



NOM DE VOTRE PROJET :	T.A.R.S
MEMBRES DE L'ÉQUIPE :	THIBAUT ALVOET, MATHILDE JEUNAUX, OLIVIER SERRA
NIVEAU D'ÉTUDE :	TERMINALE GÉNÉRALE
ÉTABLISSEMENT SCOLAIRE :	LYCÉE PRIVÉ SACRE CŒUR
ENSEIGNANT DE NSI :	XAVIER DEWALEYNE

> PRÉSENTATION GÉNÉRALE :

Pouvez-vous présenter en quelques mots votre projet ?

Le projet Tars est un assistant vocal personnalisable semblable à Alexa, Google Home ou encore HomePod transposable facilement sur un Raspberry Pi 4.

Il est possible d'ajuster l'humour avec lequel Tars vous répondra, son sarcasme, ainsi que la longueur de ses phrases, et la voix qu'il prendra pour vous répondre. Cet assistant aura besoin d'un microphone pour vous entendre, et d'une enceinte quelconque pour vous répondre.

Comment est né ce projet ?

Ce projet est né d'une idée de Thibaut, passionné par le numérique et la programmation, souhaitant créer son propre assistant vocal aux moindres coûts, conservant des fonctionnalités similaires à un assistant vocal créé par les grandes entreprises, tout en évitant certains problèmes de confidentialité qui leur sont liés.

Quelle était la problématique de départ ?

Au départ, nous avons souhaité créer un assistant vocal avec une personnalité unique, capable de répondre à des requêtes dans de nombreux domaines, ainsi que d'exécuter quelques tâches, facilement programmables par l'utilisateur s'il le souhaite.

Nous voulions également garantir la sécurité et la confidentialité des données des utilisateurs, contrairement aux GAFAM, connus pour de nombreuses controverses d'écoute et d'analyse des données audio à l'insu de leurs utilisateurs.

Quels sont les objectifs ?

Nous avons pour objectif de créer un assistant entièrement open source en proposant à nos utilisateurs une solution pratique, sûre et efficace, pour accéder à des informations du web ; alliée à une accessibilité travaillée.

Nous souhaitons simplifier la vie quotidienne en fournissant des fonctionnalités pratiques telles que la de traduction de phrases, la recherche de synonymes, de définition, de recette de cuisine, etc... tout en protégeant au maximum la vie privée et la sécurité de nos utilisateurs, en portant une attention singulière sur la collecte, l'utilisation et le stockage d'informations personnelles.

Enfin, nous voulions faciliter son intégration, en utilisant, couplé au Raspeberry Pi 4, des appareils électroniques usuels. En effet, n'importe quelle enceinte (généralement branché en jack) couplé à un microphone USB quelconque fera largement l'affaire.

Nous avons d'ailleurs utilisé, certes, un microphone de bonne manufacture car c'est ce que nous avons, mais un mini microphone USB comme ceux trouvables sur internet, très discret, aurait très bien pu faire l'affaire.

À quels besoins répondez-vous ?

Nous souhaitons offrir à nos utilisateurs une façon pratique et efficace d'accéder à de nombreuses informations et services grâce à de simples commandes vocales en réduisant l'utilisation d'appareils tierces nécessitant d'ouvrir une application, taper une question sur le web, chercher, etc...

Nous rendons notre assistant accessible à un public plus large en proposant des réponses adaptées aux goûts de chacun ; personnalisant la voix, l'humour, le sarcasme et le débit de parole, offrant des interactions plus pertinentes et adaptées à chaque individu.

En mettant en place des solutions afin de protéger la vie privée et la sécurité de nos utilisateurs, en particulier sur ce qui concerne la collecte, l'utilisation et le stockage des

données personnelles, notre assistant vocal possède à ce jour une politique de confidentialité très certainement plus éthique que ses concurrents.

> ORGANISATION DU TRAVAIL :

Pouvez-vous présenter chaque membre de l'équipe et préciser son rôle dans ce projet ?

