



École Nationale Supérieure des Sciences Appliquées et de Technologie
de Lannion

EyeTracker

ANATOLE FAUGÈRE

DIPLÔME D'INGÉNIEUR SPÉCIALITÉ INFORMATIQUE

Année universitaire 2022 - 2023

Tuteurs :

VINCENT BARREAUD
EMMANUEL CASSEAU

1 Résumé

Ce projet a pour objectif de développer un assistant de communication pour les personnes handicapées moteur. L'utilisateur pourra sélectionner par le regard une des réponses proposées sur un écran. L'application développée est basée sur la technologie Tobii-eye tracker permettant de détecter l'endroit où se situe le regard sur une interface graphique.

Cette application permet également à un aidant d'enregistrer un ensemble de questions et leurs réponses associées dans une base de données, pour une utilisation ultérieure. Deux interfaces graphiques ont été réalisées en *Python l'une plus ergonomique qui est l'interface finale, l'autre plus simple qui sert à effectuer les tests de l'application. Suite à la commande tardive de la *Tobii-eye tracker par l'*Enssat, le projet a pris un peu de retard. La *Tobii-eye tracker est en cours d'intégration dans l'application.

Mots-clés : *Tobii-eye tracker, interface graphique, *Python

2 Abstract

This project aims to develop a communication assistant for people with motor disabilities. The user will be able to select by sight one of the answers offered on a screen. The developed application is based on *Tobii-eye tracker technology to detect where the gaze is located on a graphical interface.

This application also allows a caregiver to record a set of questions and their associated answers in a database, for a later use. Two graphical interfaces have been created in *Python : a more ergonomic one which is the final interface, another one, more simple, that is used to perform the tests of the application. Following the late order of the *Tobii-eye tracker by *Enssat, the project took a little delay. The *Tobii-eye tracker is currently being integrated into the application.

Keywords : *Tobii-eye tracker, graphical interface, *Python

3 Remerciement

En premier lieu, je tiens à remercier M.Vincent Barreaud et M. Emmanuel Casseau mes tuteurs, pour leurs sympathies, leurs conseils tout au long du projet.

Je remercie Mme. Nathalie Caradec, ma professeure d'expression-communication, pour son enseignement et ses explications.

Un grand merci mes camarades de classes pour leurs conseils ainsi que leur soutien inconditionnel, et moral.

Table des matières

1	Résumé	1
2	Abstract	1
3	Remerciement	2
4	Introduction	6
5	Description du sujet	7
5.1	L'Enssat	7
5.2	Le contexte	7
5.3	Les objectifs et les contraintes	7
5.4	Planning prévisionnel	8
6	Problématique et spécifications du projet	11
6.1	La Tobii-eye tracker	11
6.2	L'environnement	12
7	L'état d'avancement	13
7.1	Développement de l'interface	13
7.2	Gestion de la base de données	16
7.3	Problème de Commande et Recherche de solutions alternatives	16
8	Conclusion	17
9	Bibliographies	18

Table des figures

1	Use Case	8
2	Diagramme de gantt partie-1	9
3	Diagramme de gantt partie-2	10
4	1 ^{er} Interface liste des propositions	13
5	1 ^{er} Interface exemple d'eye-tracker	14
6	2 ^{eme} Interface page Home	14
7	2 ^{eme} Interface page Propositions	15

