Project Onboarding

Date de la dernière révision

Version	Date	Auteur	Description
1.0	04/03/2024	Pierre	Premier Jet

Sommaire :

1 – Récapitulatif et attente du projet

Notre projet consiste à développer une application de traduction des signes en temps réel. Cette application sera conçue pour interpréter les gestes et mouvements des mains en langage des signes et les traduire instantanément en texte ou parole. Dans un second temps, il sera aussi possible de traduire des paroles en langage des signes au travers d'une modélisation et d'une projection de main virtuelle, visible grâce à la réalité augmentée. L'objectif principal est de faciliter la communication entre les personnes sourdes ou malentendantes et les entendants, en offrant une traduction précise et rapide des signes.

Attentes du Projet:

1. **Fonctionnalités Clés**: L'application devra être capable de reconnaître et interpréter un large éventail de gestes et expressions du langage des signes pour assurer une traduction précise.
2. **Temps Réel**: L'aspect crucial du projet est la capacité à traduire les signes en temps réel, offrant ainsi une communication fluide et instantanée entre les utilisateurs.
3. **Interface Intuitive**: L'application devra avoir une interface conviviale et intuitive pour garantir une expérience utilisateur agréable et accessible à tous.
4. **Précision et Fiabilité**: La traduction des signes doit être précise et fiable pour assurer une communication efficace sans erreurs majeures.
5. **Compatibilité Multiplateforme**: Il serait avantageux que l'application soit compatible avec différents appareils (smartphones, tablettes) et systèmes d'exploitation pour toucher un large public.

2 – Répartition des tâches

2.1 – différents pôles

SignIt est composé de deux principales pôles, le pôle intelligence artificielle et le pôle réalité mixte. Chaque pôle comprend une équipe, contenant plusieurs membres. Vous trouverez ici les informations de chaque membre de chaque équipe pour pouvoir vous mettre en relation avec ceux-ci.

Il est nécessaire de savoir que Pierre Ginisty détient aussi le poste de chef de projet. Kyllian Vienne est le scribe durant les réunions. Romain Rosso est responsable communication interne et externe.

2.1.1 – Pôle IA

Le pôle IA contient une équipe composée de :

- Gaillet Aîmane : aimane.gaillet@epitech.eu
- Hamda Salsabil : salsabil.hamda@epitech.eu
- Tomietto Vincent : vincent.tomietto@epitech.eu
- Rosso Romain : romain.rosso@epitech.eu

2.1.2 – Pôle XR

Le pôle XR contient une équipe composée de :

- Loucif Inasse : inasse.loucif@epitech.eu
- Vienne Kyllian : killian.vienne@epitech.eu
- Oukziz Salma : salma.oukziz@epitech.eu
- Ginisty Pierre : Pierre.ginisty@epitech.eu

2.2 – Tâches actuelles

Pour se mettre à jour sur les tâches actuelles, il est nécessaire de prendre connaissance de notre Github. Le lien du github est disponible plus bas dans ce document. De plus, les tâches sont traitées au cours des réunions hebdomadaires, il est primordial d'y assister. Les réunions ont lieu chaque lundi de 10h à 12h (UTC+1).

3 – Plateformes utilisés par l’équipe

Plusieurs plateformes sont utilisés par l’équipe au cours de ce projet.

Nous utilisons principalement Github pour la mise en commun du code, Trello pour la répartition des tâches et pour obtenir une vue d’ensemble pour la gestion de projet.
Et pour finir, Gamma pour le partage de documents et d’informations.

3.1 – Github

Le site assure un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de fonctionnalités et la gestion de tâches pour notre projet.
Lien github : « https://github.com/EpitechMscProPromo2025/T-ESP-800-project_esp800-56335-TLS-SignIt »

3.2 – Trello

Trello est un outil de gestion de projet en ligne, lancé en septembre 2011 et inspiré par la méthode Kanban de Toyota. Il repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement.
Lien Trello : « Notion – The all-in-one workspace for your notes, tasks, wikis, and databases »

3.3 – Gamma

Gamma utilise des algorithmes d’IA sophistiqués pour transformer des prompts textuels en designs de diapositives. Gamma est utilisé durant ce projet pour le partage de documents et d’informations.
Lien Gamma : « https://gamma.app/docs/Organisation-Cahier-des-charges-SIGNIT--v32fih8z8v0gjkr?mode=doc »

4 – Norme de commit

Les mots clés “DOIT” (“MUST”), “NE DOIT PAS” (“MUST NOT”), “REQUIS” (“REQUIRED”), “NE DOIT” (“SHALL”), “NE DOIT PAS” (“SHALL NOT”), “NE DEVRAIT” (“SHOULD”), “NE DEVRAIT PAS” (“SHOULD NOT”), “RECOMMANDÉ” (“RECOMMENDED”), “PEUT” (“MAY”), et “FACULTATIF” (“OPTIONAL”) dans ce document doivent être interprétés comme décrit dans RFC 2119.

1. Les commits DOIT être préfixés par un type, qui consiste en un nom, `feature`, `fix`, etc., suivi d’un côlon et d’un espace.
2. Le type `feature` DOIT être utilisé lorsqu’un commit ajoute une nouvelle fonctionnalité à votre application. ou bibliothèque.
3. Le type `fix` DOIT être utilisé lorsqu’un commit représente un correctif pour votre application.
4. Un scope facultative PEUT être fournie après un type. Un scope est une phrase décrivant une section du code encadrée par des parenthèses, par exemple, `fix (analyseur)):`
5. Une description DOIT suivre immédiatement le préfixe type/scope. La description est une brève description des modifications du code, par exemple, *fix: bug feature loading screen.*
6. Un texte plus long PEUT être fourni après la description courte, fournissant des informations contextuelles supplémentaires sur les modifications du code. Le texte DOIT commencer par une ligne vide après la description.
7. Un pied de page PEUT être fourni une ligne vierge après le texte (ou après la description si le texte manque). Le pied de page DEVRAIT contenir des références de problèmes supplémentaires concernant les modifications du code (telles que les problèmes qu’il corrige, par exemple, `Fixes #13`).
8. Les changements de rupture DOIVENT être indiqués au tout début du pied de page ou de la section du corps de texte d’un commit. Un changement radical DOIT être dans le texte en majuscule `BREAKING CHANGE`, suivi de deux points et d’un espace.
9. Une description DOIT être fournie après le «BREAKING CHANGE:», décrivant la modification de l’API, par exemple, *BREAKING CHANGE: environment variables now take precedence over config files*
10. Le pied de page NE DOIT contenir que «BREAKING CHANGE», des liens externes, des références de publication et d’autres méta-informations.
11. Des types autres que `feature` et `fix` PEUVENT être utilisés dans vos messages de commit.
12. Tous les commits DOIVENT être rédigé en anglais.