

Application pour les enfants ayant des besoins spéciaux

Mémoire de recherche réalisé par :

Edmond LATIFI

David ROCH

Sous la direction de :

Alexandros Kalousis

Carouge, le 14 janvier 2022

**Master Science de l'information
Haute École de Gestion de Genève (HEG-GE)**

Remerciements

Nous souhaitons remercier notre Directeur de mémoire de recherche, Monsieur Kalousis Alexandros ainsi que Madame Gregorova Magda, qui nous ont soutenu et suivi durant ce travail.

Pour finir, nous voulons remercier tous nos proches, famille et amis qui ont su nous soutenir en participant aux différentes expériences et nous aider à la correction de ce travail

Résumé

L'une des difficultés auxquelles sont confrontés certains enfants ayant des besoins spéciaux (par exemple, avec des troubles du spectre autistique, la dyspraxie) est leur capacité limitée à concentrer leur regard dans différentes directions rapidement et spontanément. Cela entrave considérablement leur capacité à interagir naturellement avec leur environnement à la fois socialement et physiquement. L'interaction sociale humaine repose fortement sur le contact visuel fournissant des indices importants en percevant et en transmettant des informations aux autres (par exemple, les émotions). Des mouvements oculaires limités, lents et imprécis entraînent le risque de collisions, d'instabilité des mouvements et de niveaux de stress accrus causés par des mouvements inattendus. Lorsqu'elle est identifiée assez tôt, la flexibilité du regard peut être améliorée grâce à une thérapie appropriée (exercices). Le but de ce projet est d'étudier les options afin de développer une application pour soutenir une telle thérapie.

Mots clés : Open-source, PyGaze, Ogama, MagicLeap, GazePointer, Autisme, Thérapie, eye-tracker, eye-tracking.

Table des matières

Remerciements	i
Résumé	ii
Liste des tableaux	v
Liste des figures	vi
1. Introduction	1
1.1 But	1
1.2 Population	2
1.2.1 Les utilisateur-trice-s tests.....	2
1.2.2 Les utilisateurs finaux.....	2
1.3 Equipe	3
1.4 Environnement	3
1.4.1 Outils utilisés	3
1.4.2 Matériel.....	3
1.5 Méthodologie	4
1.5.1 Scrum	4
1.5.2 Sprint.....	5
1.5.3 Calendrier.....	6
2. Risques identifiés	6
2.1 Définition de l'échelle utilisée	6
2.2 Tableau des risques	7
2.3 Matrice des risques	8
3. Mise en pratique	8
3.1 Etat de l'art	8
4. Développement	10
4.1 Oculométrie	10
4.2 Limites de l'oculométrie	10
4.2.1 Justesse et précision.....	11
5. Les Expériences	11
5.1.1 PyGaze.....	12
5.1.2 MagicLeap.....	13
5.1.3 GazePointer	13
5.1.4 Ogama.....	13
5.2 Précision des bibliothèques	14
5.2.1 1 ^{ère} partie	19
5.2.2 2 ^{ème} partie	19
6. Conclusion	20

6.1 Critique	21
Bibliographie	22
Annexe 1 : Planning	17

Liste des tableaux

Tableau 1 : Tâches des stories.....	5
Tableau 2 : Les risques.....	7
Tableau 3 : Matrice des risques.....	8
Tableau 4 : Participant-e-s à la première expérience	14
Tableau 5 : Distances avec PyGaze.....	16
Tableau 6 : Temps en secondes et millisecondes	19

Liste des figures

Figure 1 : Type de population	2
Figure 2 : Méthodologie Scrum.....	5
Figure 3 : Skyle 2 pour iPad Pro	9
Figure 4 : Exemple de pupil tracking.....	12
Figure 5 : Environnement de test.....	15
Figure 6 : Représentation virtuel de l'eye-tracker	15
Figure 7 : Comparaison des librairies	17
Figure 8 : Calcule des distances d'erreur	17
Figure 9 : Calcule de précision	18
Figure 10 : Modélisation de la solution étudiée par l'Université de Georgie et l'Institut Max-Planck d'informatique.....	20

1. Introduction

Notre projet s'inscrit dans une problématique encore non résolue aujourd'hui. En effet, il est question de développer une application aidant les thérapeutes dans leur suivi d'enfant âgé de 3 à 6 ans et ayant des besoins spécifiques (troubles du spectre autistique, la dyspraxie, etc.). Plus précisément, l'application sur tablette permettra d'évaluer la capacité visuelle de l'enfant de manière ludique, c'est-à-dire à l'aide de jeux directement intégrés dans l'application. Ensuite, cela permettra aux thérapeutes spécialisés d'adapter leur thérapie.

1.1 But

Comme il a été cité dans l'introduction, l'objectif est de développer une application répondant aux attentes de la thérapeute mandataire. Le projet a donc été divisé en deux groupes pour distinguer les différentes parties :

1. L'équipe s'occupant de la partie théorique en lien avec la thérapeute.
2. L'équipe de développement travaillant directement sur l'application.

L'objectif prioritaire de l'équipe de développement est donc l'application en elle-même. Bien évidemment, cet objectif comporte d'autres étapes qui seront tout aussi importantes comme le test des différents outils déjà existants pour le *eye-tracker*. Le projet dans sa totalité peut donc se résumer en trois grands axes :

1. Recherche des technologies existantes ;
2. Développement ;
3. Tests techniques et utilisateurs.

Les questions auxquelles nous tenterons donc de répondre sont :

Le développement de l'application est-il réalisable ?

- Si oui :
 - Sur quel(s) support(s) (hardware) et sous quel(s) système(s) d'exploitation ?
 - L'application en question répond-elle aux exigences de la thérapeute ?
- Si non :
 - Pour quelle raison ?
 - Existe-t-il une alternative ?

Plus précisément, allons-nous nous contenter d'un seul type d'hardware (par exemple exclusivement Apple ou Android), un seul type de système d'exploitation ? Ou à l'inverse, allons-nous tenter de faire une application accessible sur plusieurs plateformes ? Ces questions seront répondues dans les phases initiales de notre projet avec les différents essais des librairies existantes. Pour cela, il sera question de définir les critères de tests aux préalables qui seront appliqués à toutes les librairies et nous permettront ainsi de trouver la solution la plus adéquate pour l'implémentation de notre application.

1.2 Population

Nos utilisateurs se divisent en deux groupes :

- Les utilisateur-trice-s tests ;
- Les utilisateur-trice-s finaux.