Glossaire

- ***API** est une interface de programmation d'application (application programming interface). 11
- ***Conteneur** est une unité logicielle standard qui regroupe le code et toutes ses dépendances afin que l'application s'exécute rapidement et de manière fiable d'un environnement informatique à un autre (docker.com). 12, 17
- ***Docker** est une plateforme permettant de lancer certaines applications dans des conteneurs logiciels. (Wikipédia). 12
- ***Enssat** : École Nationale Supérieure des Sciences Appliquées et de Technologie.. 1, 6, 7, 11, 16, 17
- ***GitHub** est un service d'hébergement Internet pour le développement de logiciels et le contrôle de version à l'aide de Git. 12
- ***Jenkins** est un outil d'automatisation utilisé pour créer et tester des projets logiciels. 12, 16
- ***Python** est un langage de programmation interprété, multiparadigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. (Wikipédia). 1, 11–14, 16
- ***QML** (Qt Modeling Language) est un langage de balisage d'interface utilisateur. 14
- ***SDK** est un ensemble d'outils d'aide à la programmation d'applications mobiles (Software Development Kit). 11
- ***Tobii-eye tracker** est une technologie de suivi oculaire permettant d'utiliser vos mouvements des yeux comme une entrée supplémentaire dans un système, en complément du clavier et de la souris. 1, 7, 8, 11, 12, 16, 17
- ***Unity** est un moteur de jeu multiplateforme développé par Unity Technologies. 11

4 Introduction

J'ai effectué ce projet de septembre 2022 à février 2023, dans le cadre du projet long en 3^{eme} année à l'*Enssat. Ce projet a été l'opportunité pour moi d'appréhender toutes les étapes de conception, de développement et de test d'un logiciel.

Dans un premier temps, j'aborderai la description du projet EyeTracker ainsi que la problématique générale de celui-ci. Puis, j'explicitai les problématiques techniques du sujet ainsi que ses spécifications. Enfin, je me pencherai sur l'état d'avancement du projet jusqu'à ce jour.

5 Description du sujet

5.1 L'Enssat

L'*Enssat, (École Nationale Supérieure des Sciences Appliquées et de Technologie) est une école d'ingénieurs dans les domaines de l'informatique, des systèmes du numérique et de la photonique. L'*Enssat est une école affiliée à l'Institut Mines-Télécom. Mais il s'agit aussi d'un centre de recherche reconnu pour ces projets nationaux et internationaux. De plus ses étudiants participent à de nombreux projets locaux, proposés soit par l'école soit par la communauté locale. Il se trouve que ce projet a été présenté par un habitant dans la région.

5.2 Le contexte

Effectivement ce projet a été proposé par le grand-père de Yoan. Il se trouve que Yoan est une personne avec un fort handicap physique : il ne peut communiquer que par le regard. Le grand-père possède une *Tobii-eye tracker 4C, un appareil qui permet de détecter où se pose le regard sur un écran, et il souhaitait l'utilisée pour développer une application lui permettant de communiquer facilement avec son petit-fils. Ce projet a été soumis M. Casseau qui la transformé en sujet de projet long pour les 3^{eme} année à l'*Enssat.

5.3 Les objectifs et les contraintes

Le grand objectif du projet EyeTracker est de construire un assistant de communication qui a l'aide d'un eye tracker, va capter le regard de la personne sur un écran et l'afficher par un point de couleur. Ainsi sur une interface graphique on affiche différentes propositions de réponses à une question, la personne avec l' handicap peut via son regard sélectionner une réponse.

