

FILIÈRE INFORMATIQUE - SEMESTRE 8

Cahier des Charges

Application d'assistant vocal sur Reachy Mini



Auteurs :

RAÏS Sylvain
ZIZOUAN Widad
DIEUDONNÉ Clara
MARAIS Lucas
LAMHAMDI Aymane
ELFANI Hamza
BOUDEAU Benjamin

Encadrant :

ROLLET Antoine

Client :

N'GUYEN Steve

Table des matières

1	Introduction	2
2	Besoins fonctionnels et non fonctionnels	3
2.1	Besoins fonctionnels	3
2.2	Besoins non fonctionnels	4
3	Partie 1 : Les 100 ans de l'ENSEIRB	5
3.1	Diagramme d'états	5
3.2	Actions de Reachy selon les états et transitions	6
3.2.1	États	6
3.2.2	Transitions	7
3.3	Débuter une interaction avec le robot	8
3.4	Gestion des ordres	8
3.5	Prise de photos	9
3.6	Gestion des photos	10
3.6.1	Sauvegarde de la photo	10
3.6.2	Affichage de la photo sur un écran	10
3.7	Conversation	10
3.8	Fin d'interaction	10
4	Partie 2 : Après les 100 ans	11
4.1	Diagramme d'états	11
4.2	Filtres sur les photos	12
4.3	Recherche Internet	12
4.4	Conversations plus interactives	12
5	Deux cas d'utilisation	14
5.1	Interaction directe	14
5.2	Interaction par détection de visage	15
6	Tâches et livrables	16
6.1	Les étapes de réalisation techniques	16
6.1.1	La reconnaissance faciale	16
6.1.2	Mouvement du robot	16
6.1.3	Gestion des ordres	16
6.1.4	Synthèse vocale	17
6.1.5	Recherche Internet	17
6.2	Diagrammes de Gantt	17
6.3	Livrables	18
6.3.1	"Hello World" [07/02/22]	18
6.3.2	Full Interaction [16/02/22]	18
6.3.3	Le robot communique avec émotions [25/02/22]	18
6.3.4	Le robot sait prendre des photos [11/03/22]	18
6.3.5	Le robot communique et prend des photos [18/03/22]	18
6.3.6	100 ans de l'ENSEIRB [01/04/22]	18
6.3.7	Système de conversation amélioré [23/04/22]	18
6.3.8	Recherche internet pour des conditions de surf [30/04/22]	19
6.3.9	Full version [07/05/22]	19
7	Webographie	20
8	Annexe	21
8.1	Définition des positions	21
8.2	Ensembles de mots clé	21
8.3	Couple pour la conversation basique	23
8.4	Ensembles de phrases/mots clés pour la conversation avancée	24

1 Introduction

Le Projet au fil de l'année (PFA) est un projet regroupant 7 élèves ingénieurs de l'ENSEIRB-MATMECA, qui devront réaliser un projet pour un client jusqu'au 15 mai 2022. Ce PFA concerne le robot Reachy Mini et le client, N'GUYEN Steve, est un membre actif de l'entreprise Pollen Robotics, entreprise ayant développé le robot.

Fondée en 2016 par d'anciens chercheurs, Pollen Robotics est une entreprise qui rassemble des personnes talentueuses et indépendantes dans le but de fournir des produits et des applications accessibles et open source et d'amener l'IA et la robotique dans notre vie quotidienne. L'entreprise participe donc à l'évolution de l'IA dans la vie quotidienne, et expose son robot principal Reachy sur de nombreux événements autour de l'informatique, la robotique et l'intelligence artificielle.

Le robot Reachy Mini est l'une de leur création, aux côtés de son grand frère Reachy (robot articulé possédant des bras). Il possède une tête articulée et des antennes, le tout associé à un corps statique. Ses yeux sont formés par deux caméras, et il possède un microphone directionnel et des hauts parleurs. Fonctionnant grâce à un PC embarqué Intel NUC et un Google Coral, le robot possède tous les outils pour interagir avec le monde qui l'entoure.

Le travail demandé par le client est de programmer Reachy afin qu'il se rapproche d'un assistant vocal de type Amazon Echo, ou Google Home. La reconnaissance vocale, le traitement d'image, la synthèse vocale et les mouvements du robots seront donc à implémenter.

Le premier objectif est le bon fonctionnement du robot lors des 100 ans de l'ENSEIRB le 8 avril 2022. Le robot devra pouvoir interagir de façon simple avec les utilisateurs, ainsi que les prendre en photo selon leur volonté (photos simples, photos de groupes, ...).

Suite à cet événement, dans le prolongement du projet, les interactions seront développées de façon plus approfondie pour se rapprocher d'un réel assistant vocal avec notamment la recherche sur internet et des discussions plus naturelles.

Le cahier des charges se décompose en plusieurs parties, la première présentera les besoins fonctionnels, non fonctionnels et contraintes liés au projet. Ensuite nous diviserons ce projet en deux parties : le développement des fonctionnalités avant puis après les 100 ans de l'ENSEIRB. Enfin, nous développerons deux cas d'utilisation afin d'illustrer les capacités du robot, puis nous annoncerons les différents livrables qui seront rendus lors de la phase de construction du projet.



POLLEN ROBOTICS

FIGURE 1 – Logo de Pollen robotics

2 Besoins fonctionnels et non fonctionnels

Afin de mener à bien ce projet, nous avons identifié différents besoins que l'on peut répartir en deux catégories, les besoins fonctionnels qui correspondent aux besoins de services (fonctionnalités du robot accessibles et visibles à l'utilisateur), et les besoins non fonctionnels qui correspondent aux autres besoins, c'est à dire aux besoins techniques.

2.1 Besoins fonctionnels

Les besoins fonctionnels nécessaires à ce projet sont répartis de la manière suivante :

La reconnaissance faciale :

- Identification de visage
- Suivi de visage
- Cadrage pour prise de photos
- Échange de visages
- Mise en noir et blanc
- Similarité de visages

Gestion des données :

- Sauvegarde de la photo dans le disque dur interne
- Gérer l'espace de stockage en cas d'incapacité à enregistrer une autre photo
- Afficher la photo sur un écran connecté au robot

Mouvement du robot :

- Mouvement de tête et des antennes séparément
- Simuler des émotions
- Mouvement permettant de suivre les personnes
- Mouvement pour se tourner vers le son

Gestion des ordres :

- Transcription de la voix en texte
- Atténuation du bruit extérieur moyen et fort
- Mise en place d'un système de conversation
- Gestion des marqueurs ArUco

Synthèse vocale :

- Conversion d'un texte en son sortant du robot
- Émettre de petits sons transcrivant une "émotions"

Recherche Internet :

- Recherche des informations nécessaires
- Extraction des informations nécessaires
- Énoncer les informations

2.2 Besoins non fonctionnels

Pour ce qui est des besoins non fonctionnels, nous avons :

Les API utilisées pour les différentes parties du projet :

- OpenCV (reconnaissance faciale)
- SpeechRecognition (reconnaissance vocale)
- PyAudio (reconnaissance vocale)
- pyttsx3 (synthèse vocale)
- eSpeak (synthèse vocale)
- wave (gestion des fichiers audio)
- pydub (reconnaissance vocale)
- reachy-sdk (mouvement du robot + accès aux caméras)
- webbrowser et request (recherche internet)

Les périphériques nécessaires au bon fonctionnement du robot :

- Écran
- Clavier
- Souris

Contraintes :

- Le robot possède une tête articulée lui permettant d'effectuer ses mouvements. Cependant, cette tête peut rapidement chauffer et se mettre en "sécurité" en arrêtant ses moteurs, le robot ne sera alors plus fonctionnel. Il est donc nécessaire que le robot fasse des mouvements essentiels uniquement, et de limiter leur envergure, afin de limiter la hausse de la température.
- La reconnaissance vocale nécessite une connexion internet pour fonctionner, cela fait donc partie des contraintes. Néanmoins l'un des objectifs étant la "Recherche Internet" le robot devra forcément être doté d'une connexion internet quelle que soit l'API utilisée.

3 Partie 1 : Les 100 ans de l'ENSEIRB

Cette partie développe les besoins explicités en section 2, en détaillant les sous-besoins, pour l'exposition du robot aux 100 ans de l'ENSEIRB-MATMECA le 8 avril 2022. Le robot peut être décomposé comme une machine à états où il réalise des actions en fonction des états dans lesquels il se trouve et des transitions empruntées. Nous détaillerons dans cette partie le diagramme de ces états. Ensuite, nous expliquerons les actions réalisées par Reachy sur les états puis sur les transitions.

