Département des Technologies de l’information

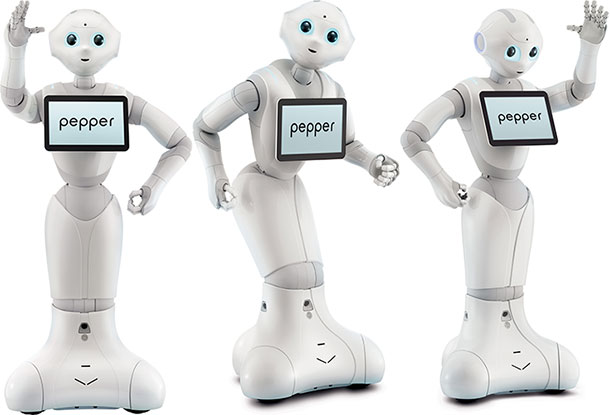
et de la communication

Filière informatique, Orientation Logiciel

**MÉMOIRE**

En vue de l’obtention du diplôme de Bachelor of Science HES-SO

PEPPER : ROBOT HUMANOÏDE POUR L’ACCUEIL ET LE DIVERTISSEMENT



Ecrit par

Michelle Vanessa Meguep Sakam

Mars 2017

Expert interne : Andres Perez-uribe

Expert externe :

# Remerciements

Je tiens à remercier de tout cœur les personnes ayant influencé la bonne tenue de ce travail.

Avant tout, je remercie le seigneur tout-puissant qui m’a permis d’arriver à la fin de ce travail, pour la force et l’ouverture d’esprit qui m’ont aidé dans la réalisation de ce travail.

Je tiens ensuite à remercier mon professeur de diplôme Mr Andres Perez-Uribe pour m’avoir donné l’opportunité de travailler sur ce sujet très pationnant, mais aussi pour sa grande disponibilité, ses conseils et la confiance qui m’a accordé.

Je remerciements également Mr Hector Satizabal pour le partage de son expérience avec les humanoïdes, Mr Raphaël Racine et Boody Yeid pour leur participation dans les processus de tests et de mise en situation du robot.

Enfin, je remercie ma famille et mes amis pour leur soutien et leur aide dans la rédaction de mon rapport.

Table des matières

[Remerciements 2](#_Toc477659710)

[Résumé 10](#_Toc477659711)

[1.1. Introduction 12](#_Toc477659712)

[1.2. Contexte 12](#_Toc477659713)

[1.3. Objectifs 12](#_Toc477659714)

[1.4. Cahier des charges 12](#_Toc477659715)

[Chapitre 2: Recherche et documentation 13](#_Toc477659716)

[2.1. Présentation du robot Pepper 13](#_Toc477659719)

[2.2. Caractéristiques techniques 13](#_Toc477659720)

[2.3. Fonctionnalités offertes par Pepper 14](#_Toc477659723)

[2.3.1 Répertoire des fonctionnalités de Pepper 14](#_Toc477659724)

[2.3.2 Certaines fonctionnalités de Pepper vues plus en détail 16](#_Toc477659725)

[Chapitre 3: Recherche et développement 17](#_Toc477659726)

[3.1 Objectif 17](#_Toc477659727)

[3.2 Technologies utilisées 17](#_Toc477659728)

[3.3 Plan 18](#_Toc477659729)

[3.4 Applications 18](#_Toc477659730)

[Application 1 : Pepper donne le nombre de personnes en face de lui 18](#_Toc477659731)

[Application 2 : Pepper dialogue avec ses interlocuteurs 20](#_Toc477659732)

[Application 3 : Pepper détermine l’âge , le genre(homme/femme) d’une personne 21](#_Toc477659733)

[Application 4: Pepper détermine la couleur de vêtements avec OpenCV 23](#_Toc477659734)

[Application 5: Pepper apprend et reconnaît des visages 26](#_Toc477659735)

[Chapitre 4: Projet 28](#_Toc477659736)

[Utiliser le SDK python pour la programmation 30](#_Toc477659737)

[Annexe 1 : Choregraphe 31](#_Toc477659738)

Table des figures

Figure 1-Pepper avec les leds allumés 16

Liste des tableaux

Tableau 1-diagramme UML des cas d'utilisation - Application 1 18

Tableau 2-diagramme UML des cas d'utilisation - Application 3 21

# Résumé

Ce document est un résumé d’un travail de recherche et de développement mené sur le robot Pepper, robot humanoïde conçu par l’entreprise robotique française Aldébaran Robotics et le groupe japonais Softbank. Il se veut être un document qui recense assez d'informations sur le robot dans le but d’être utilisé aussi bien par les utilisateurs finaux, que par toute personne désireuse de créer des applications pour le robot Pepper. Y figurent donc les résultats de travaux de recherche et de documentation, des applications créées pour Pepper et un projet réalisé dont le thème est : « Pepper : robot humanoïde pour l’accueil et le divertissement »