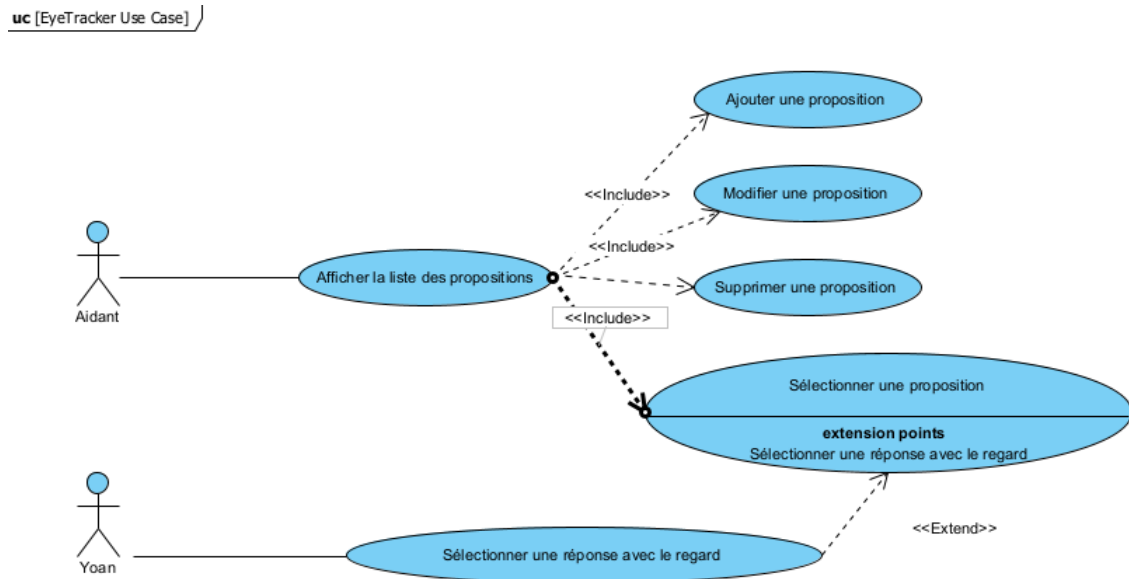


FIGURE 1 – Use Case

A l'aide ce diagramme des cas utilisateurs, on identifie facilement les besoins primaires de l'applications : Afficher les propositions, en ajouter, les modifier ... Ici on appelle une "proposition" un ensemble contenant une question et 2 à 4 réponses associées.

Les contraintes du projet sont les suivantes :

- Le grand-père de Yoan possède une *Tobii-eye tracker. Notre système devra ré-utilisé cette technologie.
- L'application devra etre extremement simple et intuitive lors de son utilisation.

