Fiche de passation projet PRESTON

Etudiant FIE-4 :	Charbonnier Gaëtan
Mail Universitaire :	gaetan.charbonnier@etud.univ-jfc.fr
Mail Personnel :	gaetancharb.98@gmail.com
Téléphone :	07-66-63-40-51

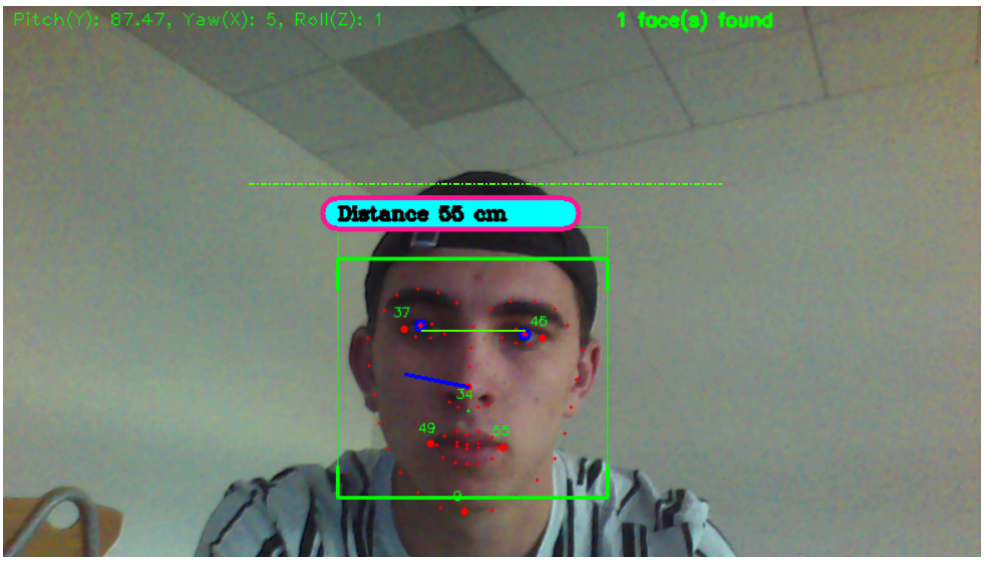
Synthèse globale Fichiers

Réalisation d'une interface Tkinter, Python : Fichier Login.py et Interface.py (Possibilité de regroupement des deux programmes, attention aux appels des boutons.)

FaceWbc_4.py est le programme principal, c'est depuis ce dernier que sont enregistrées les différentes données. Ce programme est le plus conséquent, on y retrouve en effet les deux méthodes très similaire de détection de visage, l'une permettant de détecter approximativement un visage et de de calculer sa distance en fonction de la caméra, l'autre permettant la détection des Facial Landmark (68 points spécifique du visage) qui sert au calcul des différentes variations d'angles (Pitch, Roll, Yaw).

Base de données - Fichier CSV DataWebtrack contenant 5 jeux de données (liste de Temps, liste de Pitch, Roll, Yaw, liste de Distance), plus les dates et heures d'allumage et de fermeture du programme principale FaceWbc_4.py.

FaceWbc_V5.py, ajout d'un code permettant d'obtenir une ligne à hauteur des yeux. La réflexion apportée était de récupérer l'emplacement de cette ligne ou du point central de cette ligne, dans le but de définir une hauteur au niveau de l'écran sur lequel la ligne devrait se positionner pour être à bonne hauteur. (les deux lignes Verte à ajouter au ¾ de la fenêtre et entre les deux yeux, puis à calibrer avec les point des yeux : si ligne des yeux sur ou proche de ligne de positionnement, alors ligne verte, sinon ligne rouge)
http://datahacker.rs/010-how-to-align-faces-with-opencv-in-python/
Une autre solution serait d'insérer une forme ovale dans laquelle l'utilisateur placerait son visage pour être correctement positionné.



FaceWbc_V6 a été créer à la fin du stage pour débuter le travail de la création de graphique, la différence majeure est le placement de la création du csv, ici directement à l'intérieur de la boucle while à la différence de FaceWbc_4 qui récupérait des listes de données créer dans la boucle.

FaceWbcGame2 sera le programme faisant office de préface à l'utilisateur dans le but de se positionner avant de lancer le programme principal (Ajout de la ligne de hauteur vu précédemment possible).

To Do List, Task and Reminder

1. Lancement Projet Webcam Tracking

opencv-python

pip install opencv-python

Librairie dlib

pip install numpy opencv-python dlib imutils

Accorder l'autorisation à vos application

Pour autoriser votre appareil à accéder à la caméra, sélectionnez Démarrer , puis Paramètres > Confidentialité > Caméra . Dans Autoriser l'accès à la caméra sur cet appareil, si l'accès à l'appareil photo pour cet appareil est désactivé, sélectionnez Modifier et activé l'accès à la caméra pour cet appareil.

2. Modification à apporter

Graphique temps hors écran : Si pas de visage détecté, alors distance_level = 0 (En conséquence si personne devant l'écran => Courbe tendant vers 0). Attention : Pour graphique avec moyenne de distance (Distance_AVG), ne pas prendre en compte les 0 (Distance_AVG, si distance_level >20 alors....).

Faux positifs : Ajouter des valeurs limites pour le pitch, roll, yaw, Distance pour éviter les détections de visages de faux-positifs.

Mettre la sortie vidéo dans un tkinter widget :
https://solarianprogrammer.com/2018/04/21/python-opencv-show-video-tkinter-window/

Rendre exécutable le code sur n'importe quel machine :
https://python.doctor/page-cx_freeze-creer-executables-python-cours-apprendre
Converting python to executables using cx_freeze:https://www.youtube.com/watch?v=DoHWJV8iVTQ
Cx_freeze documentation officiel: https://cx-freeze.readthedocs.io/en/latest/

Base de données : Possibilité de chercher une alternative au CSV comme une base de données en ligne possédant une visualisation des données plus intuitive. Base de donnée en ligne pour parer la limite des csv (conserver le nettoyage des données)

Caméra virtuelle avec Obs : Un problème a été soulevé assez rapidement lors d'une des réunions en distanciel via Zoom, ce problème concerne la disponibilité de la caméra lors de visioconférence.
En effet, l'application nécessite la disponibilité totale de la caméra, ce qui pose la nécessité de la création future d'une caméra virtuel en passant par une application tel que Obs (Open Broadcaster Software), un logiciel libre de capture d'écran et de streaming pour Microsoft Windows, MacOS et Linux. Les explications détaillées du Code et l'installation d'OBS de Maik Riechert, Senior Research Software Engineer at Microsoft sont présent sur son GitHub: https://github.com/letmaik/pyvirtualcam

Graphique : Récupération de liste de données complexe à extraire via l'utilisation de csv. Réalisation de nouveaux graphiques avec la nouvelle base de données.
Les graphiques à créer sont : un graphique des données journalières et mensuelles.

Authentification par reconnaissance facial : Pour ne pas relever de donnée inutile, et pour simplifier le logiciel, retirer l'authentification par mot de passe et intégrer l'aspect de facial recognition.

Création d'alerte utilisateur : Création d'alerte personnaliser en fonction des données prélevées concernant la posture de travail

Tester la solution : La phase de test en condition réelle et l’étude d’amélioration possible.