3.1 Diagramme d'états

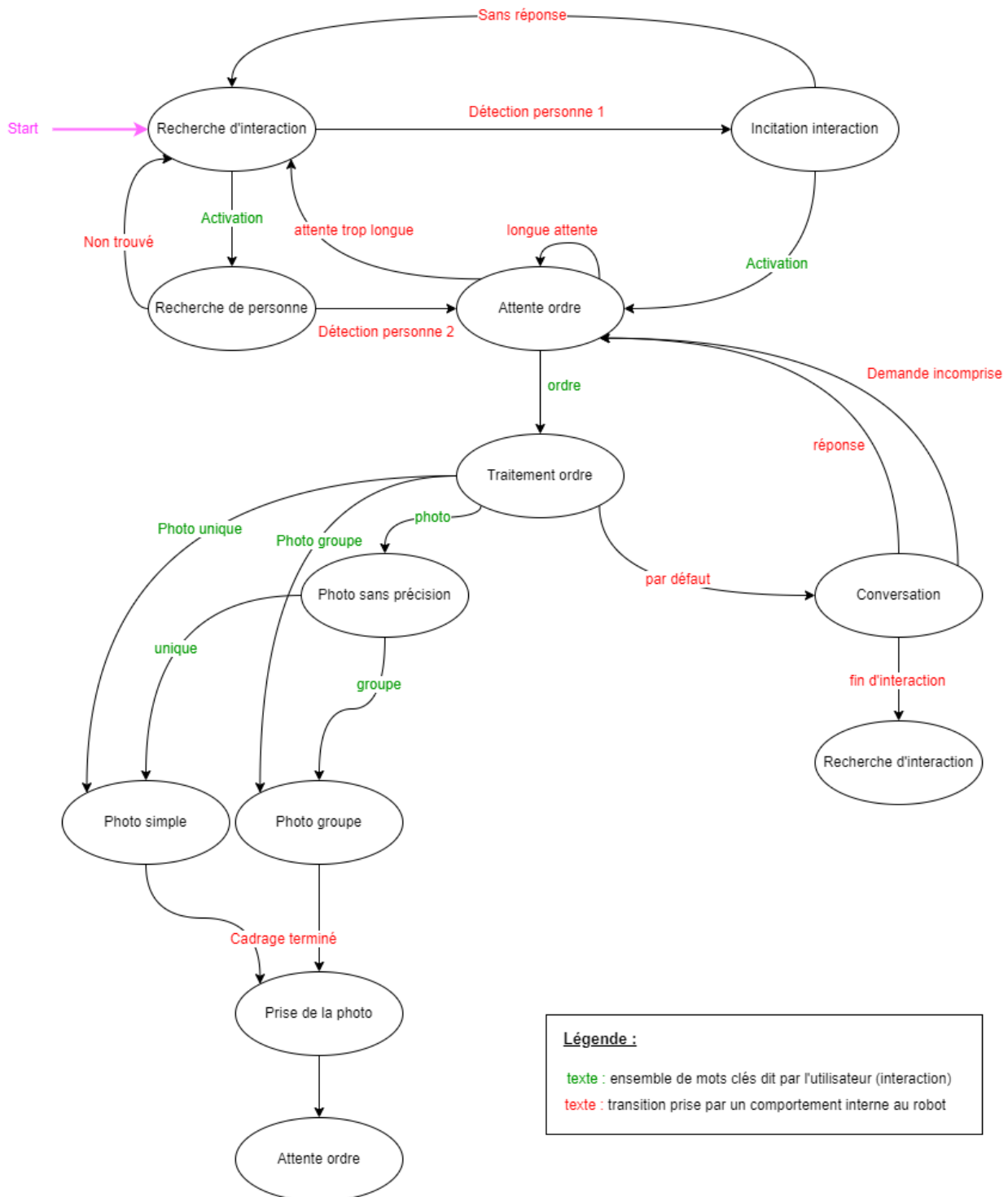


FIGURE 2 – Diagramme d'états pour les 100 ans de l'ENSEIRB

3.2 Actions de Reachy selon les états et transitions

Le diagramme est composé de deux parties : les états et les transitions. Sur chaque état et chaque transition, le robot réalisera des actions (mouvement, parole, ...) qui seront détaillées, d'abord pour les états, puis pour les transitions.

Chaque mouvement de la tête et des antennes du robot est mentionné par un *nom*, le détail technique des valeurs des positions est donné en **Annexe**, section 8.1, pour chacun de ses mouvements.

3.2.1 États

État "Recherche d'interaction" :

- Synthèse vocale : Fait des petits bruits toutes les 30 secondes
- Mouvement : Tourne la tête pour suivre la personne la plus proche du regard sauf toutes les 3 minutes où pendant 30s le robot fera le mouvement *triste*

État "Incitation interaction" :

- Synthèse vocale : Proposer une seule fois de dire "Hey Reachy" pour commencer l'interaction
- Mouvement : Fait le mouvement *incitation*

État "Attente ordre" :

- Mouvement : Fait le mouvement *écoute*

État "Traitement ordre" :

- Mouvement : Fait le mouvement *réflexion*

État "Recherche de personne" :

- Mouvement : Parcours de la zone possible avec recherche de visage (Cf OpenCV et direction sonore pour assister)

Etat "Photo sans précision" :

- Synthèse vocale : Demander "Vous préférez faire une photo simple ou une photo de groupe?"

Etat "Photo simple" :

- Mouvement : Cadrage selon OpenCV

Etat "Photo groupe" :

- Mouvement : Cadrage selon OpenCV

Etat "Conversation" :

- Synthèse vocale/Reconnaissance vocale : Si la phrase en entrée peut être interprétée, renvoie une réponse en fonction de la phrase en entrée (couples d'entrées / sorties), sinon (si la phrase d'entrée ne correspond à rien) dit "je n'ai pas compris votre demande"
- Mouvement : Fait le mouvement *réflexion*

Etat Prise de photo :

- Synthèse vocale : Dit "je prends la photo dans 5 4 3 2 1, dites Cheeeese"

3.2.2 Transitions

Transition "Détection personne 1" :

- Synthèse vocale : Après avoir trouvé, dit "Bonjour"
- Mouvement : Fait le mouvement *content*

Transition "Sans réponse" :

- Mouvement : Fait le mouvement *triste*

Transitions "Activation" :

- Synthèse vocale : Reachy répond "Hey, je vous écoute !" à la fin de la transition
- Mouvement : Fait le mouvement *écoute*

Transition "Longue attente" :

- Synthèse vocale : Dit "Je vous écoute, si vous voulez connaître l'ensemble de mes capacités demandez moi en disant "Que sais-tu faire?""
- Mouvement : Fait le mouvement *incitation*

Transition "Attente trop longue" :

- Mouvement : Fait le mouvement *triste*

Transition "Non trouvé" :

- Mouvement : Fait le mouvement *triste*

Transition "Détection personne 2" :

- Synthèse vocale : Dit "Oh je vous ai trouvé!"
- Mouvement : Fait le mouvement *écoute*

Transition "par défaut" :

- Mouvement : Fait le mouvement *écoute* ou si l'utilisateur dit un mot triste ou gentil :
 - Mot triste → mouvement *triste*
 - Mot gentil → mouvement *content*

Transition vers "photo unique", "photo groupe", "unique" et "groupe" :

- Mouvement : Fait le mouvement *écoute*

Transition "réponse" :

- None

Transition "demande incomprise" :

- Mouvement : Fait le mouvement *incitation*

Transition "Fin d'interaction" :

- Synthèse vocale Dit "Au revoir, c'était un plaisir"
- Mouvement : Fait le mouvement *remerciement*

3.3 Débuter une interaction avec le robot

Avant toute interaction, le robot émettra de légers bruits (Coucou – hmmmm – ehooo – Ayee Ayee.) pour capter l'attention des individus à proximité.

Ensuite, deux types de début d'interaction seront développés sur le robot selon si le robot a détecté l'utilisateur avant qu'il prononce "Hey Reachy" ou non.

- Dans le cas où un utilisateur prononce "Hey Reachy" avant que le robot ne le détecte. La tête se tournera vers la zone où le micro multi-directionnel identifie la voix et la reconnaissance faciale permettra d'effectuer une recherche locale d'un visage estimé "assez proche" pour être l'interlocuteur.