5.4 Planning prévisionnel

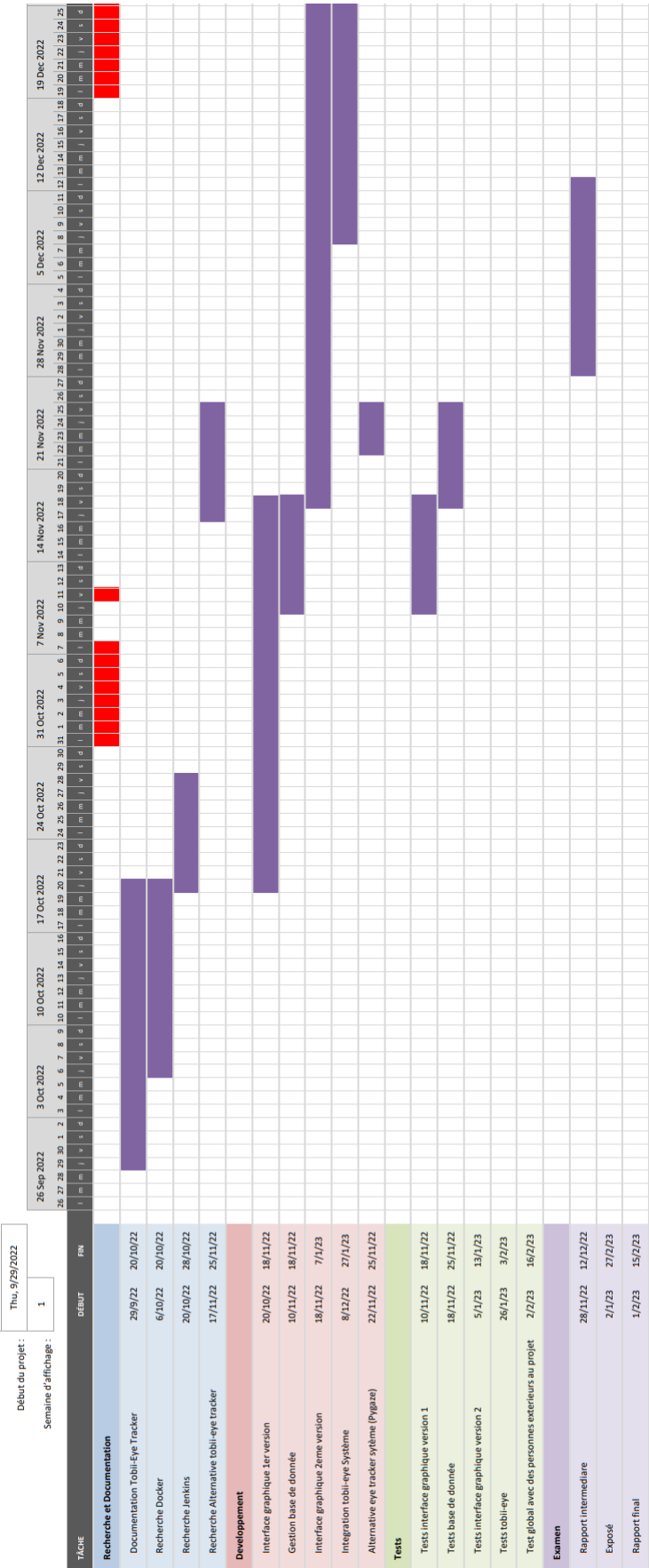


FIGURE 2 – Diagramme de gantt partie-1

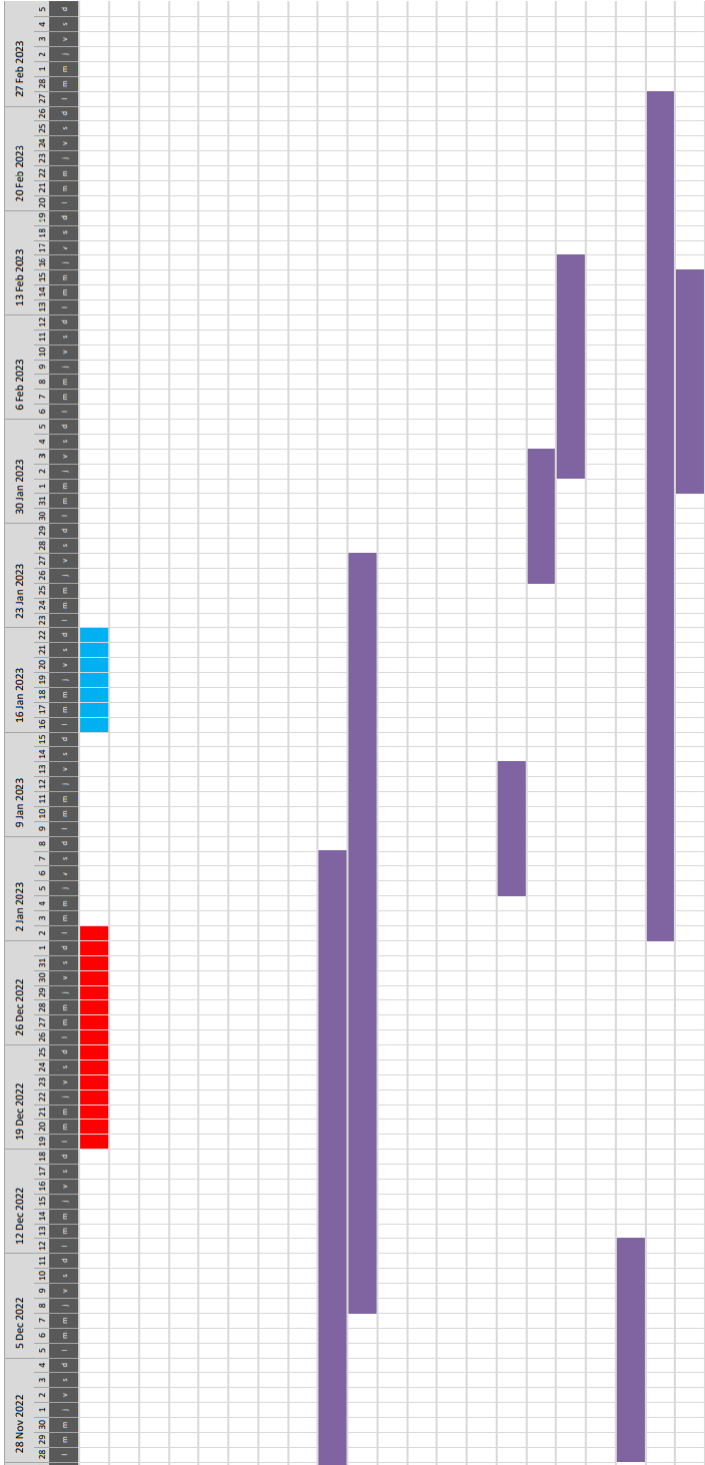


FIGURE 3 – Diagramme de gantt partie-2

6 Problématique et spécifications du projet

6.1 La Tobii-eye tracker

La *Tobii-eye tracker est une technologie de suivi oculaire permettant d'utiliser vos mouvements des yeux comme une entrée supplémentaire dans un système, en complément du clavier et de la souris.[4]

Le grand-père de Yoan possède une *Tobii-eye tracker 4C, mais cette version n'est plus en vente actuellement : on est maintenant à la 5^{eme} génération. J'ai donc fait une étude comparative entre la version 4C et la version 5.

	Tobii Eye 4C	Tobii Eye 5
Driver and software	À installer manuellement	Déjà installé sur la machine
Logiciel	Tobii Core	Tobii Expérience
Taille	335 mm	285 mm
Capteur	IS4 avec capteur NIR standard	IS5 avec capteur Tobii NIR personnalisé
Champ de vue	38*29 degrés	40*40 degrés
Taille de l'écran supporté	27" 16 :9 or 30" 21 :9 *	27" 16 :9 or 30" 21 :9 *
Taux d'échantillonnage de l'image et fréquence du regard	90Hz	133 Hz
Récupération du regard	Sélecteur de mode pour récupérer le regard	Récupération continue

Après analyse, on remarque que toutes les fonctions de l'*API de la Tobii-eye que l'on utilisera seront les même pour les version 4C et 5.

De plus la *Tobii-eye tracker, peut être utilisée sur les systèmes Windows, MacOS et Linux. Elle peut être programmée en *Python, Matlab, Octave, C, ou en C# avec *Unity [3]. Me spécialisant plus en *Python et C#, j'ai privilégié le *Python afin qu'un autre étudiant puisse poursuivre le projet et le compléter dans les années futures.