1.2.1 Les utilisateur-trice-s tests

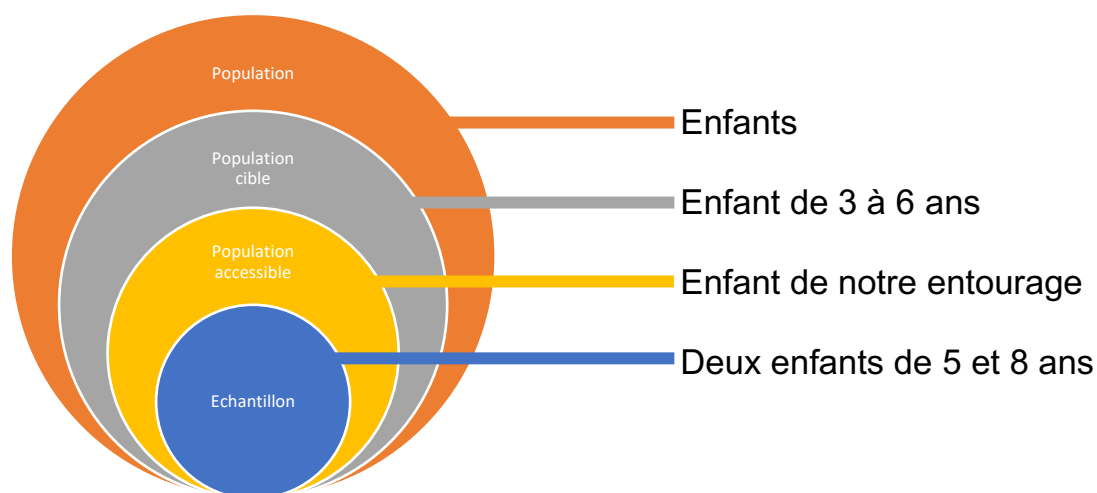
Les utilisateur-trice-s tests seront les personnes qui essaieront l'application dans sa phase de développement. En effet, notre rôle sera de trouver un maximum de personne, ici des enfants entre 3 et 6 ans idéalement, qui nous permettront d'apporter des améliorations à l'application. La population cible étant très difficile à atteindre, nous avons prévu de demander à nos entourages respectifs de faire les essais ainsi que nous, les développeurs de l'application.

1.2.2 Les utilisateurs finaux

Comme il a été dit précédemment, notre application s'adresse aux enfants ayant des besoins spécifiques (troubles du spectre autistique, la dyspraxie, etc.). Ils seront donc les utilisateurs finaux.

La thérapeute ne sera pas l'utilisatrice direct de l'application mais l'utilisera comme outil. Nous pouvons donc la considérer comme une utilisatrice.

Figure 1 : Type de population



Projet de Recherche, D. Roch & E. Latifi

1.3 Equipe

Comme il a été cité précédemment, le projet est divisé en deux équipes. Les deux équipes étaient sous la supervision de Magda Gregorova durant le semestre de printemps, puis c'est Alexandros Kalousis qui a repris la gouvernance du projet.

Concernant notre binôme travaillant sur le projet de développement directement, nous, David Roch et Edmond Latifi, partageons les mêmes tâches de manière globale. Il s'agissait de fournir la documentation ainsi que le développement d'une solution potentielle.

1.4 Environnement

Dans ce chapitre, nous reviendrons sur l'environnement du projet. Plus précisément, nous expliquerons les outils utilisés en ce qui concerne le logiciel et le matériel physique.

1.4.1 Outils utilisés

Pour ce projet, nous utiliserons GitLab ([disponible ici](#)). Cette plateforme permet de partager les différents documents, de fixer des échéances au sein de l'équipe et d'attribuer des tâches à chaque membre. Le répertoire est accessible au membre de l'équipe ainsi qu'au Product Owner.

Pour la communication, nous utilisons Slack¹ pour échanger avec le Product Owner. Les rendez-vous mensuels après chaque Sprint se feront idéalement en présentiel.

Pour ce qui est de la communication avec l'autre groupe, nous utiliserons WhatsApp².

Nous avons à disposition plusieurs supports pour la littérature tels que :

- ScholarVox : c'est une bibliothèque numérique qui propose des e-books.
- ArODES : un répertoire de publications de chercheurs.
- Rero explore : c'est le réseau des bibliothèques de Suisse occidentale pour y trouver des articles, livres, actes de conférence, recensions, etc.

1.4.2 Matériel

Concernant le matériel physique (« *hardware* » en anglais) utilisé, voici les caractéristiques :

- Ordinateurs : Apple MacBook pro MID 2015 13;
- Tablet: iPad Air & iPad Retina;
- Caméra: Logitech StreamCam, intégrée au MacBook et Tobii system.

¹ <https://slack.com/intl/fr-ch/>

² <https://www.whatsapp.com/>

1.5 Méthodologie

1.5.1 Scrum

Pour ce travail nous allons utiliser SCRUM : une méthode agile organisée en cycles de développements itératifs et incrémentaux appelés des Sprints et auxquels le mandant participe de façon active. Dans ce cas de figure, le centre de gravité n'est pas le projet en lui-même, mais bel et bien le résultat. Nous avons décidé de ne pas utiliser une méthode de projet en Waterfall car son cadre est très rigide et ne nous permet pas d'adapter nos recherches en cours de route. La méthode SCRUM s'adapte parfaitement à ce projet de recherche. En effet, dû au caractère exploratoire de notre tâche, il est difficile de prévoir à l'avance les résultats des recherches, il est donc important pour nous de travailler de manière itérative. Nous nous baserons sur les résultats de chaque itération pour adapter les recherches qui en découlent et définir les objectifs du prochain Sprint.

Les particularités de cette méthode sont les suivantes :

1. Il faut définir trois rôles :
 - Le Product Owner : il est représenté par le mandant, Madame Magda Gregorova dont le rôle est de valider les fins de Sprint et définir les spécifications.
 - L'équipe :
 - David R. : Développement
 - Edmond L. : Développement
 - Le SCRUM Master : Edmond sera le SCRUM Master de ce projet. Son rôle sera d'observer un bon suivi de la méthode, que les délais soient respectés et que les outils de collaboration soient utilisés de manière optimale.
2. Deux types de rendez-vous sont organisés :
 - Daily Meeting : il est important que les membres de l'équipe se réunissent régulièrement si ce n'est quotidiennement afin de se tenir informé de l'avancement de chacun. Dans notre cas, comme le projet est prévu un jour par semaine, nous faisons les réunions à ce moment.
 - Sprint Review : ce rendez-vous nous permet, à la fin de chaque itération, d'évaluer les tâches effectuées et d'obtenir un feedback du Product Owner. Cette réunion nous permettra de recalibrer la difficulté des tâches ainsi que l'estimation du temps.