Thibaut nous a tout d'abord guidé aux travers des différentes tâches du projet. Il a mis en place son architecture générale (développement linux, calls API, différents modules, etc...), a notamment appris à se servir du framework python « Flask » pour gérer les requêtes envoyées par l'utilisateur, ainsi que les templates HTML au format « Jinja2 » y étant liés ; permettant un affichage dynamique des éléments sur la page de configuration de Tars.

Olivier, quant à lui, développeur du son et du bon fonctionnement de ce dernier au sein de Tars, s'est occupé de développer la génération de réponse ; ainsi qu'un système de File, en lien avec notre cours de NSI, nécessaire pour le déroulement correct des phrases de Tars. Cette structure est partagée partout dans le code afin d'éviter les conflits dans l'ordre de diction des phrases. Il s'est de plus occupé du sound design de Tars, créant à partir d'un instrument en ligne les différents sons donnant une identité à Tars.

Pour finir : moi, Mathilde, ai réalisé l'aspect graphique de Tars : le style des différentes pages web (la partie CSS) via l'aide de Thibaut, le design du logo (visible sur son panneau de configuration), le design et l'impression 3D de son boîtier, la documentation, l'écriture, le tournage et enfin le montage de la vidéo.

Comment avez-vous réparti les tâches et pourquoi ?

Nos tâches étant explicitées précédemment ; nous nous sommes organisés de manière que chacun puisse de travailler de son côté. Nous mettions en commun notre avancée du projet lors de nos heures d'étude, en échangeant constamment sur le travail de chacun.

Nous communiquons sur WhatsApp pour se partager toutes les informations et garder le fil des avancées du projet.

Nous avons procédé ainsi pour profiter des qualités de chacun : Thibaut avait déjà de très solides connaissances dans la réalisation de projet informatique, et a donc su nous guider tout en développant ses aspects divers ; Olivier s'est spécialisé dans la gestion et l'agencement des requêtes via la structure objet de File, et j'ai pu, via mes qualités de designeuse et de monteuse, apprendre à utiliser le CSS pour réaliser concrètement les différentes pages web, et découvrir comment créer un modèle 3d imprimable.

Combien de temps avez-vous passé sur le projet ? Avez-vous travaillé en dehors de l'établissement scolaire ?

Nous avons grandement travaillé sur ce projet, depuis que nous sommes inscrits, majoritairement en dehors des cours, que ce soit le soir à l'internat, moi-même (Mathilde) et Thibaut, notamment lorsqu'il m'expliquait comment modéliser le boîtier et comment fonctionne le CSS, mais aussi principalement en dehors des semaines au

lycée : le week-end lors de notre temps libre, et durant les vacances de Février, de longues journées durant (pour la création du code).

Quels sont les outils et/ou les logiciels utilisés pour la communication et le partage du code ?

Nous avons utilisé GitHub pour la publication du code, avec l'aide de Git pour mettre à jour le projet directement en quelques commandes sans devoir importer un part un chaque fichier sur GitHub. Entre nous, nous avons partagé notre avancé via clé USB, ce que nous trouvions plus simple dans certains cas.

LES ÉTAPES DU PROJET :

Présenter les différentes étapes du projet (de l'idée jusqu'à la finalisation du projet)

Nous avons connu beaucoup d'étapes, notamment celles-ci :

1. Recherche de l'idée.

Tout d'abord, nous nous sommes réunis au lycée pour discuter de l'éventualité de participer à ce projet. Une fois tout le monde au point sur nos ambitions, nous avons réfléchi à une idée qui pourrait nous plaire. C'est là qu'un souvenir de Thibaut est survenu : il avait quelques mois auparavant eu l'idée de ce projet, suite à son visionnage d'Interstellar. Cette idée nous a grandement plu, à Olivier et moi. Nous nous sommes alors embarqués pour quelques mois, mois tourmentés par le travail et ses découvertes, pour un résultat dont nous sommes très fiers.