(le C# et *Unity n'étant pas enseigné à l'*Enssat).

J'ai utilisé la licence : "Getting Started development license" [2]. Cette licence est destinée à un usage non commercial et à la non distribution. Il s'agit d'une licence limitée qui fournit des droits de développement avec les *SDK Tobii et les logiciels et l'*API [1] liés à l'oculométrie.

6.2 L'environnement

J'ai conteneurisé mon application, car les *Conteneurs permettent l'isolation des applications entre elles et du système sous-jacent. Ils permettent aussi la portabilité, puisque les applications n'ont pas à être liées au système d'exploitation hôte. Pour cela j'ai utilisé le système de gestion de *Conteneur *Docker. Mon *Conteneur possède une image Debian plus une image *Python3.8 ainsi que différents packages nécessaires au fonctionnement de la *Tobii-eye tracker. Ce conteneur possède 2 ports vers l'extérieur : le premier pour l'affichage graphique de l'application, et le second pour se connecter à la Tobii.

J'utilise aussi le système *Jenkins afin de créer un pipeline de tâches, essentiellement pour des tests.

Enfin je stocke mes codes sources sur *GitHub : <https://github.com/FAtole/EyeTracker>

7 L'état d'avancement

7.1 Développement de l'interface

Afin de répondre aux exigences du projet, il m'a paru indispensable d'avoir une interface graphique utilisateur. En *Python il existe différents bibliothèques pour construire une application de bureau avec une interface graphique : tkinter, PyQt, WxPython, Kivy ...

La première version de l'interface a été effectuée avec tkinter [6] car c'est framework très simple qui permet d'avoir un résultat très rapidement en *Python.