Panos Fitsilis, [lien web](#)

1.5.2 Sprint

Dans cette section, nous retrouvons nos tâches (sprints)

Figure 2 : Méthodologie Scrum

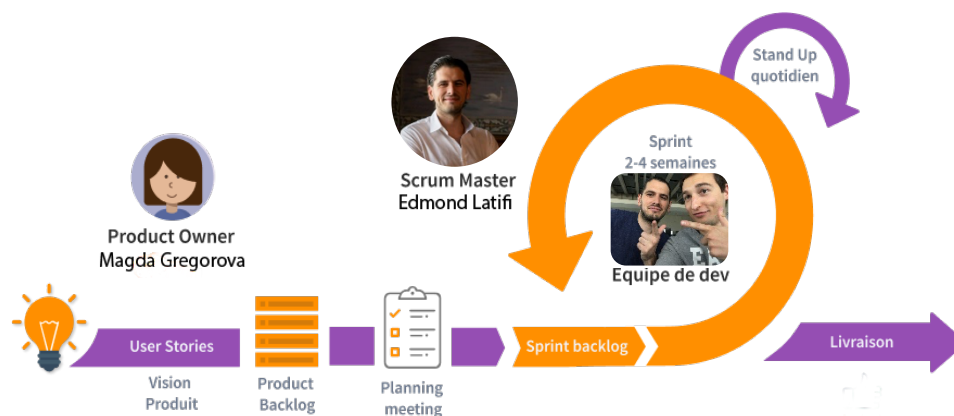


Tableau 1 : Tâches des stories

ID Story	Tâches
1	<ul style="list-style-type: none"> • Répertorier les librairies eye tracking • Document Planification • Mise en place de l'environnement de travail
2	<ul style="list-style-type: none"> • Recherche globale sur le sujet • Devis de recherche
3	<ul style="list-style-type: none"> • Contraintes matériels utilisés • Document Data management plan
4	Choix des librairies intéressantes
5	Test globale des librairies choisis
6	<ul style="list-style-type: none"> • Implémentation librairie n°1 • Implémentation librairie n°2
7	<ul style="list-style-type: none"> • Implémentation librairie n°3 • Choix de la librairie
8	Code

9	Code
10	Phase de test
11	Modifications
12	Finalisation du projet

1.5.3 Calendrier

Voir annexe N°1

2. Risques identifiés

Avant le début du projet, nous avons identifié les principaux risques ainsi que les différentes stratégies de mitigation. Dans le tableau ci-dessous, nous avons une description du risque avec sa probabilité de survenir, son degré d'impact sur le projet ainsi que la solution apportée pour minimiser sa probabilité et/ou son impact.

2.1 Définition de l'échelle utilisée

Échelle	Probabilité	Impact
1	Inexistante	Ennui inexistant
2	Basse	Ennui léger
3	Modérée	Problème modéré
4	Élevée	Problème majeur
5	Certain	Échec du projet

2.2 Tableau des risques

Tableau 2 : Les risques

ID	Description	Probabilité	Impact	Solution
1	Une communication difficile avec le groupe (Diogo B. & Bruno B.).	1	4	Nous avons d'abord défini un support de communication : Groupe sur WhatsApp. Tous nos documents concernant le projet sont visibles parmi nos deux groupes. À la fin de chaque sprint, nous nous réunissons pour discuter des tâches et ainsi garder une cohésion avec l'autre groupe et une synchronisation des tâches.
2	Après le test des bibliothèques, aucune d'entre elles ne correspondrait à nos attentes.	2	4	Nous n'arrivons pas à satisfaire exactement l'objectif final de l'application
3	La technologie hardware n'est pas encore existante.	2	5	Il n'y a pas de solution et le projet ne peut se faire.
4	Manque de matériels à notre disposition.	2	2	Nous pouvons faire une demande auprès de Mme Gregorova pour un emprunt de matériel. Nous avons d'ores et déjà contacté notre entourage pour leur emprunter certains périphériques disponibles : iPhone X, iPad pro, Android One plus, etc.
5	Nos compétences dans un langage donné ne sont pas assez développées pour implémenter l'application dans les temps.	2	4	Nous allons procéder à une évaluation des différents langages et décider lequel est le plus adapté.
6	Une fonctionnalité déjà implémentée par un tiers nous demande d'apprendre un nouveau langage.	3	2	Tous les langages ont plus ou moins les mêmes concepts, mais si besoin, nous n'utiliserons pas la fonctionnalité.

Nous n'avons pas trouvé de solutions sur le marché qui corresponde à toutes les attentes de la thérapeute. À titre d'exemple, le produit qui se rapproche le plus de notre solution idéale serait le Skyle 2 proposé par l'entreprise Inclusive Technology en Grande-Bretagne.

Figure 33 : Skyle 2 pour iPad Pro



(Inclusive Technology, [site web](#))

Voici les caractéristiques de l'appareil en question :

- Modèle de compatibilité : iPad Pro ;
- Fonctionnalité de *eye-tracking* ;
- Possibilité de connecté des outils externes (bouton par exemple) ;
- Prix : 2'196.00 CHF ;
- Il faut ajouter à cela l'iPad Pro : 1198.00 CHF ;
- **Total : 3'394.00 CHF.**

De ce fait, en accord avec notre mandante, nous nous sommes dirigés vers une solution de développement complète.

4. Développement

4.1 Oculométrie

Selon Dre Mélanie Becker³, « l'oculométrie est ainsi très utilisée pour décrire et étudier les processus cognitifs de base en jeu lors de la lecture, ou lors d'autres types de traitement de l'information. Elle constitue, par conséquent, un outil prometteur pour le suivi des processus cognitifs et des stratégies d'apprentissage. Même si, il faut le noter, cette méthode est encore jeune dans ce domaine ». Pour cela, l'oculométrie se fait via une caméra qui, de manière générale, utilise la technologie infrarouge et permet ainsi de calculer les différentes informations liées au regard. Nous distinguons trois méthodes d'oculométrie :

1. **Système fixe** : c'est une caméra infrarouge indépendante qui enregistre directement le regard.
2. **Système mobile** : il s'agit d'une paire de lunettes que l'utilisateur-trice met directement sur lui. Ces lunettes sont équipées de caméras miniatures qui enregistre les mouvements de l'œil de l'individu qui les porte.
3. **Logiciels** : cette dernière, que nous utiliserons dans notre projet, permet via un logiciel de détecter et enregistrer le regard.

L'intérêt pour l'utilisation des informations sur le regard est croissant, non seulement comme instrument de recherche, mais aussi pour améliorer notre interaction quotidienne avec les ordinateurs. Les applications et les techniques d'interaction basées sur le regard vont de la simple saisie du regard comme le pointage, les gestes du regard ou des applications qui utilisent le regard pour déduire des intentions de l'utilisateur-trice.

4.2 Limites de l'oculométrie

Pour cette phase du projet, nous avons dû comprendre les limites de l'*eye-tracking* en dehors des laboratoires dans lequel il est rigoureusement contrôlé. Pour cela, nous avons pour objectif de développer une application assez robuste pouvant être utilisée par un large éventail d'utilisateurs et dans divers environnements. À cette fin, nous avons recueilli des données de suivi oculaire auprès de 10 personnes différentes en utilisant quatre traqueurs distincts et dans deux conditions d'éclairage. Nous avons constaté que la précision peut varier énormément entre l'utilisateur ou utilisatrice, et les points cibles à l'écran. Nous avons observé qu'il y avait des différences entre l'éclairage, les trackers eux-mêmes et les régions de l'écran. Avec l'aide de différentes sources citées dans la bibliographie, nous avons pu déterminer la taille appropriée des cibles.

Au vu de ce qui a été dit précédemment, nous en avons déduit que toutes applications telles que la nôtre devraient répondre à ces questions élémentaires :

- Quelle est la partie de l'écran la plus efficace à utiliser ?
- Quelle précision pouvons-nous attendre ?
- Quelle est la taille minimale utilisable d'un élément à l'écran ?
- Quelle est la distance à avoir entre l'écran et l'utilisateur/utilisatrice ?