Une fois la personne détectée, Reachy lèvera les antennes vers le haut et regardera la personne pour lui faire comprendre qu'il l'a bien vue (mouvement *écoute*) et qu'il est prêt à l'écouter. L'interaction pourra alors commencer.

Dans le cas où le robot n'identifie pas la personne qui a dit "Hey Reachy", il fera le mouvement *triste*.

- Dans le cas où Reachy n'entend pas "Hey Reachy", le robot cherchera des interlocuteurs et les incitera à interagir avec lui, cela constitue le deuxième cas. En effet, avant qu'une personne ne rentre en interaction avec Reachy en lui disant "Hey Reachy", le robot suivra le visage de la personne passant le plus proche et fera, tout les 3 minutes et pendant 30s, le mouvement *triste*.

S'il détecte un utilisateur potentiel qui s'approche de lui, Reachy lèvera les antennes vers le haut et regardera la personne pour lui faire comprendre qu'il l'a bien vue et qu'il est prêt à l'écouter (mouvement *incitation*). Cela incitera la personne à communiquer avec le robot car il lui aura bien montré qu'il l'a détectée.

Après un "Hey Reachy" de la part d'un utilisateur le robot répondra "Bonjour" et l'interaction pourra commencer.

Dans un cas de nuisances sonores trop importantes, Reachy ne pourra pas identifier la commande vocale "Hey Reachy". Il sera alors possible pour un utilisateur extérieur de recourir à un marqueur ArUco¹ représentant un code afin d'indiquer à Reachy certaines actions. Ces actions seront en priorité l'initiation d'interaction, équivalent à prononcer "Hey Reachy" et à la prise de photo, simple ou en groupe.

Afin de permettre ces différents mouvements du robot nous allons utiliser l'API `reachy-sdk` de Pollen-Robotics développée pour le robot Reachy et par conséquent programmer en utilisant le langage *Python*. Elle permet de tourner la tête du robot vers la position demandée en une certaine durée. Ainsi que de bouger les antennes du robot à la vitesse souhaitée vers la position souhaitée. L'ensemble des positions, la durée et la vitesse sont paramétrables car ce sont des valeurs prises en paramètres des différentes fonctions.

3.4 Gestion des ordres

Pour donner des ordres à Reachy, les commandes devront contenir des mots clés spécifiques à chaque groupe de commandes.

Tout d'abord, la commande d'activation du robot nommée "Activation" dans le diagramme d'activité est paramétrable. Elle peut être définie par "Hey Reachy", "Ok Reachy", "Bonjour Reachy", etc. Une fois l'interaction débutée comme expliqué en section 3.3, le robot sera en attente d'un ordre (une commande lancée par l'utilisateur).

De manière générale, toutes les commandes, donc ensemble de mots clés et ensemble de couples entrée/sortie, seront spécifiées dans un ou plusieurs fichier(s) `.txt` afin de pouvoir les modifier sans avoir à toucher au code. Une commande à l'intérieur d'un fichier sera de la forme :

```
cmd : [nom de la commande]
. [mot clé]
. [mot clé]
...
```

Chaque commande aura donc un nom précédé de "cmd :" qui permettra à python de savoir de quelle commande il s'agit. Chaque mot clé sera précédé d'un "." et le marqueur de fin d'une liste de mots clés est "...".

1. Les marqueurs ArUco sont identifiables avec la bibliothèque OpenCV et sont faciles à implémenter

Pour écouter les ordres, le robot fera le mouvement *écoute*. Si Reachy n'a pas compris la demande de la personne, il fera le mouvement *incitation* pour faire comprendre qu'il n'a pas compris et annoncera "Je n'ai pas bien compris votre demande". Aussi, si au bout d'un certain temps la personne ne pose pas de questions Reachy fera le mouvement *incitation* et il proposera d'énumérer ses fonctionnalités. En cas de non interaction trop longue, le robot fera l'émotion triste avant de se remettre à chercher un autre utilisateur dans son état initial.

Comme indiqué sur le diagramme d'états, le robot gèrera des commandes pour la prise de photos qui seront détaillées dans la partie 3.5 et également des commandes de type conversation (cf 3.7).

À la suite d'une commande de type "Que sais-tu faire?" , Reachy énumérera toutes ses fonctionnalités à l'utilisateur.

Pour réaliser la partie reconnaissance vocale, nous avons choisi d'utiliser la bibliothèque open source `SpeechRecognition` ainsi que `PyAudio` qui offrent des fonctions pour l'écoute, l'enregistrement, la reconnaissance vocale et la traduction du son en texte.

En ce qui concerne la réalisation de la partie synthèse vocale, nous avons choisi d'utiliser la bibliothèque `pyttsx3` qui est une bibliothèque python open source qui convertit le texte en parole. Elle fonctionne hors ligne et utilise le synthétiseur vocal `eSpeak`.

Afin d'être utilisé durant les 100 ans de l'ENSEIRB-MATMECA le robot devra être fonctionnel dans un environnement bruyant. Il devra donc effectuer une réduction du bruit extérieur. Pour cela nous avons choisi d'utiliser la bibliothèque `wave` afin de pouvoir gérer les fichiers d'extension ".wav" ainsi que `pydub` afin de réaliser le filtrage. Chaque enregistrement sera donc stocké dans un fichier `enregistrement.wav` qui s'écrasera à chaque fois et qui sera traité avec `wave`.

3.5 Prise de photos

OpenCV est une bibliothèque proposant de nombreuses fonctionnalités de vision par ordinateur, spécialisée dans le traitement d'images en temps réel. Cette bibliothèque sera utilisée pour toute utilisation des caméras du robot et permettra la reconnaissance de visages, fonctionnalité utilisée pour la prise de photo. Elle permettra également d'effectuer les différents filtres photos cités dans les fonctionnalités à implémenter après les 100 ans de l'ENSEIRB-MATMECA, en section 4.2.

Il y a deux types distincts de prise de photos, la photo simple d'une personne et la photo d'un groupe de personnes.

Pour entrer dans la partie prise de photo, la commande devra contenir l'un des mots {"photo", "selfie"}. Afin de spécifier le type de la photo, il y aura deux cas de figure : si la commande contient l'un des mots {"moi", "me"}, alors ce sera une **photo simple**. Et si elle contient un ou plusieurs mots clés suivants {"nous", "les", "groupe", "ensemble"}, alors ce sera une **photo de groupe**.

Exemple de commandes pour une photo simple : "Prends-moi en photo", "Hey Reachy, est ce que tu peux me prendre en selfie?" .

L'utilisateur pourra demander une photo sans préciser le type de prise de photo. Dans ce cas, Reachy invitera l'utilisateur à préciser s'il souhaite une photo simple ou de groupe. L'utilisateur pourra également demander une photo simple ou une photo de groupe directement.

Dans le cas où Reachy comprend qu'il doit faire une photo mais ne connaît pas encore quel type d'image est souhaité, il dira : "Vous préférez faire une photo simple ou une photo de groupe?" .

Après connaissance de la demande de prise de photos et du type de photo, Reachy ajustera la position de sa tête pour correctement cadrer la photo. Reachy cadrera la photo en conséquence du type demandé à l'aide des positions des visages obtenues grâce à la reconnaissance faciale d'OpenCV.

Dans le cas d'une photo simple, Reachy centrera son regard sur la position de l'unique visage à prendre en photo (le plus proche dans son champ de vision). Pour une photo de groupe, Reachy considérera tous les visages présents dans son champ de vision, il supprimera ceux estimés trop loin (trop petits par rapport à la moyenne des tailles des visages en vue) et il orientera son champ de vision vers la position moyenne des visages encore

concernés.

Ensuite Reachy prendra la photo par enregistrement d'un instantané sur la capture vidéo avec OpenCV, et ce juste après l'annonce d'un compte à rebours : "Je prends la photo dans 5..4..3..2..1.. CHEEESE!".

Des images représentant un marqueur ArUco pourront être utilisées afin d'indiquer à Reachy quel type de prise de photos l'utilisateur souhaite en cas de nuisance sonore trop importante.

3.6 Gestion des photos

Une fois que la photo a été prise, elle sera sauvegardée sur la mémoire du robot et ensuite affichée afin que les utilisateurs puissent la voir.

