n'a pas pu être finalisée à cause de la communication qui n'a pas été finalisée avec le service.

#### **5. Démarche empirique :**

- Afin de faciliter la génération des vidéos ainsi limiter le temps d'attente pour les utilisateurs, nous avons exploré plusieurs pistes, surtout au niveau de transmission du flux de données.
- Dans un premier temps nous avons généré toutes les séquences ainsi les icônes et les Gifs au démarrage du serveur. Lorsque l'utilisateur clique sur la génération de séquence, nous choisissons une suite de séquences (10 vidéos) à lui transmettre via notre service. Ce qui limitera le temps d'attente.

- Une autre façon de procéder mais sans doute plus gourmande en temps, mais plus légère pour le démarrage du serveur, c'est de générer les séquences dès le clique, et transmettre la suite de séquences.
- Nous avons également amélioré la qualité des vidéos pour une meilleure diffusion.
- Plus il y'a de vidéos Optional, plus nous avons plus de séquences, car il fallait explorer toutes les possibilités.
- La taille des vidéos joue un rôle essentiel dans la robustesse du générateur.
- Le calcul de toutes les combinaisons est une tâche importante, c'est pourquoi nous avons choisi de générer les vidéos au démarrage du serveur et pas au click de l'utilisateur.

## **6. Conclusion:**

Le projet réalisé dans le cadre de ce module est très intéressant et assez riche en contenu, nous avons pu monter en compétence en manipulation des modèles ainsi que la métamodélisation, l'utilisation des DSL et la transformations de modèles.