³ <https://www.reseau-canope.fr/agence-des-usages/loculometrie-perspectives-pour-une-meilleure-conception-des-interfaces-pour-lapprentissage.html>

Actuellement, peu de recherches ont été faites pour démontrer la faisabilité lors d'un développement sur une application et qu'elles en sont les limites. Pour utiliser un *eye-tracker* open source dans le cadre de notre recherche, nous devons comprendre les caractéristiques et les limites du suivi oculaire dans la pratique.

4.2.1 Justesse et précision

La qualité d'un *eye-tracker* réside dans sa précision ainsi que la justesse de la mesure du regard. Mais avant d'aller plus loin, il est nécessaire de définir ces deux termes qui sont mesurés de manière séparée :

Justesse : Il s'agit de comparer la distance entre le point que l'on souhaite atteindre avec le regard avec le point capturé par l'*eye-tracker*.

Précision : Celle-ci mesure la dispersion du regard lorsque ce dernier fixe un point.

De plus, lors de la mesure du regard, le logiciel se base sur le reflet de la cornée ainsi que la pupille. Cette méthode est d'ailleurs utilisée dans la majorité de cas concernant la mesure du regard. De ce fait, plusieurs types d'interférence peuvent subvenir lors des différents essais. Voici une liste non exhaustive des éléments à prendre en considération :

- Les paires de lunettes ;
- Le maquillage prononcé au niveau des yeux ;
- Les lentilles de contact ;
- La luminosité utilisée ;
- La résolution de la caméra ;
- La focalisation de la caméra.

5. Les Expériences

Pour la phase d'initialisation qui nous permet de nous situer, nous avons répertorié un certain nombre de bibliothèques gratuites permettant le suivi du regard. Nous avons rapidement constaté qu'il existe des centaines de projets⁴ disponible sur Github permettant le suivi de l'œil ainsi que les analyses des informations obtenus. Une majorité de ces projets ont été développés pour des utilisations bien précises tel que savoir la direction du regard grâce aux yeux et à l'orientation du visage, Le suivi de regard pour la lecture d'un document mais dans le cadre de l'utilisation des casques de réalité virtuelle. Cependant un grand nombre d'entre eux ne sont plus mis à jour et datent facilement de cinq à dix ans.

Au vu du temps à disposition pour notre travail de recherche et sur conseil également de M. Gregorova, il était entendu de ne pas trop se disperser et essayer de se focaliser sur certains d'entre eux. Nous avons prédéfini certains critères que devait avoir les bibliothèques pour les utiliser.

⁴ Liste des projets open source sur Github : <https://awesomeopensource.com/projects/eye-tracking>

Voici les critères que nous avons défini :

- Régulièrement maintenu ;
- Utilisable avec Python, C#, Java ou JavaScript;
- Fourni avec suffisamment en documentation ;
- Gratuit et sans frais supplémentaire.

Nous avons explicitement omis le langage Swift – langage de programmation crée par Apple pour développer des applications sur ses systèmes d’exploitation – de notre liste, car pour l’utilisation de certaines fonctionnalités nous devons avoir un compte développeur payant sur leur plateforme.

Nous avons retenu quatre principaux *eye-trackers* :

- PyGaze
- MagicLeap
- GazePointer
- Ogama

5.1.1 PyGaze

Développé en Python et contenant une excellente documentation, c’est une des bibliothèques qui a le plus retenu notre attention pour ce projet. Pygaze n’est pas considéré comme une bibliothèque, mais un *toolbox* qui est entièrement open-source et dédié 100% au eye-tracking. Il regroupe un ensemble de plusieurs petits packages tels que PyGame, PsychoPy, pylink, SensoMotoric Instruments, etc. L’implémentation de PyGaze se fait très facilement grâce à la marche à suivre sur GitHub ou directement sur le site internet de PyGaze⁵. Il dispose également d’un grand nombre d’exemples⁶. Deux projets découlant de PyGaze ont été créés *PyGaze Analyser*⁷ et *Webcam eye-tracker*⁸.

Figure 4 : Exemple de pupil tracking



Webcam library (pyGaze) https://youtu.be/KO_4KYosxVs

⁵ Installation de PyGaze : <http://www.pygaze.org/installation/>

⁶ Exemples de petits projet utilisant PyGaze : <http://www.pygaze.org/documentation/examples/>

⁷ <http://www.pygaze.org/2015/06/pygaze-analyser/>

⁸ <http://www.pygaze.org/2015/06/webcam-eye-tracker/>

5.1.2 MagicLeap

Il s'agit d'un package qui peut être utilisé avec Unity. Concernant Unity, c'est un moteur de jeu, permettant de créer des jeux vidéo sur n'importe quel plateforme et de manière simplifier. En effet, la quasi-totalité de toutes les contraintes physiques que nous pouvons trouver dans un jeux vidéo sont déjà codés et prête à l'emploi. Pour cela, il suffit d'utiliser des méthodes préconstruites. De plus, il existe une licence gratuite lorsque l'outil est utilisé dans un projet personnel. Un bon nombre de jeux vidéo tel que Cities : Skylines, Among Us, Call of Duty : Mobile, ont été développés avec Unity qui pour rappelle, supporte le langage C#. MagicLeap est un package qui est principalement utilisé pour créer des jeux vidéo en réalité augmentée. Dans notre cas, nous utilisons la [fonction de eye-tracking](#) disponible dans ce package.

5.1.3 GazePointer

GazePointer est un logiciel intégrant plusieurs fonctionnalités, comme la possibilité de pouvoir faire bouger le curseur avec les yeux. Pour fonctionner, il suffit de télécharger le logiciel et d'avoir webcam connectée. Nous avons trouvé ce logiciel intéressant car il permet de pouvoir effectuer des tests rapidement sans devoir coder en profondeur et de pouvoir le comparer avec les librairies que nous voulions utiliser.

5.1.4 Ogama

Il s'agit d'un projet open source créer par Adrian Voßkühler pendant ses études à l'université. Malheureusement, le projet n'est plus tenu à jour depuis 2016, mais les personnes intéressées pouvaient reprendre le projet et continuer le maintien de cette dernière. Cependant, depuis 2016, seule une mise à jour a été faite et cela en mars 2021.

Ce qui a attiré notre attention sur ce projet, c'est qu'il permet d'enregistrer et d'analyser les données des mouvements des yeux ainsi que de la souris. Ses principales fonctionnalités comprennent le pré traitement et le filtrage des données relatives au regard et à la souris, la définition de zones d'intérêt, le calcul de la saillance⁹ et le calcul de la distance de Levenshtein¹⁰. Les données fournies peuvent être utilisées directement avec différents logiciels statistiques.

⁹ <https://halshs.archives-ouvertes.fr/halshs-00690831/document>

¹⁰ Algorithme de Levenshtein https://fr.wikipedia.org/wiki/Distance_de_Levenshtein

5.2 Précision des librairies

Premièrement, nous avons contacté 10 participant-e-s qui ne sont autres que nos proches en essayant d'avoir le plus de variétés possibles dans les profils.