3.6.1 Sauvegarde de la photo

Une fois que les photos auront été prises, elles vont être enregistrées dans la mémoire du robot. Nous avons choisi de nommer la photo avec la date, l'heure et les minutes de quand elle a été prise, par exemple 08_04_2022-15_45.jpg. Cependant, sa mémoire étant limitée, il nous faudra donc gérer le fait de vouloir sauvegarder une photo supplémentaire lorsque qu'il n'y aura plus assez d'espace.

Dans ce cas, nous favoriserons les photos récentes à celles prises précédemment. Pour cela, lorsque l'enregistrement de la photo ne pourra pas être fait par manque d'espace mémoire, un message sera envoyé pour avertir les administrateurs. Les administrateurs exécuteront une commande permettant de supprimer le nombre souhaité des plus anciennes photos (suppression à l'aide du module python `os`). Cela nous permettra de libérer assez d'espace pour enregistrer de nouvelles photos.

3.6.2 Affichage de la photo sur un écran

En plus de pouvoir sauvegarder les photos, un besoin spécifie que les utilisateurs puissent voir la photo une fois qu'elle aura été prise pour voir si elle leur convient. Pour cela, la photo sera affichée en temps réel sur un écran connecté au robot via un câble HDMI.

3.7 Conversation

Dans un premier temps Reachy pourra entretenir une conversation très simple avec son interlocuteur. Cette conversation sera composée d'entrées et de sorties, par exemple le robot répondra "Bonjour" si on lui dit "Salut", "Bonjour" ou bien "Hello". Il répondra également qu'il va très bien si on lui pose la question mais on ne pourra pas converser de manière complexe car il ne retient pas l'état de la conversation (cela sera amélioré par la suite, cf 4.4). L'ensemble des couples entrées / sorties est précisé en annexe (cf 8.3).

Lorsqu'une personne sera en train de discuter avec le robot, il réagira à ce que la personne lui dit. Par exemple avoir l'air triste lorsque qu'une personne lui dira des mots considérés comme méchants et inversement content lorsqu'il entendra des mots considérés comme gentils. Si Reachy ne comprend pas les mots dit par la personne, il fera le mouvement *incitation* pour faire comprendre qu'il n'a pas compris.

3.8 Fin d'interaction

Lorsque l'interaction sera terminée, par la demande de l'utilisateur avec une commande ou si le robot ne reçoit plus d'ordre, Reachy remerciera la personne d'avoir interagi avec lui tout en faisant le mouvement *remerciement*.

4 Partie 2 : Après les 100 ans

Après les 100 ans de l'**ENSEIRB-MATMECA**, nous développerons les fonctionnalités liées aux filtres sur les photos, à la recherche par Internet ainsi qu'aux conversations. Ces fonctionnalités sont détaillées ci-dessous.

4.1 Diagramme d'états

Voici le diagramme d'état de la version finale du robot :

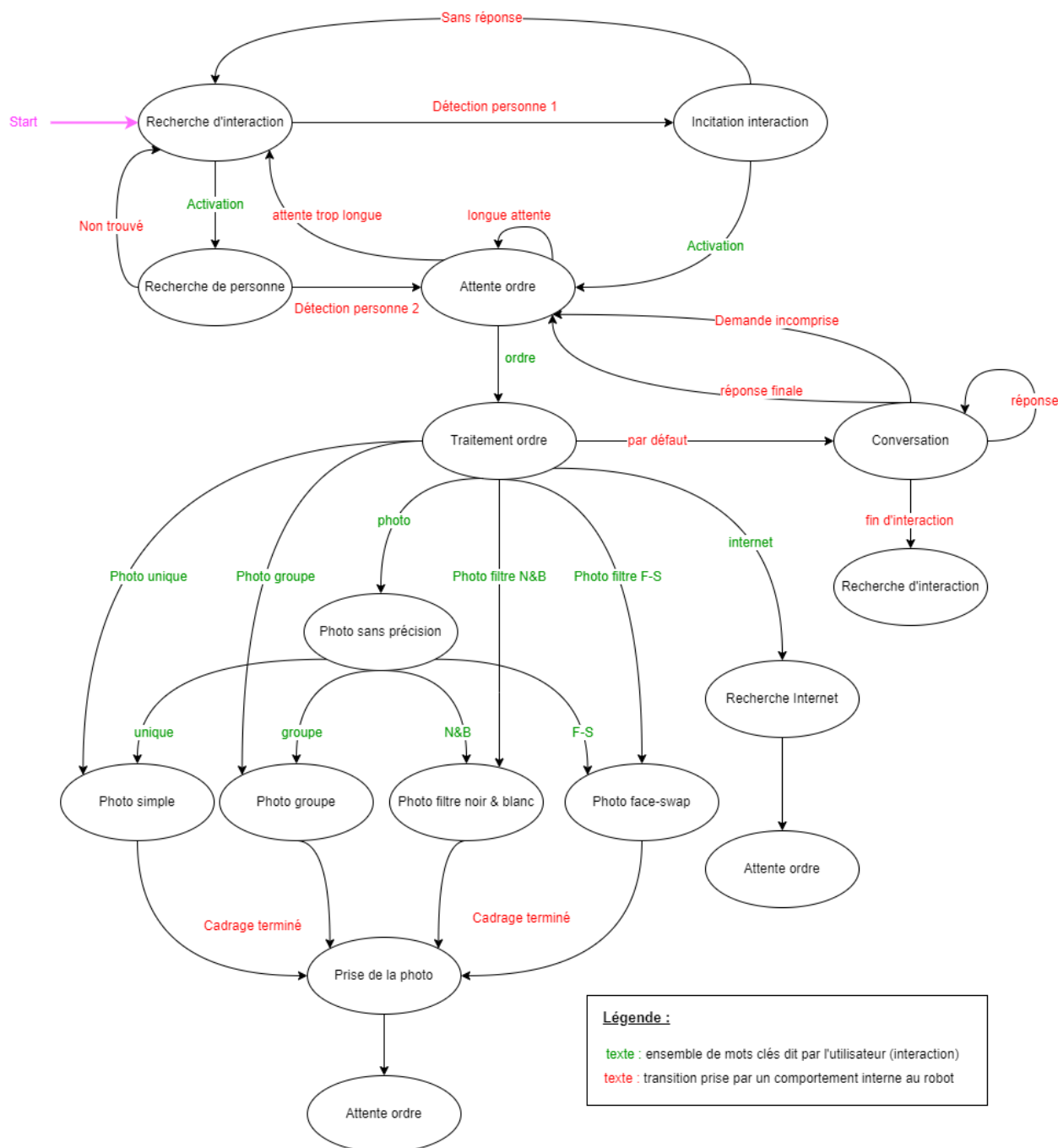


FIGURE 3 – Diagramme d'états final du Reachy Mini

4.2 Filtres sur les photos

Additionnellement à la prise de photos, des filtres pourront être appliqués à ces photos :

- Le premier filtre consistera à faire de l'échange de visages, où les deux visages les plus proches sur la photo seront échangés.
- Le second filtre consistera à mettre la photo en noir et blanc.

Si le développement est assez avancé, les filtres pourront être appliqués en direct sur la capture vidéo, et une dernière fonctionnalité consistera à indiquer la similarité des deux visages les plus proches sur la photo.

Ces fonctionnalités seront implémentées à l'aide de l'API d'OpenCV.

4.3 Recherche Internet

L'un des besoins fonctionnels du client est la recherche sur internet, notamment pour des conditions de surf à une date et un lieu précis. Ainsi nous utiliserons les API `webbrowser` et `request` afin de réaliser des recherches selon des urls et extraire des données trouvées sur une page web.

Si l'utilisateur réalise une demande contenant un des mots clés `{condition, surf}`, alors le robot lui demandera pour quel jour `J` et quel lieu `L` réaliser la recherche des conditions de surf. Reachy pourra ainsi effectuer une recherche internet en complétant l'url correctement avec `J` et `L`, et en extraire les informations importantes qu'il dira à l'utilisateur par l'intermédiaire de la synthèse vocale.

Ces recherches seront effectuées sur le site <https://www.surf-report.com/>, et le robot explicitera la hauteur des vagues, la température sur place ainsi que le temps.

Lorsque la personne aura demandé d'effectuer une recherche, il fera le mouvement *réflexion* ainsi que le mouvement *incitation* s'il n'a pas compris la demande de la personne.