Chapitre 1 : Introduction

## Introduction

Depuis la création des premiers robots au début du 20ème siècle, le domaine de la robotique ne cesse de connaître des innovations, surtout avec l’évolution sans cesse croissante de l’électronique et de l’informatique. Aujourd'hui, il n’y a rien de plus normal que de trouver un robot relayant l’homme dans presque tous les domaines, aussi variés soient-ils. Faut croire qu’Ils ont complètement révolutionné notre quotidien. En effet, on les retrouve dans les chaines de production des industries, l’éducation, l’assistance aux personnes, le transport, relayant l’homme dans les tâches domestiques ou le servant de compagnon etc. Une des branches de la robotique qui mérite tout son intérêt est celle des robots humanoïdes. Conçus à l’image de l’homme, ces robots sont capables de reproduire les gestes et les comportements des humains. L’un de ces robots est le robot Pepper, de l’entreprise Softbank Robotics. C’est le dernier robot conçu par l’entreprise française Aldébaran Robotics et le groupe japonais Solftbank après ses prédécesseurs NAO et Roméo. Lors de précédents travaux de recherche et de développement au sein de la HEIG-VD(Haute École d’ingénierie et de gestion du canton de Vaud), le robot Nao a été utilisé pour des travaux tels que…. En 2016, lorsque l’entreprise Softbank Robotics a commencé la commercialisation de Pepper en Europe, la HEIG-VD s’en est approprié un afin de continuer dans ses travaux de recherche et de développement. Celui-ci promet de nouvelles fonctionnalités, innovantes, telles que la détection des émotions, l’usage d’une tablette incorporée au robot, des caméras plus performantes, plus de capteurs etc. ; des fonctionnalités qui mériteraient d’être explorées et exploitées. L’objectif de ce travail qui se déroule à l’institut IICT de la HEIG-VD dans le cadre d’un travail de Bachelor est donc d’explorer les capacités offertes par ce robot Pepper et ensuite les utiliser pour créer un programme ludique, destiné à l’accueil et au divertissement des personnes.

## Contexte

L’IICT(Institut for information and Communication Technologies) de la HEIG-VD est un institut de l’informatique et des télécommunications qui réalise chaque année près d’une cinquantaine de projets de recherches appliquées et de développement parmi lesquels des projets en rapport avec l’intelligence artificielle et la robotique. Ce travail s’inscrit donc dans la suite logique des travaux menés dans cet institut.

## Objectifs

Les objectifs de ce projets sont :

* Respecter les spécifications énumérées dans le cahier des charges
* Produire des résultats pertinents et détaillées

## Cahier des charges

# Le but de ce travail est de programmer le robot Pepper afin d’interagir avec le public. Il s’agira spécifiquement de penser des scénarios où le robot accueille et amuse le public au moyen de jeux qui impliqueront la programmation de nouveaux comportements chez le robot. Il se décomposera en les phases suivantes :

# • une première phase d’apprentissage où l’étudiante se familiarise avec tous les outils de son travail et explore les capacités du robot

# • Une deuxième phase où l’étudiante programme de nouveaux comportements dans le robot qui respectent des scénarios décrits

# • Une dernière phase qui représente un projet plus conséquent où l’étudiante utilise les connaissances acquises lors des deux premières phases.

# Chapitre 2: Recherche et documentation



## Présentation du robot Pepper

Pepper est un robot à l’apparence humaine, développé en 2012 par le groupe japonais Sofbank et la société française Aldébaran Robotics. Il a été conçu dans le but principal d’être un robot émotionnel ; c’est-à-dire capable de reconnaître et répondre à certaines émotions humaines. Depuis juin 2014, date de son dévoilement au public, Pepper a rejoint de nombreux magasins, entreprises, aéroports etc. où est principalement chargé d’accueillir, informer et distraire les visiteurs ou clients pendant l’attente. Il a également rejoint à partir du 20 juin 2015 des nombreux foyers au Japon en particulier où il sert de robot de compagnie. Pepper est devenu un robot très célèbre que l’on voit apparaître partout :