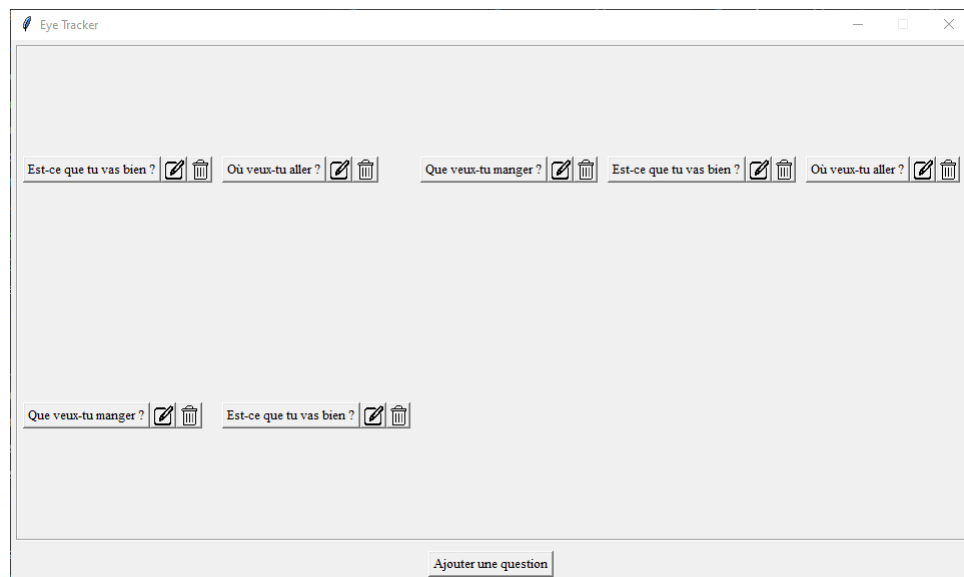
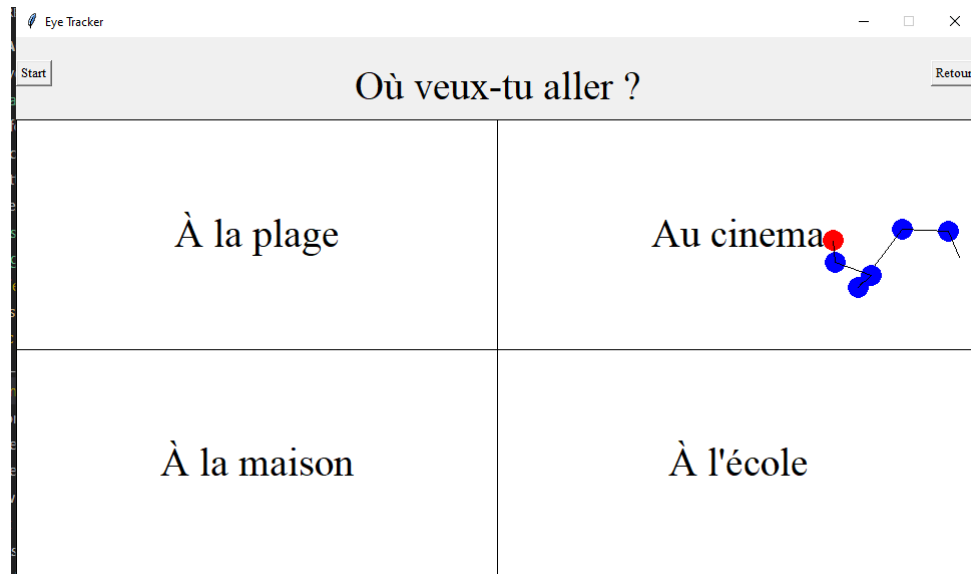