4.4 Conversations plus interactives

Après les 100 ans de l'ENSEIRB la partie conversation du robot devra être améliorée. Le robot devra être capable de tenir une conversation simple avec son interlocuteur, il devra notamment pouvoir tenir une conversation du type :

- (I) Bonjour.
- (R) Hello !
- (I) Qui es-tu ?
- (R) Je suis Reachy et vous ?
- (I) Je m'appelle Test.
- (R) Enchanté Test
- (I) Comment vas-tu ?
- (R) Je me porte bien merci, et vous Test ?
- (I) Je vais très bien
- (R) Super
- (I) Que fais-tu ici ?
- (R) Je propose mes services de photographe pour l'évènement d'aujourd'hui.

Pour cela nous allons continuer avec des mots clés mais en ajoutant une notion de mémoire de l'état de la conversation et comme l'indique le nouveau diagramme d'états on ne sort pas forcément directement de l'état Conversation à chaque commande. La figure 4 résume les interactions ajoutées :

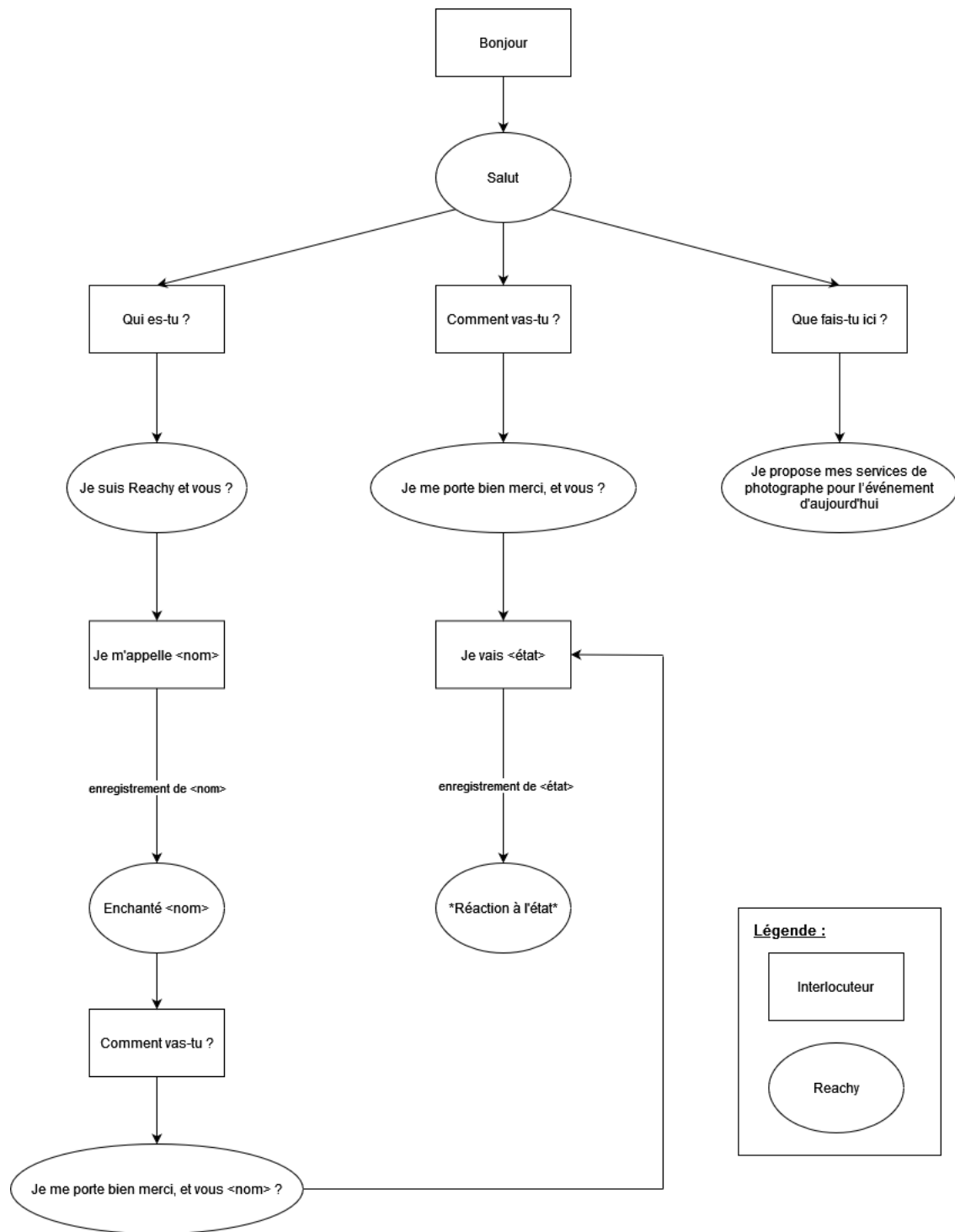


FIGURE 4 – Schéma des Conversations avancées

On peut voir les rectangles de la Figure 4 comme des noms d'ensembles de mots clés qui sont spécifiés en annexe (cf 8.2). Les ellipses quant à elles peuvent être vues comme les sorties car ce seront les phrases dites par le robot.

Cette partie conversation avancée pourra être densifiée de manière optionnelle, voici ce qui pourra être rajouté :

- Des couples entrée/sortie marrants comme par exemple : "Je suis ton père" / "Vous devez confondre, je ne suis pas Luke Skywalker".
- Des devinettes avec un fichier contenant un ensemble de couples question/réponses, lues grâce aux commandes natives de python. L'interlocuteur pourra demander une devinette, le robot choisira un couple

question/réponses, posera la question et vérifiera la conformité de la réponses avec celle qu'il connaît (mots clés pour les réponses).

- Des questions de culture générale sur le même modèle que les devinettes.
- Des blind tests sur des extraits audio de 10 secondes sur le même modèle que les devinettes.

5 Deux cas d'utilisation

Les principales fonctionnalités du robot peuvent être mises en application à l'aide de cas d'utilisation. La couverture du diagramme d'état est un indice de la force de ces cas d'utilisation. Il serait possible d'obtenir une pleine couverture en réalisant des cas d'utilisation pour chaque type de photo, et chaque type de conversation (recherche internet ou non, conversation simple ou non). Nous développerons ici les deux principaux cas qui permettent une couverture suffisamment élevée.

5.1 Interaction directe

Cas d'utilisation	Interaction directe
Description	L'utilisateur n'est pas détecté par le robot, il se dirige vers lui et lui demande de le prendre en photo.
Acteurs	- Un utilisateur - Reachy
Scénario	<ul style="list-style-type: none"> - L'utilisateur arrive et dit "Hey Reachy" au robot. - Reachy détecte la personne en redirigeant sa tête vers lui, fait le mouvement <i>écoute</i>. Il lui répond: "Hey, je vous écoute !" et se met en état d'attente d'ordre. - L'utilisateur lui répond: "Je veux faire une photo" - Reachy redresse sa tête et lui demande: "Vous voulez une photo simple ou une photo de groupe ?" - L'utilisateur précise qu'il veut une photo simple. - Reachy cadre la photo, vérifie que l'utilisateur est bien centré dans l'image, et ensuite se met à prendre la photo: "Je prends la photo dans 5 4 3 2 1 ... cheeese !". - Reachy sauvegarde la photo, l'affiche sur l'écran connecté à lui et se met en état d'attente d'ordre. - L'utilisateur part sans dire au revoir. - Reachy dit: "Je vous écoute, si vous voulez connaître l'ensemble de mes capacités demandez moi en disant "Que sais-tu faire ?" " - Reachy ne détecte plus la personne, il fait le mouvement <i>triste</i>, redresse la tête et se remet dans l'état de recherche d'interaction.
Pré-conditions	- Stockage non saturé

FIGURE 5 – 1^{er} cas d'utilisation

Dans un premier cas, on considère un utilisateur qui n'est pas détecté par le robot. Il adresse la parole à Reachy en lui disant "Hey Reachy". Reachy recherche donc l'utilisateur en tournant la tête vers lui.

Une fois la personne détectée, le robot, fait le mouvement *écoute*, et répond "Hey, je vous écoute!" . Il se met alors en attente d'un ordre de l'utilisateur. Ce dernier demande "Je veux faire une photo". Reachy redresse sa tête et demande : "Vous préférez faire une photo simple ou une photo de groupe?". L'utilisateur précise qu'il veut faire une photo simple.