Tableau 4 : Participant-e-s à la première expérience

N°	Age	Sexe	Lunettes	Spectre de l'autisme
1	28	M	Non	Non
2	30	M	Non	Non
3	46	F	Non	Non
4	18	M	Non	Non
5	55	F	Oui	Non
6	58	M	Non	Non
7	8	M	Oui	Non
8	10	F	Non	Non
9	5	F	Non	Non
10	10	M	Non	Non

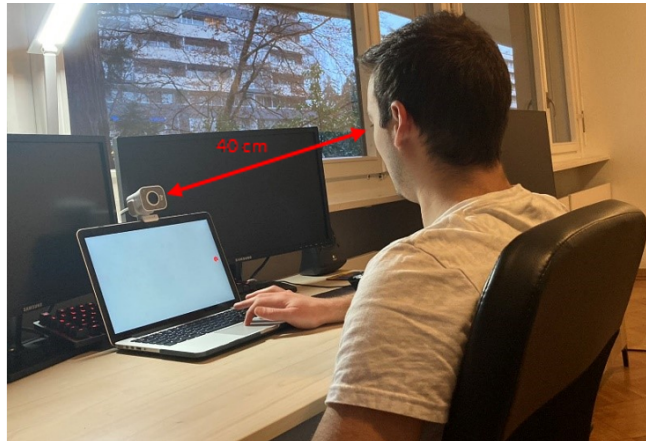
Pour notre première expérience, nous avons défini quelques critères bien spécifiques selon les recommandations de l'article scientifique : *Toward Everyday Gaze Input: Accuracy and Precision of Eye Tracking and Implications for Design*.¹¹. L'utilisateur-trice doit regarder un point rouge de 50 pixels de rayon sur un écran 13" et une résolution de 2560 x 1600 pixels. Au total, cinq points seront affichés. Voici la position centrée des cinq points, les coordonnées étant différentes en programmation qu'en mathématique. En effet, les coordonnées (0,0) ne se situe pas au milieu de l'écran, mais tout en haut à gauche de ce dernier.

1. (1280, 800) au milieu
2. (640, 400) en haut à gauche
3. (1920, 1200) en bas à droite
4. (100,1200) en bas très à gauche
5. (2460, 400) en haut très à droite

¹¹ https://www.microsoft.com/en-us/research/wp-content/uploads/2017/01/everyday_eyetracking-1.pdf

Les participant-e-s se plaçaient à environ 40 centimètres de l'écran et la caméra utilisée est une Logitech StramCam permettant d'enregistrer des vidéos 1080p avec 60 images par secondes. Les participant-e-s ont participé deux fois à l'expérience avec les quatre bibliothèques sélectionnées.

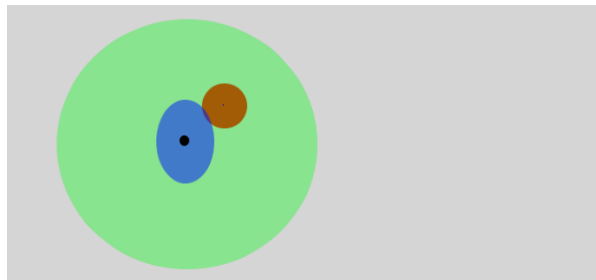
Figure 5 : Environnement de test



Projet de Recherche, D. Roch & E. Latifi

Lorsque le/la participant-e fixe le point correctement, il doit valider sa position de regard avec un clic de la souris. Puis, les coordonnées du regard sur l'écran sont retournées. Sur l'image ci-dessous le point rouge représente le point fixe que le participant-e doit regarder et en bleu, la zone estimée du regard du participant-e par l'*eye-tracker*, avec en son centre le point noir. Ce sont les coordonnées du point noir qui nous sont retournées.

Figure 6 : Représentation virtuel de l'*eye-tracker*



Projet de Recherche, D. Roch & E. Latifi

Nous avons calculé la distance entre notre point fixe y qui est notre point objectif, et notons le centre trouvé par l'*eye-tracking* x .

$$\text{Calcul de distance : } \|x - y\|$$

Si la distance est < 100 alors le point rouge a été touché par le regard.

$$\text{if } \|x - y\| \leq 100 : \\ \text{nbjuste} = \text{nbjuste} + 1$$

Pour chaque point fixe rouge, nous prenons la moyenne des distances des deux expériences

$p = participants$

$r = point\ fixe\ rouge$

$i : [1 \ à \ 10]$ représente les participants

$e : [1 \ et \ 2]$ représente le numéro de l'expérience

$f : [1 \ à \ 5]$ représente les points rouge fixe.

$$moyenne\ d'un\ point\ rouge\ pour\ un\ participant : r_f p_i = \frac{1}{2} \sum_{e=1}^2 \|x_f - y_{e_f}\|$$

Tableau 5 : Distances avec PyGaze

Participants	fixe_1	fixe_2	fixe_3	fixe_4	fixe_5	Moyenne
1	85,0050	98,1896	345,9822	95,5448	38,0938	132,5631
2	97,0000	705,3277	766,1567	84,77312	136,0269	426,1278
3	138,1587	53,4555	817,4500	94,9501	38,0089	228,4046
4	41,8647	79,6353	89,6545	153,8945	202,2589	113,4616
5	596.24354				589,1450	589,1450
6	65,6599	496,6999	598,1379	45,1860	39,4888	249,0345
7		894,6870	390,3790	1569,0000		951,3553
8	130,3790	531,9964	640,1625	98,6010	71,5030	294,5284
9	56.24354	1654,2600	65,1000	29,0150	31,1450	444,8800
10	85,1450	534,6400	40,9870	27,6500	87,1000	155,1044
						<u>358,4605</u>

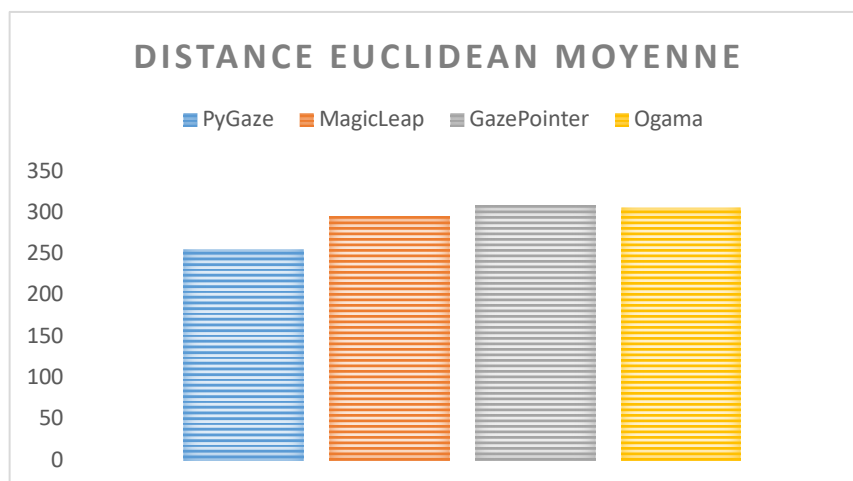
Comme nous pouvons le constater sur le tableau ci-dessus, lorsque nous avons effectué l'expérience avec PyGaze, nous avons rencontré quelques soucis avec nos deux participant-e-s qui portaient des lunettes. En effet, certains points trouvés par l'eye-tracker ne retournaient pas de valeur (pas défini). Nous avons donc retiré ces deux participant-e-s de notre expérience pour la suite.

