# TD7 – Projet - Note d’intention

*(max 5 page)*

### Explications

Nous avons fait un **système de filtre sur la vidéo webcam en direct**. Pour la fenêtre et la gestion des Menu, nous utilisons l’interface graphique Tkinter. On peut désactiver ou (ré)activer les filtres dans le Menu nommé *« Choix des filtres* ».

Voici les filtres qu’on peut appliquer :

- filtre lunette : Des lunettes incrustées sur la vidéo et positionnés au niveau des yeux.

- filtre foulard : Un foulard incrusté sur la vidéo et positionnés autour de la tête.

- filtre grain de beauté : Un grain de beauté incrusté sur la vidéo et positionnés au-dessus des lèvres.

- filtre saturation : Filtre appliquée sur toute l’image, saturation plus élevée, pour donner l’effet année 70.

- filtre fond : Un sous menu contenant différent arrière-plan

Remarque : *On ne peut utiliser qu’un fond à la fois, si l’un est activé, tous les autres sont désactivés par défaut.*

- fond océan : active un arrière-plan statique avec un feuille de palmier au premier plan

- fond cabane : active un arrière-plan statique de nuit, avec une cabane sur l’eau

- fond animée plage : active un arrière-plan animé (des mouettes volant sur le haut de l’écran, sur 4 images seulement l’animation)

Une image contenant texte, Page web, Site web, logiciel

Description générée automatiquement

Le **thème est « une femme des années 70 à la plage ».**

Pour quitter la fenêtre et arrêter le programme, il faut aller dans *Choix des filtres>Quitter*.

### Toutes les images générées

A faire durant le TD les vidéos !

### Les problèmes rencontrés et limites

### Mise en place de la barre des Menu

Il était difficile de mettre en place la barre de menu sur la fenêtre « Frame » avec la vidéo, il y avait toujours 2 fenêtres différentes. On avait réussi à mettre une seule image de capture d’écran sur la fenêtre, mais pas la vidéo en entier.

Avec la fonction, *« master.after(10, update\_image)* », on a pu mettre la vidéo en entier, plutôt que qu’une capture d’écran.

### Activation des filtres

Il fallait trouver un moyen d’optimiser le code lors de l’utilisation des filtres. Par exemple, quand on active le filtre « lunette », la vidéo ralentit drastiquement. Ainsi, il fallait que nous pensions à le désactiver. C’est pourquoi, nous avons mis en place les variables globales *bool\_activate\_filtre\_lunette* ou même *bool\_activate\_filtre\_saturation*. Lorsque l’on sélectionne *« Choix de filtre> Activer/Désactiver filtre lunette », c*ela appelle simplement la fonction *filtre\_lunette()* qui va mettre le booléen *bool\_activate\_filtre\_lunette* en son booléen contraire. Et dans l’implémentation du code d’affichage des frames, nous entrons (ou non) dans la condition, *if bool\_activate\_filtre\_lunette.*

Une image contenant capture d’écran, texte, logiciel, Logiciel multimédia

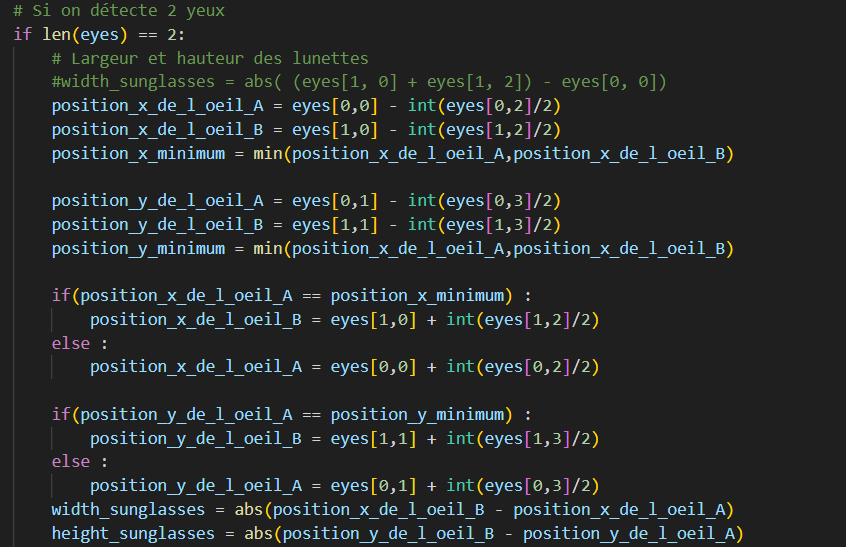
Description générée automatiquement

#### Limite :

La vidéo ralentit lorsqu’on utilise certains filtres. Il faudrait éteindre certains filtres pour le rendre plus fluides.

### Positionnement des lunettes

Il était également difficile de bien positionner les lunettes sur les yeux. Il fallait que l’on calcule les positions des lunettes en fonction des positions des yeux. On a dû décortiquer le vecteur « *eyes* » (le résultat de détection des yeux) pour obtenir les dimensions exactes :



#### Limite :

A quelques fréquences, les lunettes disparaissent. Le filtre lunette ne s’active que lorsque l’on détecte les 2 yeux. Cependant, parfois, il n’en détecte qu’un seul. Aussi, lorsqu’on pivote légèrement notre tête, les lunettes ne pivotent pas avec nous.

### Mise en place du filtre Foulard et du Grain de beauté

Pour le visage, on voulait mettre un filtre foulard et grain de beauté. On a donc dessiné et fait son masque par la même occasion via le logiciel Procreate. Le plus dur était de calibrer le filtre sur la tête ou au-dessus de la bouche. On a donc modifié les paramètres jusqu’à ce que les filtres aient l’air cohérent.

Autre difficulté, lorsque des pixels de notre filtre foulard dépassait la zone d’image de la webcam, ça nous faisait une erreur « out of band». Ainsi, pour contrer cela, on a fait le choix de faire apparaitre la partie hors champs sur son bord opposé (traitement des bords : extension).

### Lenteur du programme

Pour l’incrustation de l’image de fond, à l’origine, on voulait faire une double boucle qui parcourait toute l’image et incrustait l’arrière-plan. Cependant, cela ralentissait drastiquement le programme. Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Pour le remplacer et pour l’optimiser, nous avons l’opérateur par défaut ‘ == ’(l’égalité) des images :

Une image contenant texte, Police, capture d’écran, nombre

Description générée automatiquement

Ainsi, le traitement est redevenu fluide.

### Calibrage de la détection d’arrière-plan

Au début, lorsqu’on testait le filtre arrière-plan sur plusieurs différents fonds (une fois, on a testé sur un fond blanc puis sur un fond bleu), parfois, le filtre ne s’activait pas totalement (juste quelques bruits sur les coins de la vidéo). On constatait que ça fonctionnait surtout pour les fonds blancs. On a regardé notre code et on a constaté que dans notre condition d’incrustation d’image, on a posé une égalité parfaite entre le plan actuel et le plan de fond. Mais aussi, on examinait seulement le canal rouge (le ‘2’ su image ci-dessous). C’est pour cela que le filtre arrière-plan pouvait varier.



Ainsi, au lieu d’avoir une condition d’égalité parfaite, on a mis une condition sur une intervalle d’écart type de 20 d’intensité. Enfin, au lieu d’analyser le canal rouge, on analyse plutôt sur le niveau de gris.

