

TD4 : IHM VideoTracker

Compétences

- Création de packages Python
- Définir l'IHM du projet VideoTracker
- Mettre en place le pattern MVC

IHM — VideoTracker

Ensemble des dispositifs matériels et logiciels permettant à un utilisateur d'interagir avec un système interactif.

Caractéristiques - prototypée

Obtenir une interface finale passe par plusieurs étapes :

- Croquis (sketch) = aperçu global de l'interface (idée générale)
- Maquette (mockup, wireframe) = interface détaillée (sans interaction)
- Prototype = version incomplète d'une interface (avec interactions)

Caractéristiques - utilisateur

Identifier les caractéristiques pertinentes de l'utilisateur

- Taille, âge, genre, handicaps
- Niveau de formation, habitudes culturelles (e.g., format des dates, sens d'écriture)
- Novice, expert, professionnel
- Usage occasionnel, quotidien



Caractéristiques - évaluation

L'évaluation repose généralement sur des scénarios. Un scénario associe :

- Un utilisateur (ou un groupe)
- Un contexte (environnement et contraintes)
- Une ou plusieurs tâches (que l'utilisateur doit accomplir)

Une méthode de conception IHM nécessite de collecter des informations sur les utilisatrices, leurs tâches ou sur les évaluations des interfaces.

Questions

1. Dans le cahier des charges du projet VideoTracker, identifiez tous les besoins en rapport avec l'IHM et transformez les en exigences.
2. Dessinez sur une feuille de papier l'IHM de votre application VideoTracker.
3. Téléchargez l'archive `src_student.zip`.
4. Placer correctement les fichiers de l'archive dans le répertoire du projet VideoTracker.
5. Mettre à jour votre dépôt git local et distant.
6. Quelles sont les instructions qui permettent de lancer l'application ?
7. Étudiez la section 6.4 de ce lien <https://docs.python.org/fr/3/tutorial/modules.html> puis ajoutez les imports nécessaires pour lancer l'application
8. Que contient `la_vue` de notre application ?
9. Dans le concept MVC, quels sont le(s) rôle(s) des classes `FileRepo` et `Point`. Placez-les dans le bon répertoire.
10. Terminez la partie IHM liée au besoin de la sauvegarde des données.

Lecture d'une vidéo

Pour lire une vidéo nous allons utiliser le package OpenCV.

Reading Video using OpenCV

Reading and writing videos in OpenCV is very similar to reading and writing images. A video is nothing but a series of images that are often referred to as frames. So, all you need to do is loop over all the frames in a video sequence, and then process one frame at a time. In this post, we will demonstrate how to read, display and write videos from a file, an image sequence and a webcam. We will also look into some of the errors that might occur in the process, and help understand how to resolve them. Learn OpenCV : <https://learnopencv.com/reading-and-writing-videos-using-opencv/>

1. Téléchargez le fichier `play_video.py`. Placez une vidéo de votre choix dans le même dossier, renseignez `path_video` dans le code et lancez la vidéo.
2. Implémentez la (les) tâche(s) affectée(s) à la lecture complète de la vidéo.