Le robot se met donc à cadrer la photo afin d'avoir l'utilisateur au milieu de l'image. Ensuite, il rentre en phase de prise de photo et lance le compte à rebours pour capturer la photo en disant : "Je prends la photo dans 5..4..3..2..1..CHEEESE!". Une fois la photo prise, elle sera sauvegardée et affichée sur l'appareil connecté au robot.

Reachy se retrouve alors de nouveau en attente d'un ordre. L'utilisateur est parti sans dire au revoir, donc le robot considère l'attente trop longue, et précise "Je vous écoute, si vous voulez connaître l'ensemble de mes capacités demandez moi en disant "Que sais-tu faire?"". Comme l'utilisateur n'est plus là, le robot fait le mouvement *triste* puis redresse la tête pour la recherche de visages qui est le premier état.

5.2 Interaction par détection de visage

Cas d'utilisation	Interaction après détection de visage
Description	L'utilisateur est détecté par le robot qui l'incite à interagir
Acteurs	- Un utilisateur - Reachy
Scénario	<ul style="list-style-type: none"> - Le robot recherche avec qui interagir, il fait des mouvements et des bruits mignons pour attirer l'attention. - Le robot détecte une personne, il bouge ses antennes, fait le mouvement <i>content</i> et dit "Bonjour !" - Le robot lui propose de dire "Hey Reachy" tout en faisant le mouvement <i>incitation</i> pour commencer l'interaction. - L'utilisateur dit "Hey Reachy" - Le robot lui répond: "Hey, je vous écoute" - L'interlocuteur lui demande: "Comment vas-tu ?" - Le robot lui répond: "Je vais très bien" - L'utilisateur lui demande: "Sais-tu préparer des cookies ?" - Reachy ne comprend pas la demande et répond: "Je n'ai pas bien compris votre demande" en faisant le mouvement <i>incitation</i>. - L'utilisateur lui répond méchamment: " Tu es nul et tu ne sers à rien !" - Reachy détecte des mots tristes et fait le mouvement <i>triste</i> (10 secondes). - Reachy redresse sa tête et attend l'ordre suivant. - L'utilisateur lui dit: "Au revoir !" - Reachy comprend que c'est la fin d'interaction et lui répond: "A bientôt" et réalise le mouvement <i>remerciement</i> - Reachy se remet dans l'état de base et recherche avec qui interagir.

FIGURE 6 – 2^{ème} cas d'utilisation

Dans ce deuxième cas, l'interaction est débutée par la détection d'un utilisateur par Reachy.

Le robot recherche une personne tout en faisant des mouvements et des bruits mignons pour attirer l'attention. Il détecte donc une personne à proximité, et se met à agiter les antennes avec le mouvement *content* et dit "Bonjour".

Il propose ensuite au client de dire "Hey Reachy" tout en faisant le mouvement *incitation* pour commencer l'interaction. L'utilisateur déclenche effectivement l'interaction en disant "Hey Reachy" et Reachy répond "Hey, je vous écoute". Ensuite l'interlocuteur demande "Comment vas-tu?", le robot lui répond alors "Je vais très bien".

Ensuite l'utilisateur demande : "Sais-tu préparer des cookies?". Reachy dans ce cas ne comprend pas et répond gentiment "Je n'ai pas bien compris votre demande" en faisant le mouvement *incitation*.

L'utilisateur enchaîne ensuite la conversation en disant "Tu es nul de toute façon!". Le robot identifie donc que ce sont des mots tristes, et fait le mouvement *triste* pendant 10 secondes.

Après il redresse sa tête et attend le prochain ordre ou la prochaine réplique. Le client lui dit "A bientôt", Reachy comprend donc que le client s'écarte. Il lui répond "A bientôt", réalise le mouvement *remerciement* et se remet dans l'état de base qui est la recherche de personne.

6 Tâches et livrables

Dans cette partie, nous détaillerons les étapes de la réalisation technique, ensuite, nous présenterons les diagrammes de Gantt et nous listerons tous les livrables.

6.1 Les étapes de réalisation techniques

6.1.1 La reconnaissance faciale

- **Identification de visage** [2 jours] : Obtention en capture vidéo d'une structure de donnée renseignant des informations sur la position des visages et affichage à l'écran clarifiant la réussite de l'identification (carré autour des têtes, points particuliers du visage, ...)
- **Suivi de visage** [1-2 semaine(s)] : Obtention des angles horizontaux et verticaux entre la normale à la caméra et le centre des visages de la capture vidéo. Adaptation des angles ressortis pour un suivi optimal du visage le plus proche.
- **Cadrage pour prise de photo** [1-2 semaine(s)] : Prise en compte des visages les plus proches, puis recherche du barycentre des centres des visages pour diriger le regard vers ce barycentre en guise de cadrage
- **Échange de visages** [1-3 semaines moyennant qualitativité] :
 - [3 jours] : Échange des deux têtes les plus proches (par les carrés les entourant)
 - [5 jours] : Échange des deux visages les plus proches (par une méthode plus poussée). Les visages n'auront pas les bonnes proportions
 - [1 semaine] : Correction des proportions et adaptation des traits faciaux
 - [5 jours] : Adaptation de la colorimétrie des visages échangés
- **Mise en noir et blanc** [1 jour] : Colorimétrie de la photo transformée en noir et blanc
- **Similarité de visages** [5 jours] : Pourcentage de similarité des deux visages les plus proches sur la photo

6.1.2 Mouvement du robot

- **Premier mouvement de tête et des antennes séparément** [1-3 jour(s)] : effectuer un mouvement quelconque avec la tête, de même pour les antennes.
- **Faire avoir des émotions** [1-2 semaine(s)] : Combiner des mouvements de la tête et des antennes pour faire comme si le robot avait des émotions ("content", "triste", "pensif", ...).
- **Faire avoir des émotions à partir d'une position initiale connue** [1-2 semaines] : Lui faire avoir les émotions dans l'orientation où la tête était avant ; pour cela lui donner les déplacements sous la forme de 3 angles qui seront ensuite transformés sous la forme de coordonnées cartésiennes.
- **Mouvement lié aux photos** [1 semaine] : Récupérer, de la partie photo, les valeurs des angles dont le robot doit bouger pour pouvoir prendre puis cadrer la photo.
- **Mouvement permettant de suivre les personnes** [1 semaine] : Tourner la tête pour qu'elle soit toujours en face de la personne qui, soit interagit avec Reachy, soit la personne la plus proche lorsque le robot n'interagit avec personne.
- **Mouvement pour se tourner vers le son** [1 semaine] : Récupérer, de la partie reconnaissance vocale d'où vient le son, les valeurs des angles dont le robot doit bouger pour pouvoir plus facilement trouver où se trouve la personne qui parle.
- **Fluidifier les mouvements** [2 semaines] : Faire que les mouvements soient plus fluides et plus naturels afin qu'ils se rapprochent plus de ceux d'un être humain.

6.1.3 Gestion des ordres

- **Transcription de la voix en texte** [1 jour] : écouter l'interlocuteur et transformer ses paroles en texte.
- **Atténuation du bruit extérieur moyen** [8-13 jour(s)] : atténuer les bruits extérieurs afin de se concentrer sur la source principale du son et pouvoir être utilisé dans un environnement moyennement bruyant
- **Mise en place d'un système de conversation basique** [3-5 jour(s)] : le robot doit pouvoir répondre à de commandes telles que "Bonjour" ou bien "Qui es-tu?".
- **Gestion des marqueurs ArUco** [5-8 jours] : reconnaître un marqueur ArUco, le retranscrire en entier et l'associer à une commande existante pour pouvoir être utilisé sans reconnaissance vocale.

- **Atténuation du bruit extérieur fort** [9-14 jour(s)] : atténuer les bruits extérieurs afin de se concentrer sur la source principale du son et pouvoir être utilisé dans un environnement fortement bruyant.
- **Amélioration du système de conversation** [12-15 jour(s)] : le robot doit pouvoir tenir une conversation basique (cf 4.4).

6.1.4 Synthèse vocale

- **Conversion d'un texte en son** [1 jour] : faire saisir un texte et le transformer en parole.
- **Faire parler le robot** [4 jours] : vérifier les sorties audio du robot et ses paramètres et le faire parler.
- **Faire des petits bruits** [4 jours] : faire produire des petits bruits mignons et des phrases pour attirer l'attention.
- **Amélioration de la voix** [1 semaine] : améliorer la voix du robot.