* Dans les publicités télévisées . Exemple : pour la promotion des nouvelles voitures Zoé, Mégane, Clio de la marque Renault
* Dans l’enceinte de grands magasins ou boutiques de marques. Exemple : 1000 robots Pepper ont été déployés au Japon dans les différents points de vente Nescafé du grand Nestlé dans le but de vendre les machines à café . À Paris, il a rejoint les rayons du magasin Carrefour dans le but de divertir et renseigner la clientèle.
* À la un des articles de presse .Exemple : parution dans les articles du journal suisse « 20 minutes.ch »
* L’invité d’honneur dans de grands salons technologiques. Exemple : au salon Européen de la robotique InnoRobo de mai 2016, au « Disrupt SF » à SanFrancisco en septembre 2016 ; salon organisé par Tech Crunch, le média américain de référence sur les nouveaux techniques. Le 22 mars 2017, il sera présenté lors du « women in robotics Happy Hour » à San Francisco.
* Pepper est aussi beaucoup prisé par des entreprises informatiques et des développeurs passionnés de robotique, qui développent des applications commerciales ou non sur Pepper. La société Softbank Robotics qui la conçoit, développe elle-même plusieurs applications disponibles sur son « App Store ». Pepper est vendu au prix de 15000 euros soient près de 16000 francs suisses mais il est également possible de faire un abonnement(location) et payer mensuellement . Déjà près de 10 000 exemplaires ont été écoulés.

## Caractéristiques techniques

### 



De la taille d’un enfant d’une dizaine d’année, Pepper n’est pas très grand mais à une taille suffisante pour permettre une bonne interaction avec l’homme. Il fait en effet 1m21 de hauteur pour un poids de 28 kg, porte sur son torse une tablette mais ne possède pas de jambes. De manière plus détaillée, il possède les éléments suivants :

* Affichage : Pepper porte sur son torse une tablette à écran tactile de 12 pouces qui lui permet d’offrir toutes les fonctionnalités offertes par une tablette ; navigation sur internet, accès à des contenus multimédias, consultation et envoie des courriers électroniques, installation d’application etc.
* Audition : Pepper possède 4 microphones qui lui permettent de localiser les sons et identifier les émotions contenues dans la voix.
* Expression : pour s’exprimer, Pepper dispose de 2 hauts parleurs.
* Vision: Pepper dispose d’une caméra 3D pour la profondeur et de deux autres HD qu’il utilise pour s’imprégner de l’environnement qui l’entoure. Il est ainsi capable d'identifier les personnes, les objets, les mouvements et les différencier, mais aussi reconnaître les émotions qui se dessinent sur les visages et éviter les obstacles.
* Déplacement : Avec ses 3 roues omnidirectionnelles à la place des jambes, Pepper se déplace facilement. Il est capable d’effectuer des rotations de 360° sur lui-même. Sa vitesse maximale est fixée à 3 km/h. Aussi, La matière principale utilisée pour fabriquer Pepper est le polyuréthane. C’est une matière souple qui permet une fluidité dans ses mouvements.
* Acquisition des données : Pepper dispose de plusieurs capteurs de différents types qui lui permettent de collecter les données sur son entourage. Il dispose notamment de 5 capteurs tactiles(3 dans la tête et 2 dans les mains) qui lui permettent d’être sensible au toucher, de 2 sonars, 6 lasers, un capteur gyroscopique pour éviter les obstacles et rester en équilibre, un capteur incorporé dans sa batterie qui permet d’avoir les informations sur son état de charge et sa température.
* C onnexion réseau: Pepper est capable de se connecter à internet via une connexion Ethernet ou par Wi-Fi(IEEE 802.11a/b/g/ns)Autonomie : Pepper dispose d’une batterie lithium-ion d’une capacité de 30 Ah/795Wh qui lui garantit une grande autonomie de près d e12h.
* Dégré de liberté : Pepper a un degré de liberté égal à 17 ; ce qui lui permet de faire plusieurs mouvements et des plus complexes car partant de sa tête à sa base, ses articulations sont très mobiles

## Fonctionnalités offertes par Pepper

Il s’agit dans cette section, de présenter dans un premier temps les fonctionnalités offertes par le robot Pepper et ensuite les exposer de manière plus détaillée.

### Répertoire des fonctionnalités de Pepper

Pepper et son ancêtre Nao partage de nombreuses fonctionnalités en commun. D’ailleurs, il a été conçu en se basant sur le robot Nao déjà existant. Comparer ces deux robots est donc un bon moyen pour lister les fonctionnalités de Pepper mais aussi voir ce qu’il a de nouveau et comprendre l’interêt qu’il succite. Les différences entre les deux robots sont donc axés sur ce que a Pepper de différent de Nao

Nao et Pepper possèdent tous deux les capacités suivantes :

1. Communication

Les robots Nao et Pepper sont capables d’entendre et parler avec la possibilité de choisir la langue de communication. La langue par défaut est l’anglais pour les deux robots mais il est possible d’installer jusqu’à 7 autres langues pour Pepper et 18 pour Nao. La séconde langue chosie après l’anglais est gratuite et le choix définitif et les autres installées par la suite sont payantes. En outre, ils communiquent aussi avec des gestes et des leds placés sur la partie supérieur de leur corps.