De plus nous pouvons voir que les points fixe qui étaient le plus aux extrémités de l'écran et celui au centre ont une distance plus petite que les autres.

Pour la comparaison de nos quatre librairies, nous avons retiré les deux participant-r-s portant des lunettes et avons calculé la moyenne de toutes les distances de chaque point pour les 8 participant-e-s.

$$moyenne\ total = \frac{1}{8} \sum_{i=1}^8 \sum_{f=1}^5 \sum_{e=1}^2 \|x_f - y_{i_{e_f}}\|$$

Figure 77 : Comparaison des librairies



Toutes les expériences ont été faites dans un environnement semi-contrôlé, par exemple, la luminosité de la pièce dépendait du moment de la journée et des lampes, la caméra et la distance reste inchangées.

Pour pousser l'expériences un peu plus loin, nous avons gardé les quatre premiers participant-e-s et les deux première librairies (PyGaze et MagicLeap) avec deux caméras différentes ainsi que deux luminosités différentes :

Caméra :

1. Intégré à un MacBook Pro mid 2013
2. Logitech StreamCam

Luminosité :

1. Faible ~ 100 lumen
2. Haute ~1500 lumen

Nous notons E comme étant l'Expérience, C = Camera et L = Luminosité

C_1 = Caméra du MacBook

C_2 = Logitech

L_1 = Luminosité faible

L_2 = Luminosité normal

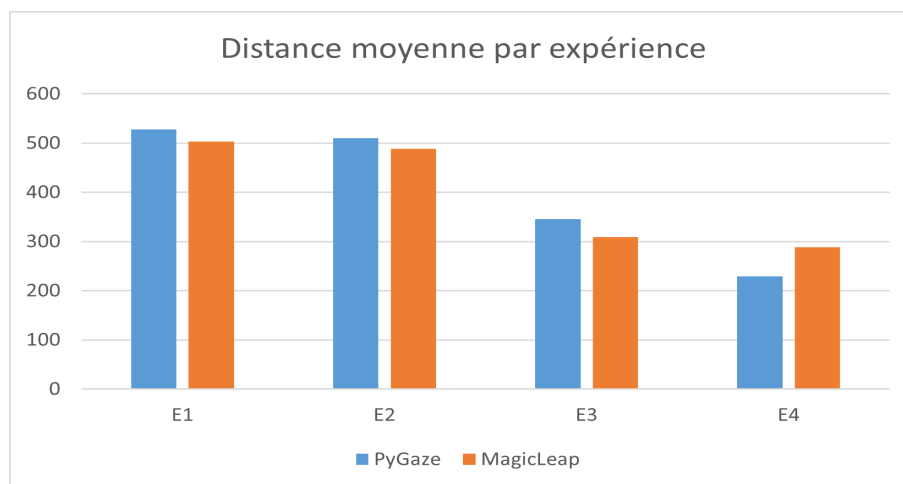
E_1 = C_1 avec L_1

E_2 = C_1 avec L_2

E_3 = C_2 avec L_1

E_4 = C_2 avec L_2

Figure 88 : Calcule des distances d'erreur

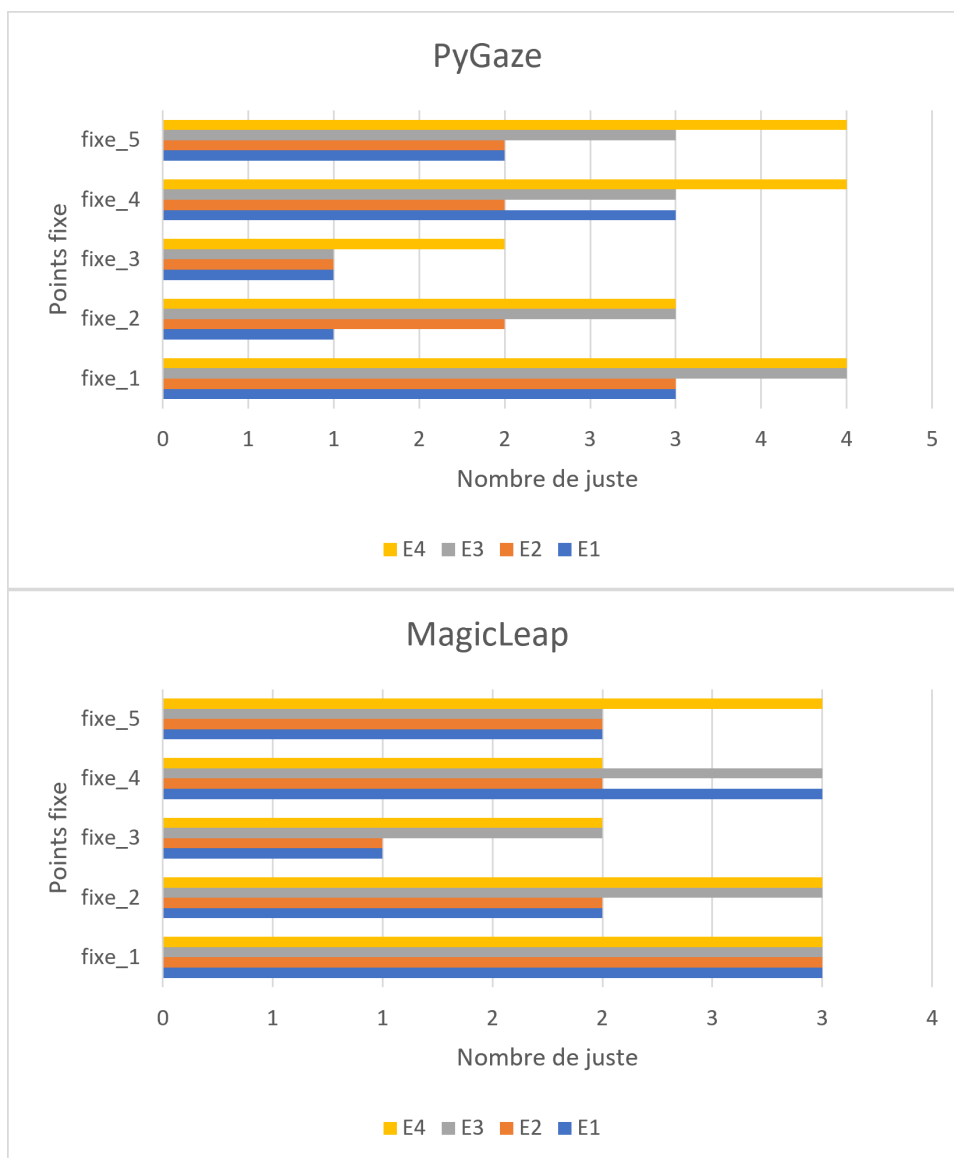


Ce que la figure ci-dessus illustre, c'est que le choix de la caméra à un impact assez significatif sur la précision de notre *eye-tracker*. De plus, nous pouvons constater que la luminosité influence également l'efficacité du capteur de regard.

