Manuel d’utilisation

Introduction

Analyses With VLC permet de faire des analyses de film à l’aide de l’applicationVLC.

Principes de base

- Le film concerné est toujours le film ouvert dans VLC (plus tard, on pourra imaginer qu’il soit conservé dans le config hot de **AWV** afin de pouvoir faire des opérations comme l’exportation sans avoir à lancer le film précédemment).
- On pratique **l’analyse depuis le terminal**, grâce à la commande `awv` qui doit être définie dans le bash-profile.
- **Pour éditer les données**, on passe par les fichiers `YAML` produit. On n’édite pratiquement aucune donnée directement.

Principe de base dans la collecte

- tout premier mot qui termine par un « - » (tiret) interrompt la lecture
-

tout texte placé après un «	» (trait droit) dans le résumé ne s’affichera pas dans le résumé (ça sert par exemple à indiquer qu’un évènement concerne un personnage, ou un autre élément)

Obtenir de l’aide

Jouer la commande :

`🅰️🅰️ help`

> Ajouter l’option `-w/—write` pour modifier le fichier. ~~~ 🅰️🅰️ help -w ~~~

Opérations de base

Pour lancer l’application

- Ouvrir le film dans VLC
- ouvrir une fenêtre de Terminal
  - TIP: on peut même en ouvrir une deuxième pour pouvoir relancer la lecture, par

exemple, lorsque l'on est en mode d'édition d'un élément, ou se déplacer à un autre endroit.
    - TIP: on peut aussi ouvrir une autre fenêtre en lançant la comment `follow` pour suivre les scènes et les éléments déjà collectés.
- la placer sous le film VLC (il faut pouvoir voir les deux parfaitement)
- taper `/Users/philippeperret/Programmation/Analyses_With_VLC/main.rb`

> Note : c'est un exécutable, donc il se lancera tout seul.

Terminer l'application

- taper `exit` ou `quit` dans l'invite pour finir le programme.

Commandes de navigation de base

Commande	Effet	Description / notes
`<vide>`	Met en route la lecture si le film est arrêté Stoppe la lecture dans le cas contraire	C'est un « toggle »
`play`	Pour mettre en route la lecture	
`stop` / **`pause`**	Pour stopper la lecture (pause)	
`-xxx`	Pour remonter de `xxx` secondes	Ne change rien à la lecture. Elle continue si elle était en route, elle s'interrompt dans le cas contraire.
`+xxx`	Avance de `xxx` secondes	Idem

Création assistée d'un élément quelconque

Une procédure particulière s'applique pour créer des évènements particuliers comme des scènes, des infos, etc. Ils fonctionnement de cette manière :

- on frappe une lettre pour « poser un temps » (par exemple un « s » pour une nouvelle scène)

- le programme mémorise le temps et demande le résumé de l'élément (en continuant de jouer le film sans si la lettre est suivie de '-' — tiret)
- le programme enregistre l'élément consigné,
- il reprend la lecture si elle a été arrêtée par un tiret.

Les lettres commandes

Lettre	Produit	Description
`s`	Une nouvelle scène	
`s-`	Une nouvelle scène en arrêtant le temps	
`m`	Une nouvelle marque	Cf. Les marques
`m-`	Une nouvelle marque en arrêtant le temps	Mais c'est un peu inutile dans le sens où la pose d'une marque fait toujours pauser la vidéo.
`n`	Une note	
`n-`	Une nouvelle note en arrêtant le temps	
`i`	Ajouter une information	
`i-`	Ajouter une information en arrêtant la lecture.	
`q`	Pour quitter l'application	

Liste des commandes

Quand on a lancé l'application on peut jouer les commandes depuis l'invite de cette application.

Opération	Commande	Description
`help`	Pour ouvrir ce fichier d'aide.	Avec l'option `-w/--write`, on ouvre le fichier Markdown

Opération	Commande	Description
		pour le modifier.
`film`	Pour gérer le film	Cf. Le film
`goto` `<temps>`	Rejoint le temps donné en argument, ou la scène, dans un second temps	La valeur de `<time>` peut être : 1) un nombre de secondes, 2) une horloge
`goto- <temps>`	Rejoint le temps en arrêtant la vidéo	
`scene` `<résumé|autre>`	Crée une scène au temps courant de la vidéo	`<résumé>` est un résumé court de la scène. Elle pourra être décrite plus en détail plus tard. Après la barre verticale, on peut mettre, par exemple, les IDs des personnages qui apparaissent dans la scène mais n'ont pas pu être placés dans le résumé.
`scenes`	Affiche la liste des scènes courante, pour en rejoindre une en particulier	La liste des scènes est définie dans le fichier `scenes.txt` du dossier du film concerné.

Opération	Commande	Description
		Hérite des méthodes des classes de fichier `YAML`
`personnages`	Affiche la liste des personnages, avec leur ID, leur fonction, etc.	Hérite des méthodes des classes de fichier `YAML`
`export`	Produit l'exportation de l'analyse du film. De nombreuses options permettent de définir cette exportation.	
`info[s]`	Affiche les informations sur le film ou les demande si elles ne sont pas définies.	Ces informations sont consignées dans le fichier `infos.yaml` du dossier AWV du film. Hérite des méthodes des classes de fichier `YAML`
`note[s]`	Affiche les notes sur le film ou les demande si elles ne sont pas définies.	Hérite des méthodes des classes de fichier `YAML`
`pfa`	Permet de travailler sur le Paradigme de Field Augmenté.	
`intrigues`	Commande principale pour travailler avec les intrigues	Hérite des méthodes des classes de fichier `YAML`
		Cette commande est à jouer de

Opération	Commande	Description
`follow`	Suivre les scènes et les éléments collectés.	préférence dans une fenêtre indépendante, puisqu'elle « bloque » l'utilisation du prompt.