6.1.5 Recherche Internet

- **Reconnaissance des informations nécessaires** [5 jour] : Transformer les paramètres énoncés par l'utilisateur en url de recherche.
- **Extraction des informations nécessaires** [1 semaine] : Extraire sur la page web les informations utiles (hauteur de vagues, température, temps).
- **Énoncer les informations** [1 semaine] : Énoncer ces informations à l'utilisateur de façon précise et concise.

6.2 Diagrammes de Gantt

Toutes ces étapes de la réalisation peuvent être résumées dans un diagramme, nommé diagramme de Gantt, qui permet de définir des dates précises concernant l'évolution du robot. De plus, cela permet la visualisation de livrables qui seront rendus pour réaliser des cycles de construction du projet et retour du client. Nous avons découpé ce diagramme en deux sous diagrammes, un concernant avant les 100 ans de l'**ENSEIRB-MATMECA** et un après.

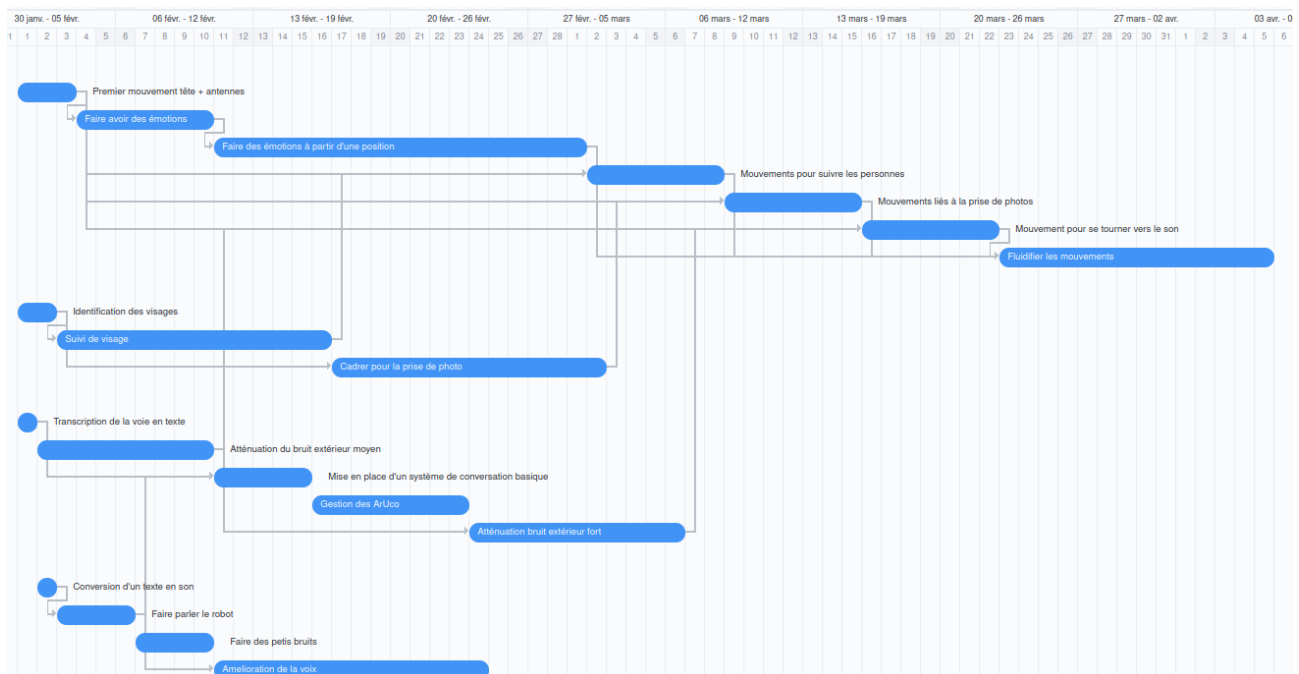


FIGURE 7 – Diagramme de Gantt avant les 100 ans

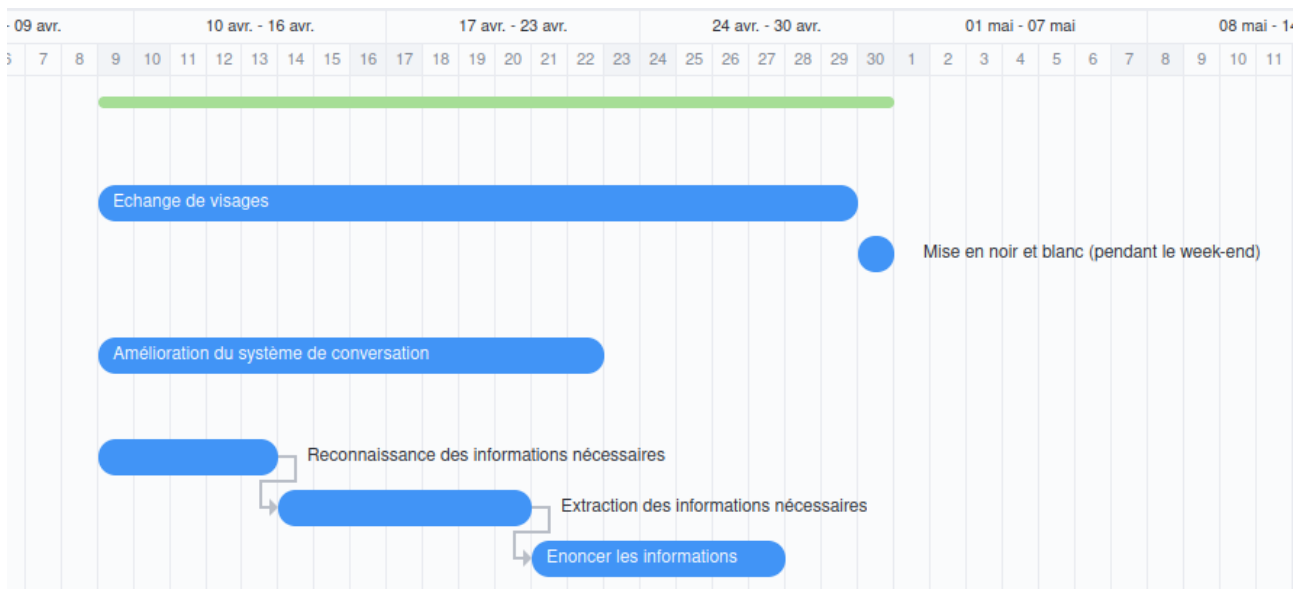


FIGURE 8 – Diagramme de Gantt après les 100 ans

6.3 Livrables

Voici les livrables qui se dégagent du diagramme de Gantt :

6.3.1 "Hello World" [07/02/22]

L'objectif de ce livrable est de fournir un robot capable d'effectuer les actions les plus basiques de chacun des axes majeurs de son fonctionnement. Il devra donc pouvoir effectuer un mouvement simple, reconnaître un visage, entendre une voix et la transformer en texte et dire une phrase quelconque.

6.3.2 Full Interaction [16/02/22]

L'objectif de ce livrable est de fournir un robot capable de lier ses axes majeurs de fonctionnement entre eux de manière basique. Il devra donc reconnaître une personne, bouger en conséquence, écouter une voix et répéter la phrase dite.

6.3.3 Le robot communique avec émotions [25/02/22]

L'objectif de ce livrable est de fournir un robot capable d'avoir une conversation basique avec l'utilisateur tout en montrant ses émotions.

6.3.4 Le robot sait prendre des photos [11/03/22]

L'objectif de ce livrable est de fournir un robot capable de cadrer une ou plusieurs personnes pour les prendre en photo.

6.3.5 Le robot communique et prend des photos [18/03/22]

L'objectif de ce livrable est de fournir un robot capable d'avoir une conversation plus dynamique avec l'utilisateur après l'avoir détecté et capable de le cadrer et le prendre en photo.

6.3.6 100 ans de l'ENSEIRB [01/04/22]

L'objectif de ce livrable est de fournir un robot capable d'effectuer tout ce qui est spécifié dans la partie 3. Les mouvements du robot sont plus fluides et le robot propose différents filtres et fonctionnalités lors de la prise d'une photo.

6.3.7 Système de conversation amélioré [23/04/22]

Le robot peut entretenir une conversation avec un utilisateur de manière fluide et interactive (mémorisation du prénom, action en fonction du sentiment)

6.3.8 Recherche internet pour des conditions de surf [30/04/22]

Le robot permet d'annoncer les conditions de surf pour une date et un lieu précis selon la demande de l'utilisateur.