Pour notre projet, il convient aux utilisateur-trice-s finaux de ne pas devoir dépendre d'une caméra. Par conséquent, il doit être possible d'avoir un bon taux de précision avec tout type de caméra. Pour cela, il faudrait manipuler plusieurs paramètres comme revoir la taille des points fixes selon la résolution de l'écran et de la qualité de la caméra. Mais nous n'avons pas eu le temps de nous consacrer sur ce dernier point, car il s'agirait d'un autre travail de recherche à part entière.

Pour les quatre expériences mentionnées plus haut, nous avons regardé le nombre de fois que le *eye-tracker* a capté correctement le regard du participant-e sur le point fixe et nous avons reporté ce graphique. De dernier pourrait bien évidemment être plus explicite avec un nombre plus élevé de participants et davantage de points fixe. Comme décrit sur la figure ci-dessous, les points fixes 1,4 et 5 ont quasiment un sans-faute et la précision en est améliorée avec l'expérience 4.

Figure 99 : Calcul de précision



Pour la dernière expérience, que nous avons fait du 11 au 18 décembre 2021, nous avons redéfinis deux nouveaux points fixes parmi les cinq précédents :

1. (1280, 800) au milieu
2. (640, 100) très en haut à gauche
3. (1920, 1200) très en bas à droite
4. (100,1500) en bas très à gauche
5. (2460, 400) en haut très à droite

Les huit participants ont accepté de passer cette ultime expérience qui se déroule en 2 parties

5.2.1 1^{ère} partie

L'objectif de celui-ci est de regarder la précision du regard avec les nouveaux points définis et dans un environnement contrôlé.

$$\text{Précision} = \frac{\text{nbjuste}}{100}$$

if $\|x - \hat{x}\| \leq 100 \text{ pixels} : \text{nbjuste} + 1$

x = le point fixe

\hat{x} = le point regardé

5.2.2 2^{ème} partie

L'objectif est de regarder en combien de temps le regard se dirige sur le point fixe affiché à l'écran. Comme nous l'avait mentionné Monsieur Kalousis lors de la présentation des posters, il aurait été intéressant de voir quand la personne a regardé le point sur une durée et non pas seulement au moment où l'utilisateur-trice clic sur la souris.

```
from datetime import datetime

# PSEUDO CODE
t0 = datetime.now() # heure, minutes, secondes, millisecondes à un
instant t
#
if eye_tracking - pointFixe <= 100:
    time = datetime.now() - t0
```

Les résultats obtenus pour la première partie sont plutôt concluants avec une précision de 90%. Cependant sur le temps de réaction, comme sur le graphique ci-dessous, il est difficile de conclure si l'eye-tracker PyGaze est utilisable pour la détection de trouble de l'autisme. En effet, pour le neuvième participant-e, les résultats sont assez corrects mais il se peut que le calibrage au début se soit bien fait pour elle ou que sa position était meilleure pour détecter le regard. De ce fait, il faudrait un plus grand nombre d'échantillons et plus de temps pour pouvoir interpréter ses données et les rendre utilisables.

Tableau 6 : Temps en secondes et millisecondes

Participant	fixe_1	fixe_2	fixe_3	fixe_4	fixe_5
1	00:00:02,980	00:00:04,687	00:00:02,111	00:00:04,982	00:00:03,541
2	00:00:03,198	00:00:07,320	00:00:01,227	00:00:03,006	00:00:03,226
3	00:00:02,021	00:00:08,820	00:00:01,776	00:00:04,279	00:00:06,463
4	00:00:02,010	00:00:05,101	00:00:02,363	00:00:02,302	00:00:01,960
6	00:00:02,000	00:00:03,005	00:00:02,189	00:00:05,246	00:00:07,636
8	00:00:06,320	00:00:04,109	00:00:02,799	00:00:02,740	00:00:01,509
9	00:00:01,202	00:00:02,090	00:00:00,899	00:00:00,878	00:00:01,006
10	00:00:02,003	00:00:01,988	00:00:01,159	00:00:01,919	00:00:02,451

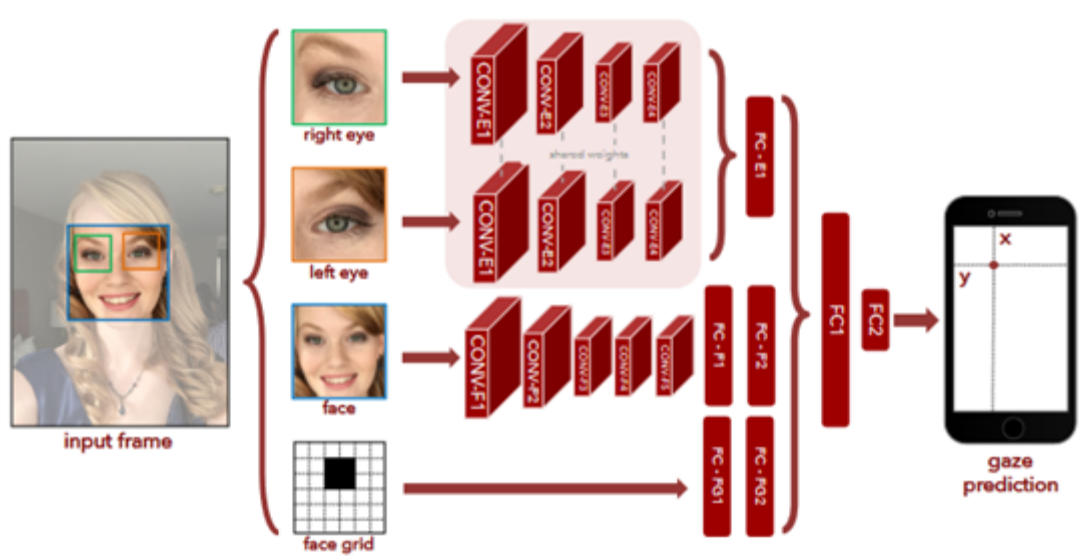
6. Conclusion

Dans le cas de notre projet de recherche et après plusieurs essais, nous concluons que l'utilisation de librairie open source (accès libre) n'est pas possible. En effet, la précision et la justesse n'étant pas assez élevées pour un domaine comme la médecine, la solution n'est donc pas optimale et, par conséquent, utilisable.

Pour augmenter la précision et la justesse des résultats obtenus, certaines études retravaillent « logiciellement » la capture du regard. Cela augmente donc les deux caractéristiques citées précédemment, mais augmente malheureusement la latence qui est un élément important quand la population cible sont les enfants atteints d'un trouble autistique.