1. Bouger les membres ou se déplacer

Ils peuvent bouger tout leurs membres pour effectuer des gestes, attraper des objets, et se déplacent en évitant de se heurter à des obstacles. Pepper ne possède pas de pieds, il ne peut donc pas s’asseoir.

1. Danser

Ils peuvent enchainer des gestes et exécuter ainsi des chorégraphies faciles.

1. Détecter et reconnaître un visage, une voix, des objets

A l’aide de leurs cameras haute définitions et des algorithmes sophistiqués pour le traitement d’image et la reconnaissance, les robots Nao et Pepper sont capables de détecter lorsqu’un son est émis, la presence d’une personne, d’objets ou d’images, de les mémoriser et ensuite les reconnaître lorsqu’il les perçoivent ou voient pour une seconde fois.

1. Se connecter à internet

Chaque robot de Softbank Robotics possède une page internet accéssible via l’adresse IP à laquelle le robot est connecté. On peut à partir de celle-ci configurer le robot. On peut ainsi changer la langue de communication, mettre à jour les applications, ajuster le volume etc.

1. Installer ou programmer de nouvelles applications

La société Softbank a dans son app store une panoplie d’applications pour Nao et Pepper, prêts à l’emploi et dont il est possible d’installer directement dans les robots. Cependant, il est possible de dévéloppement ses propres applications et de les installer dans les robots. En effet, les deux robots sont open source et supportent de nombreux langages et plateformes de programmation.

Mais ce qui distingue Pepper de Nao est le suivant :

1. Les émotions : Pepper est capable d’analyser le visage ainsi que la voix de ses interlocuteurs pour décéler des émotions ; il peut ainsi modifier son comportement en conséquence et miner en retour des émotions.
2. La communication : Pepper dispose en plus sur son torse d’une tablette tactile lui permettant d’interagir avec l’homme. Il peut ainsi proposer des ménus à sélectionner, faire tourner des applicatins sur sa tablette etc.
3. La taille : Pepper est nettement plus grand que Nao. Il a une hauteur de 1m21 contrairement à Nao qui ne fait que 58 cm ; ce qui lui donne une apparence plus proche de celle de l’homme et rend l’interaction homme-machine beaucoup plus naturelle.
4. La mobilité

Pepper dispose pour sa mobilité de trois roues omnidirectionnelles qui lui permettent de se déplacer plus vite que Nao, 3km/h contre 1km/h. Aussi, les actions de Nao sont encore assez mécaniques. On peut d’ailleurs entendre le bruit de ses membres lorsqu’il bouge ou se déplace. C’est une caractéristique qui a été amélioré chez Pepper. Il est plus souple et ses actions moins mécaniques

1. Les dégrés de liberté
2. L’équilibre

Pepper tient très bien en équilibre contrairement à Nao qui peut tomber en essayant certains déplacements ou en restant trop longtemps débout ; il se fatigue. Lorsque Pepper est fatigué, il se met en pause. Pour cela, il se replit sur ses hanches, baisse la tête et stop le programme en cours. Les leds et sa tablette reste toute fois allumés indiquand qu’il n’est pas éteint.

1. Le poids

Nao a un poids de 58 kg contrairement à Pepper qui ne fait que 28 kg. Il est donc plus léger et plus facile à déplacer.

1. Autonomie : Pepper a une baterie d’une autonomie de 12h contrairement à NAO dont la baterie ne peut faire que 90 mins hors charge.

### Certaines fonctionnalités de Pepper vues plus en détail

Le but de cette section est d’apporter des informations plus précis sur certaines fonctionnalités de Pepper

1. La vision

Pepper possède une camera 3D pour la profondeur qui peut produire une image d’une résolution allant jusqu’à 320x240 pixels pour 20 trames par seconde et deux autres cameras 2D d’une résolution maximale de 2560X1080 pixels pour 5 trames par séconde contrairement à Nao qui ne possède que deux cameras HD d’une résolution de 1280X960 pixels pour 30 trames par séconde. La résolution des caméras de Pepper permet de mieux simuler la vision humaine.

