Université de Bordeaux
Semestre 2

2021-2022

TD2 : Cycle de développement

Compétences

- À partir d'un besoin, mettre en place un cycle de développement complet
- Découvrir la notion d'IHM et la programmation par événements
- Utiliser le module tkinter

IHM — Interface Homme Machine

Les IHM servent à rendre les programmes plus interactifs. Elles simplifient la vie des utilisateurs mais elles demandent plus de temps pour les concevoirs et les développer. Un programme sans interface exécute des instructions les unes à la suite des autres. Avec une IHM le programme attend un événement - pression d'une touche du clavier, clic de souris - pour exécuter une un bout de code. C'est comme si le programme avait une multitude de points d'entrée.
Il existe plusieurs modules pour développer une interface graphique.

- tkinter : le plus simple mais limité. Visuellement, tkinter est moins joli que d'autres extensions mais c'est un package facile à installer et pour cette raison c'est celui que nous utiliserons.
- wxPython : est une librairie communautaire.
- PyQt5 : est une librairie professionnelle bien plus fournie et bien plus complète.

La conception d'une interface graphique se déroule généralement selon deux étapes. La première consiste à dessiner l'interface, c'est-à-dire choisir une position pour les objets de la fenêtre (boutons, zone de saisie, liste déroulante, ...). La seconde étape définit le fonctionnement de la fenêtre, c'est-à-dire associer à chaque objet des fonctions qui seront exécutées si un tel événement se réalise (pression d'un bouton, pression d'une touche, ...).
Ces deux étapes sont très fortement liées avec l'expression des **exigences**.

1. À partir du cahier des charges exprimer les exigences correspondantes aux besoins liés :
   - au chargement de la vidéo
   - à la lecture complète de la vidéo
2. Écrire les tâches associées à chacune des exigences.

Le package tkinter

L'application

Télécharger le fichier `src_etudiant.zip`.

1. Quel est le fichier qui permet de lancer l'application ?
2. Étudier la section 6.4 de ce lien https://docs.python.org/fr/3/tutorial/modules.html puis ajouter les imports nécessaires pour lancer l'application
3. Quelle est la classe responsable de la vue graphique (IHM) ?

L'IHM de l'application

Les interfaces graphiques sont composées d"objets ou widgets ou contrôles. Voici quelques excemples de widgets :

- **Label** : A widget used to display text on the screen
- **Button** : A button that can contain text and can perform an action when clicked
- **Entry** : A text entry widget that allows only a single line of text
- **Text** : A text entry widget that allows multiline text entry
- **Frame** : A rectangular region used to group related widgets or provide padding between widgets

Ce lien https://realpython.com/python-gui-tkinter/ donne une description détaillée et des exemples d'utilisation de ces widgets. Pour prendre un bon départ il est important d'avoir une idée précise de la structure de son IHM et des différents widgets dont vous aller avoir besoin.
Un autre lien en français (à partir de la page 49) : https://hal.archives-ouvertes.fr/hal-02126596/document

1. Identifier tous les besoins en rapport avec l'IHM et les transformer en exigences
2. Dessiner sur une feuille de papier l'IHM de votre application VideoTracker. Cela ne devrait pas être trop difficile puisqu'il suffit de rassembler les morceaux des différentes exigences liées à l'IHM.

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. Télécharger le fichier `play_video.py`. Placer une vidéo de votre dans le même dossier renseigner `path_video` dans le code et lancer la vidéo.
2. Implémenter la (les) tâche(s) affectées à la lecture complète de la vidéo