2. Premiers essais et approches de l'IA, remise en question de certains points.

Lors des premiers jours, Olivier et Thibaut ont d'abord essayé et découvert quelques bibliothèques Python qui se sont avérés très utiles, notamment SpeechRecognition pour la reconnaissance vocale, celle de ChatGPT pour la génération des réponses, et plus tard celle d'Elevenlabs pour l'utilisation des voix humanoïdes. Nous avions à ce moment pour but d'établir une connexion Peer to Peer entre le Raspberry Pi et le téléphone pour les faire communiquer, puis après plusieurs recherches, avons compris que connecter les deux appareils au même réseau serait bien plus simples, logiques et efficaces.

3. Développement de l'interface (partie graphique).

Une fois le serveur local Flask rapidement mis en place par Thibaut, j'ai pu commencer à designer, sur ses pages HTML, le CSS qui rendra l'expérience utilisateur plus agréable. Nous avons alors fait différent choix cruciaux pour l'identité de Tars, tels que celui de la police d'écriture utilisée, des couleurs, ainsi que d'autres aspects, comme l'agencement global de chaque page.

4. Développement de l'interface (partie fonctionnelle).

Une fois fait, Thibaut s'est occupé de mettre en place les différentes requêtes que l'utilisateur peut faire à Tars via les pages web. Il a alors déployé pour la première fois Linux sur son Raspberry Pi, pour se familiariser avec l'architecture, et s'occuper du support de différentes fonctionnalités sur les systèmes d'exploitation, repris en mains par la suite par Olivier.

5. Mise au point final de l'IA, ajout du wakeword.

Olivier a alors, avec Thibaut, finalisé le fonctionnement de Tars concernant la diction de la réponse, avec la création d'un système de file d'attente pour dire les phrases les unes à la suite des autres, utiliser une autre voix si un problème intervient, ou encore si une phrase est lancée, tel que « Personnalité de Tars mise à jour », lorsque les paramètres sont modifiés, pour que les phrases ne se superposent pas. C'est aussi à ce moment que le wakeword « dis moi Tars » a été mis en place, avec quelques contraintes qui ont suivis. Nous avons en premier lieu utilisé la librairie Porcupine avec leur service sur internet permettant de créer un wakeword (machine learning entraîné à détecter une certaine phrase). Cela était initialement un choix temporaire, car nous souhaitons ne pas dépendre d'API sur cet aspect là.

6. Finalisations des derniers détails.

Pour finir, nous avons terminé quelques fonctionnalités qui étaient encore en cours de développement, et avons, pour l'instant, lors de la remise du projet, en vain réussi à développer notre propre modèle de machine learning pour le wakeword, choses étant pour l'instant hors de nos capacités.

> FONCTIONNEMENT ET OPÉRATIONNALITÉ :

Au moment du dépôt du projet, tout, excepté notre propre modèle de machine learning pour le wake word, est fini et parfaitement opérationnel. Est disponible sur le GitHub la copie de Tars, une « image », qui suivant les instructions, peut être copiée sur une autre carte SD pour le dupliquer (cette carte SD par la suite insérée dans un autre Raspberry Pi 4). Pour gérer les bugs, nous avons créé différentes fonctions s'exécutant lorsque ces derniers ont lieu, disant à l'utilisateur quoi faire. Nous pensons avoir bien répondu à la problématique que nous avons posé.

> OUVERTURE :

A moyen terme, nous pourrions par exemple intégrer de nouvelles fonctionnalités avancées de planification et de gestion de tâches, telles que la création et modification de calendriers, de rendez-vous pour mieux organiser son emploi du temps, ou encore le contrôle à distance d'autres appareils pour exécuter des tâches. Les ajouts suggérés précédemment pourraient aisément être ajouté à Tars par l'utilisateur lui-même, via le module `tars_functions.py` créé et expliciter sur Github. Elargir ses capacités linguistiques en prenant en charges un plus grand nombre de langues, serait aussi un bonne idée, permettant aux utilisateurs du monde entier de bénéficier de Tars, quelle que soit leur géolocalisation ou leur langue. Notre projet repose sur de multiples API, ce qui peut être un défaut, mais nous pensons que les technologies actuelles ne permettent pas à un Raspberry Pi de tout supporter localement. Malgré ces quelques remarques, nous percevons notre projet comme une réussite face à la problématique posée initialement, et aux enjeux y étant liés. Merci pour votre lecture !