6.3.9 Full version [07/05/22]

L'objectif de ce livrable est de fournir un robot capable d'effectuer tout ce qui est spécifié dans ce cahier des charges. Les filtres sur les photos auront donc été rajoutés.

7 Webographie

- [1] <https://www.pollen-robotics.com>
- [2] <https://github.com/pollen-robotics/reachy-sdk>
- [3] <https://docs.pollen-robotics.com>
- [4] <https://docs.python.org/fr/3/library/wave.html>
- [5] <https://opencv.org>
- [6] <https://pypi.org/project/SpeechRecognition>
- [7] <https://pypi.org/project/PyAudio>
- [8] <https://pypi.org/project/pydub>
- [9] <https://pypi.org/project/pyttsx3>
- [10] <https://towardsdatascience.com>
- [11] <https://docs.python.org/3/library/webbrowser.html>
- [12] <https://fr.python-requests.org/en/latest>

8 Annexe

8.1 Définition des positions

Une position est la combinaison d'un mouvement de la tête et des antennes. Le mouvement de la tête est réalisé en fonction de 3 coordonnées (x , y , z) qui déterminent un point de l'espace où Reachy regarde. Pour se rendre dans une certaine position, il est possible de déterminer la durée que prendra le mouvement. Le mouvement des antennes est caractérisé par une vitesse limite des antennes ainsi que deux angles qui correspondent au positionnement de chaque antenne.

Nous détaillons ici les différentes positions de notre robot, dont nous pourrions factoriser le code.

Position	Cordonnées	Durée	Angle Antenne Gauche	Angle Antenne Droite	Vitesse Antennes
écoute	(0.5, 0, -0.05)	2.0	0	0	/
triste	(0.5, 0, -0.3)	2.0	140.0	-140.0	70
content	(0.5, 0, -0.05)	2.0	± 20.0	± 20.0	300
incitation	(0.5, 0, 0.05)	1.0	35.0	-35.0	70
réflexion	(0.5, 0.15, 0.15)	1.0	-40.0	40.0	70
remerciement	(0.5, 0, -0.20)	0.5	-40.0	40.0	70

Le mouvement content est un mouvement où les antennes alternent leur position. Ainsi, elles alterneront en différé entre -20 et +20. Le mouvement remerciement correspond à la mise en position écoute, puis remerciement, puis écoute. Ainsi, le robot baisse la tête comme pour saluer son interlocuteur.

8.2 Ensembles de mots clé

Nous avons choisi de définir des ensembles de mots pour les commandes. La transition dans le diagramme d'états est faite selon le type de transition que l'on a choisi.

Voici les différentes transitions :

- "one in" = si la phrase de l'utilisateur contient un mot de l'ensemble de mots.
- "all in" = si la phrase de l'utilisateur contient tous les mots de l'ensemble de mots.
- "n in" = si la phrase de l'utilisateur contient n mots de l'ensemble de mots.

Ensemble "Hey Reachy" ("all in") :

- Hey
- Reachy

Ensemble "Photo" ("one in") :

- photo
- selfie
- photographie
- photographe
- cliché

Ensemble "unique" ("one in") :

- moi
- me

Ensemble "Groupe" ("one in") :

- nous
- groupe
- ensemble
- les

Dans le diagramme il y a une transition directe entre "traitement ordre" et "photo simple" ou "photo groupe". En fait il s'agit d'une double analyse de la phrase de l'utilisateur par Reachy. Si la phrase est "prend moi en photo" on passera d'abord dans l'état "Photo sans précision" avec la détection de "photo" puis on passera dans l'état "Photo simple" avec la détection de "moi".

Exemples "Photo unique" :

- Prend **moi** en **photo**.
- Tu peux **me** prendre en **photo** ?
- Tu peux **me** **photographier** ?
- Prend **moi** en **selfie**.

Exemples "Photo groupe"

- Prend **nous** en **photo**
- Prend une **photographie** de **groupe**
- Fait un **cliché** de notre **groupe**
- Tu peux nous **prendre** en photo **ensemble** ?

Passons maintenant aux ensembles de mots clés concernant les fonctionnalités implémentées après les 100 ans de l'ENSEIRB.

Ensemble "N&B" ("one in") :

- noir et blanc
- monochrome

Ensemble "F-S" ("one in") :

- échangeant
- échange

Exemples "Photo N&B"

- Prend une **photo monochrome**
- Prend une **photo** en **noir et blanc**

Exemples "Photo F-S"

- Prend une **photo** avec un **échange** de visage
- Prend une **photo** en **échangeant** nos visages

Ensemble "internet" ("one in") :

- conditions
- surf

Ce cas est différent car il demande une date et un lieu on devra donc repérer une date qui pourra être spécifiée de cette façon, on repère plusieurs cas :

Cas pas de date :

On recherche les conditions de surf le jour actuel.

Exemple : "quelles sont les conditions de surf à Lacanau ?"

Cas une date peu précise (jour de la semaine) :

On recherche les conditions de surf correspondantes au jour de la semaine précisé le plus proche de la date actuelle. Exemple : "quelles sont les conditions de surf Samedi à Lacanau ?"

Cas une date précise :

On recherche les conditions de surf le jour spécifié.

Exemple : "quelles sont les conditions de surf le Jeudi 27 Janvier à Lacanau?"

Cas pas de lieu :

On recherche les conditions de surf au Grand Crohot.

Exemple : "quelles sont les conditions de surf aujourd'hui?"

Cas un lieu précis :

On recherche les conditions de surf lieu précisé.

Exemple : "quelles sont les conditions de surf aujourd'hui à Lacanau?"

8.3 Couple pour la conversation basique

Nous n'avons pas pour objectif de permettre une longue conversation avec le Reachy, il s'agirait plutôt de couple d'entrées et de sorties sans garder en mémoire l'état de la conversation actuelle. Le Reachy utilisera la bibliothèque **random** afin de pouvoir gérer l'aléatoire, ici une sortie est présentée sous la forme du texte de sortie suivi du pourcentage de chance de renvoyer cette sortie entre parenthèse.

Entrée :

- Bonjour
- Coucou
- Salut
- Hello

Sortie :

- Bonjour (50%)
- Salut (50%)

Entrée :

- Comment vas-tu ?
- Comment ça va ?

Sortie :

- Très bien et vous ? (100%)

Entrée (mots clés; "one in") :

- gentil
- sympathique
- mignon
- beau

Sortie :

- Merci (100%) + Mouvement : content

Entrée (mots clés; "one in") :

- méchant
- moche
- inutile
- nul

Sortie :

- Mouvement : *triste*

Entrée incomprise

Sortie :

- Je n'ai pas bien compris votre demande ? (100%)

8.4 Ensembles de phrases/mots clés pour la conversation avancée

Ensemble "Qui es-tu?" ("one in") :

- Qui es-tu ?
- Qu'es-tu ?

Ensemble "Je m'appelle [nom]" ("one in") :

- Je m'appelle [nom]
- Je suis [nom]
- [nom]

L'ensemble "Je m'appelle [nom]" fait suite à "Qui es-tu?" donc Reachy sait que son interlocuteur va lui transmettre son nom c'est pourquoi si la réponse est composée d'un seul mot, Reachy l'interprétera comme le nom de son interlocuteur.

Ensemble "Que fais-tu ici?" :

Cet ensemble est un peu particulier, il est composé de 2 ensembles de mots clés à détecter. En effet, la phrase doit contenir l'un des mots suivants :

- Fais
- Pourquoi

Ainsi que l'un de ces mots :

- ici
- là

Ensemble "Comment vas-tu?" ("one in") :

- Comment vas-tu ?
- Ça va ?
- Tout va bien ?

Ensemble "Je vais [état]" ("one in") :

- Je vais [état]
- [état]

L'ensemble "Je vais [état]" se rapproche de "Je m'appelle [nom]" puisque Reachy s'attend ici aussi au type de réponse puisqu'il fait suite à une question de Reachy, un état seul peut donc suffire.

La suite fait la liste des états possibles détectés par Reachy ainsi que les mots clés associés.

Ensemble "bien" ("one in") :

- bien
- très bien
- super
- ça va
- en forme

Réponse de Reachy (Réaction à l'état) : "Super" + mouvement *content*

Ensemble "mal" ("one in") :

- pas (afin de détecter les états du type "ça ne va pas")
- mal
- triste
- fatigué

Réponse de Reachy (Réaction à l'état) : mouvement *triste*