Méthodes des classes de fichier `YAML`

On appelle « classes de fichier `YAML` » les classes comme `PFA` ou `AWV::Intrigues` qui utilisent un fichier `YAML` pour enregistrer leurs données.

Ces classes héritent du module `YAMLFileMethods` qui définit les méthodes suivantes :

`edit`
: Permet d'éditer le fichier `YAML`
Par exemple `personnages edit` ou `intrigues edit`.

`list`
: Liste toutes les instances.
Par exemple `note list` ou `infos list` (au pluriel ou non, peu importe)

Le film

`AWV` travaille toujours avec un film ouvert dans `VLC`.

Tous les éléments de l'analyse et de la collecte sont placés dans un dossier à la même hauteur que le fichier vidéo du film.

Définir les données du film (informations)

Les données du film (titre, etc.) sont consignés dans un fichier `infos.yaml`.

Si ce fichier n'existe pas, l'application demande les informations pour le créer automatiquement.

Pour l'éditer, il suffit ensuite de jouer la commande :

🅰️🅰️ `film infos`

Éditer toutes les données du film

On peut ouvrir le dossier complet de l'analyse du film dans un IDE (par exemple dans Sublime Text) à l'aide de la commande :

```
AA film edit
```

Analyse

Le but de **AWV** est principalement d'assister à la création des analyses. Pour ce faire, plusieurs outils sont spécialement dédiés à cette question.

Les « Marques »

Les « marques » sont des éléments de collecte de n'importe quel type qu'on place aux endroits où surviennent des évènements, des informations, etc. Ils permettent un traitement très modulaire de l'analyse, en pouvant créer des types propres à chaque analyse.

Ils sont conservés dans le fichier `marques.yaml` du film.

Une marque se caractérise par :

- son type (qui est défini pour chaque film et conservé dans la données `config.yaml` du film),
- son résumé,
- son temps de début (`start_time`),
- son temps de fin (`end_time`) (et donc sa durée),
- sa description (qui peut être éditée et modifiée par le fichier)

Création d'une nouvelle marque

```
m[ <type>[ <résumé>|<autres données>]]
```

Si l'on donne seulement `m` (ou `m-` pour stopper la vidéo), le programme demande le type de l'élément et permet d'en créer un nouveau.

Si le type est donné, et qu'il n'existe pas encore, le programme demande confirmation que c'est bien un nouveau type qui est fourni (pour éviter d'oublier de mettre le type).

Édition des marques

Comme pour les autres éléments, on peut éditer les marques à l'aide la commande :

```
AA marques edit
```

Liste des marques

Comme pour les autres éléments, on peut éditer les marques à l'aide la commande :

```
🅰🅰 marques list
```

Paradigme de Field Augmenté

C'est la commande **`pfa`** qui permet de définir les éléments du PFA. En lançant la commande sans argument, le programme présente une liste des choses qu'on peut faire. Parmi celles-ci :

- Construire le paradigme de Field Augmenté à partir des informations données.
- Définir un point du paradigme de Field.
- Initier un nouveau paradigme (un film pour posséder une infinité de paradigme différents).

Définir un nouvel élément

Commande complète :

```
🅰🅰 pfa ne[ <type>][ --pfa=<index>]
```

Si le `type` n'est pas défini, la commande le demandera (ce qui permet de tous les voir, par la même occasion).

Si `—paf` n'est pas défini, c'est le paradigme courant qui sera utilisé (il sera rappelé à titre indicatif et pourra être modifié).

La commande demande ensuite de donner une description du nœud.

Construire le ou les paradigmes de Field Augmenté

```
🅰🅰 pfa build
```

S'il y a plusieurs paradigmes définis, la méthode demande lequel faire (on peut aussi les faire tous). Pour faire directement un PFA en particulier, ajouter son index (les paradigmes sont comptés à partir de 1).

```
🅰🅰 pfa build <index>
```

Classe des méthodes des fichiers **`YAML`**

`pfa` est une classe qui hérite des*méthodes fichiers* *`YAML`*, on peut donc utiliser toutes les [méthodes des classes de fichiers `YAML`][].

Les Intrigues

C'est la commande **`intrigues`** qui permet de définir les éléments des intrigues. En lançant la commande sans argument, le programme présente la liste des choses qu'on peut faire :

- **Construire le rapport statistique** à partir des intrigues définies.
- **Définir une nouvelle intrigue**.
- **Définir les scènes des intrigues**. Cet outil permet de passer en revue chaque scène pour indiquer à quelle intrigue elle appartient (on affiche le résumé de la scène puis la liste des intrigues, qu'on peut choisir avec une lettre).

Annexe

Faire des essais

On peut faire des essais ou jouer du code sur les données du film courant en définissant le code du fichier `lib/commandes/try/code_to_run.rb` puis en jouant la commande `try`.

Lexique

Collecte
: la collecte est l'opération qui consiste à relever toutes les informations dans le film, qui doivent servir à l'analyse future.

Premier mot
: le "premier mot" de la ligne de commande est considéré comme la commande à exécuter.
note : s'il se termine par un tiret, on arrête la lecture de la vidéo.