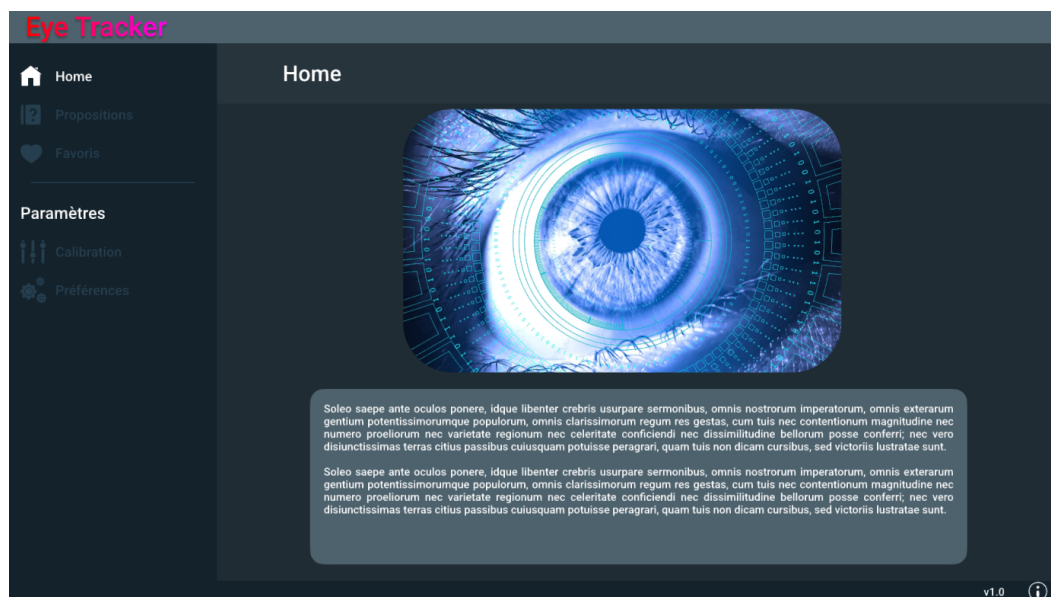
FIGURE 4 – 1^{er} Interface liste des propositions

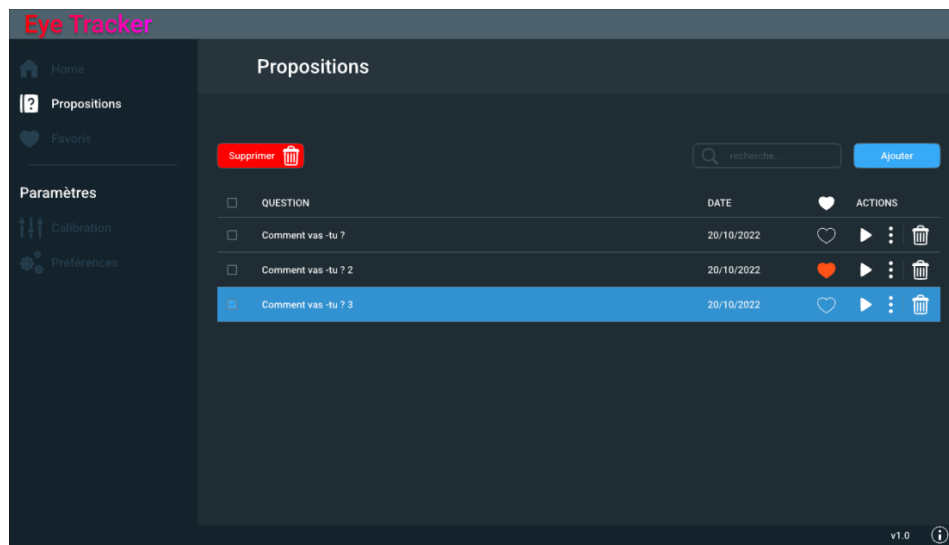
Cette interface possède l'intégralité des fonctionnalités attendues excepté l'import des données de la Tobii-eye.

FIGURE 5 – 1^{er} Interface exemple d'eye-tracker

Cette interface est simple et fonctionnelle mais peu adaptée au cas de Yoan. J'ai décidé d'en refaire une autre plus ergonomique. J'ai d'abord fait des croquis sur un logiciel de dessin, que j'ai validé avec mes tuteur, avant de les implémenter.