Cependant, une solution potentielle que nous avons trouvée en nous renseignant serait le renforcement du capteur grâce à l'intelligence artificielle. En effet, une recherche effectuée entre l'Université de Georgie et l'Institut Max-Planck d'informatique utilisait une base de données fournit par Amazon pour entrainer leur *eye-tracker*. Le but de la recherche était d'utiliser la caméra d'un *smartphone* pour suivre le regard.

Figure 1010 : Modélisation de la solution étudiée par l'Université de Georgie et l'Institut Max-Planck d'informatique



(Eye tracking for everyone, [lien web](#))

Dans le cadre de ce projet, la thérapeute travaillant avec les enfants souhaitait avoir un résultat immédiat. Cela pourrait être intéressant si les résultats de l'utilisation de l'application étaient traités post-utilisation.

Il s'agirait donc d'explorer cette voie dans un second temps du projet pour ainsi proposer un capteur de regard plus robuste et efficace, tout en étant mobile et sans matériel physique supplémentaire.

Une autre alternative serait d'investir dans la solution citée dans le chapitre 3.1. En effet, l'appareil *Skyle 2* proposé par l'entreprise Inclusive Technology a certes un coût relativement élevé mais leur système permet d'avoir au niveau hardware une solution robuste. Il s'agirait donc d'utiliser leur outil en développant de notre côté l'application qui permettra à la thérapeute de l'utiliser dans son cadre précis. L'application en question serait le jeu qui permet d'interagir avec les enfants. Nous aurions donc le côté hardware qui serait externe à notre équipe mais l'application développée en interne.

6.1 Critique

Concernant le projet, une des principales critiques que nous pouvons formuler serait le temps que chaque tâche nous demandait. En effet, certaines tâches en particulier nous demandaient un temps conséquent pour être effectuées de manière optimale. À titre d'exemple, les essais des différentes bibliothèques représentent une grande majorité de notre temps consacré au projet. Pour compléter ce point, le fait d'avoir une demi-journée pour le projet réduisait d'autant plus le temps que nous pouvions passer sur les différentes tâches. Cela nous a montré l'importance d'avoir un plan défini et une phase d'initialisation de projet complète pour éviter ce type de situation.

Autre fait, c'est le manque de données concernant les eye-trackers pour pouvoir les comparer à nos résultats. Il aurait été intéressant, dans notre cas, d'avoir des données de références pour pouvoir ainsi faire des comparaisons de qualité.

Bibliographie

Anagnostopoulos, Vasileios-Athanasios & Kiefer, Peter. (2016). Towards Gaze-based Interaction with Urban Outdoor Spaces. 1706-1715. 10.1145/2968219.2968339.

Drewes, Heiko & De Luca, Alexander & Schmidt, Albrecht. (2007). Eye-gaze interaction for mobile phones. 364-371. 10.1145/1378063.1378122.

Gazenav : Giannopoulos, Ioannis & Kiefer, Peter & Martin, Raubal. (2015). GazeNav: Gaze-Based Pedestrian Navigation. <https://doi.org/10.1145/2785830.2785873>

Toyama, Takumi & Kieninger, Thomas & Shafait, Faisal & Dengel, Andreas. (2011). Museum Guide 2.0 – An Eye-Tracking based Personal Assistant for Museums and Exhibits. In Proc. of Int. Conf. on Re-Thinking Technology in Museums volume 1, 2011.

Palama, A. (2019). La perception intermodale (audio-visuelle) des expressions émotionnelles (joie, colère, peur ou dégoût) chez les bébés âgés de 1 à 6 mois et les enfants âgés de 5 à 10 ans : Apports de l'oculométrie. doi:10.13097/archive-ouverte/unige:123713 Retrieved from <https://nbn-resolving.org/urn:nbn:ch:unige-1237130>

23 mars 2021. Wikipédia [en ligne]. [Consulté le 01.09.2021]. Disponible à l'adresse : <https://fr.wikipedia.org/wiki/Oculom%C3%A9trie>

2020. Apple [en ligne]. 2 mars 2020. [Consulté le 02.09.2021]. Disponible à l'adresse : <https://support.apple.com/fr-ch/HT208108>

2021. PyGaze [en ligne]. [Consulté le 14.07.2021]. Disponible à l'adresse : <https://www.pygaze.org/about>

FEIT, Anna, WILLIAMS, Shane, TOLEDO, Arturo, PARADISO, Ann, KULKARNI, Harish, KANE, Shaun, RINGEL MORRIS, Meredith, 2017. «Toward Everyday Gaze Input: Accuracy and Precision of Eye Tracking and Implications for Design», Microsoft Research[en ligne]. janvier 2017. [Consulté le 15 août 2021]. Disponible à l'adresse : https://www.microsoft.com/en-us/research/wp-content/uploads/2017/01/everyday_eyetracking-1.pdf

Michael Barz, Florian Daiber, and Andreas Bulling, 2016. Prediction of gaze estimation error for error-aware gaze-based interfaces. Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications[en ligne]. 14 mars 2016. [Consulté le 28 août 2021] Disponible à l'adresse: <http://dx.doi.org/10.1145/2857491.2857493>

Krafka, Kyle, KHOSLA, Aditya, KELLNHOFER, Petr, KANNAN, Harini, BHANDDARKAR, Suchendra, MATUSIK, Wojciech, TORRALBA, Antonio, 2016. «Eye Tracking for Everyone», MIT[en ligne]. 2017. [Consulté le 16 novembre 2021]. Disponible à l'adresse : https://gazecapture.csail.mit.edu/cvpr2016_gazecapture.pdf

Palama, A. (2019). La perception intermodale (audio-visuelle) des expressions émotionnelles (joie, colère, peur ou dégoût) chez les bébés âgés de 1 à 6 mois et les enfants âgés de 5 à 10 ans : Apports de l'oculométrie. doi:10.13097/archive-ouverte/unige:123713 Retrieved from <https://nbn-resolving.org/urn:nbn:ch:unige-1237130>

2021. PyGaze [en ligne]. [Consulté le 15.12.2021]. Disponible à l'adresse : <https://www.pygaze.org/about>

Annexe 1 : Planning

	Sprint 0	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8	Sprint 9	Sprint 10	Sprint 11	Sprint 12
15.03.2021													
22.03.2021													
29.03.2021													
05.04.2021													
12.04.2021													
19.04.2021													
26.04.2021													
03.05.2021		1											
10.05.2021													
17.05.2021													
24.05.2021		devis de recherche											
31.05.2021			2										
07.06.2021													
14.06.2021													
21.06.2021													
28.06.2021				DMP									
05.07.2021				3									
12.07.2021					4								
19.07.2021													
26.07.2021						5							
02.08.2021													
09.08.2021							6						
16.08.2021													

23.08.2021								7					
30.08.2021													
06.09.2021									8				
13.09.2021									billet de blog				
20.09.2021													
27.09.2021									8				
04.10.2021													
11.10.2021													
18.10.2021													
25.10.2021													
01.11.2021													
08.11.2021													
15.11.2021													
22.11.2021													
29.11.2021													
06.12.2021													
13.12.2021													
20.12.2021													
27.12.2021													
03.01.2022													12
10.01.2022													Final