La 2^{eme} version est basée sur les librairies PyQt et PySide [7]. J'utilise le logiciel Qt Design Studio pour générer des pages en *QML, qui sont ensuite appelées en *Python.

FIGURE 6 – 2^{eme} Interface page Home

FIGURE 7 – 2^{eme} Interface page Propositions

La page d'Accueil, des Propositions, ainsi que la fonctionnalité d'ajout de Propositions on été implémenté.

7.2 Gestion de la base de données

Les données sous forme de questions / réponses sont stockées en format csv. On utilise la librairie csv pour lire et écrire dans ce fichier.

Les fonctions de lecture, de modifications, d'écriture, et de suppression de propositions ont été implémentées et testées automatiquement avec *Jenkins(tests unitaire, tests fonctionnels, tests d'intégration).

7.3 Problème de Commande et Recherche de solutions alternatives

Le projet à rencontré un point bloquant : mes tuteurs n'ont pas eu l'autorisation de paiement de l'*Enssat pour acheter la *Tobii-eye tracker 5. Ce qui a entraîné un retard monstrueux dans le projet.

J'ai cherché des solutions alternatives à la *Tobii-eye tracker 5. J'ai recherché parmi les logiciels de suivi oculaire avec une webcam, car on peut facilement se procurer des webcams.

Il existe quelques logiciels gratuits de suivi des yeux :

- xLabs
- GazePointer
- MyEye
- PyGaze
- GazeParser

Mais parmi la liste exhaustive, de nombreux logiciels ne sont pas utilisables : il n'ont pas de documentation, on ne peut pas analyser les données produites, ne sont plus mis à jour depuis longtemps ...

J'ai testé PyGaze [8], une boîte à outils open source pour le suivi oculaire en *Python publié par trois chercheurs (de l'Université d'Oxford, de l'Université d'Aix-Marseille et de l'Université d'Utrecht). Mais malgré de nombreux essais, je n'ai pas réussi à faire fonctionner les scripts de démonstrations. Donc j'ai abandonné ce framework.

Finalement, mes tuteurs ont pu passer la commande de la *Tobii-eye tracker 5 le 25 novembre et on l'a reçu le 5 décembre.

8 Conclusion

Rappelons ce qui a été fait :

- Le développement de 2 interfaces graphiques.
- La gestion "backend" de la base de données.
- L'encapsulation de l'application dans un *Conteneur.

Au niveau du travail qui me reste à faire, il faut encore implémenter la liaison entre la *Tobii-eye tracker et l'interface graphique, compléter la deuxième interface graphique. Et enfin faire des tests en conditions réelles.

Pour conclure, ce projet est représentatif d'un problème complexe tels ceux qu'on pourrait rencontrer en entreprise et de mettre en perspective les notions vues durant ma formation à l'*Enssat. Cela m'a donné un aperçu du travail d'ingénieur, surtout au niveau de l'acquisition de nouvelles compétences.

9 Bibliographies

Références

- [1] API Tobii Python <https://developer.tobiipro.com/python/python-sdk-reference-guide.html> (2022).
- [2] Licence Tobii <https://developer.tobii.com/wp-content/uploads/2021/01/Tobii-Tech-Getting-Started-SDLA-29-Sept-2020FINAL.pdf> (29 Septembre 2020).
- [3] Tobii Developer Zone Unity <https://developer.tobii.com/pc-gaming/unity-sdk/api-overview/> (2022).
- [4] Tobii eye-tracker <https://www.tobii.com/products/eye-trackers> (2022).
- [5] What's the difference between Tobii Eye Tracker 4C and 5? <https://help.tobii.com/hc/en-us/articles/360008539058-What-s-the-difference-between-Tobii-Eye-Tracker-4C-and-5-> (1 decembre 2022).
- [6] Documentation Tkinter <https://docs.python.org/fr/3/library/tk.html> (7 decembre 2022).
- [7] Documentation Qt <https://doc.qt.io/> (2022).
- [8] Documentation PyGaze <http://www.pygaze.org/documentation/> (2014).