1. La communication
2. Communication corporelle
3. Les gestes

Pepper communique aussi à travers des gestes. En effet, à certains mots, sont associés des gestes spécifiques .Exemple : quand pepper dit bonjour, il l’accompagne d’un geste de salutation. Dependant de la langue de communication choisie, les gestes sont différents afin de s’adapter aux différentes cultures. Si la langue est par exemple le japonais, il baisse la tête puis la lève. En français, il ouvre les deux bras comme pour acceuillir. Le programme de base vendu avec Pepper comporte une liste de gestes par défaut associés à des mots spécifiques comme le bonjour, l’au revoir, le merci, le s’il vous plaît etc. mais il est possible de modifier tout ceci ou d’associer à d’autres mots des gestes spécifiques. Il faut pour cela programmer Pepper.

1. Les Leds

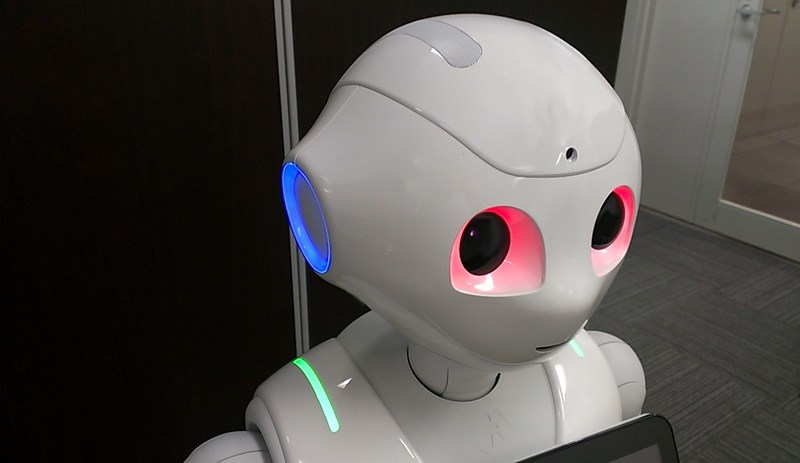


Figure 1-Pepper avec les leds allumés

Comme on peut le voir sur la figure ci-dessus, Pepper a une led dans chacun de ses yeux, de ses oreilles et sur ses épaules. Celles-ci lui permettent de communiquer à chaque instant les informations sur son état à travers un code de couleur dont en voici quelques exemples :

-Lors de la détection d’image, d’objet ou de sons, les leds des yeux prennent la couleur verte si la détection à reussi et rouge sinon

-Lors d’un dialogue avec Pepper, lorsque les leds de ses yeux deviennent blueues et tourne en boucles, cela veut indiquer que Pepper est en mode d’écoute. Lorsqu’un son est ensuite détecté, les leds des yeux prennent la couleur verte.

//ajouter plus d’informations à ce sujet

1. Communication Vocale

…parler de la voie à la base qui est celle d’une femme mais qui peut ête ajusté(plus grave ou égu, lent ou vite)

1. Communication des émotions

…traduit ses émotions dans ses textes et gestes(happy plus égu, sad plus triste)

1. La vision

…pepper se récule quand on s’approche trop de lui, chercher et suis du regard, distance maximale de vision, angle de vision, influence de l’éclairage, influence de la couleur de peau,

1. Connection internet

…possibilité de connecter un cable ethernet(expliquer la procédure), connection par wifi, accès à la page internet, choix de la langue,compte aldebaran

1. Documentation

…aldebaran, communité,

1. Mobilité

…fermer le port de charge pour pouvoir se déplacer, ..préférer des surfaces lisses et planes,

1. Developpement d’applications pour Pepper

Appstore aldebaran modifiable ou partir du crach

1. Languages et plateformes supportés
2. Open source, interface programmable(tablette)
3. Connection des librairies externes
4. Modes de programmation
5. Outils disponibles
6. Les Apis

# Chapitre 3: Recherche et développement

## Objectif

Le but de cette partie est de montrer comment créer ses propres applications et les faire tourner dans le robot Pepper. Pour ce faire, quelques applications ont été dévéloppées en respectant des scénarios décrits à l’avance.

## Technologies utilisées

Pour programmer ses comportements dans Pepper, les technologies suivantes ont été utilisées :

* Langage de programmation : Le langage de programmation utilisé est le python 2.4. Le choix de ce langage s’explique par le choix de la platforme de programmation Choregraphe, le niveau d’abstraction du langage et aussi par le désir d’apprendre un nouveau langage.
* Os : toutes les applications ont été dévéloppées et testées uniquement sur windows 7
* Plateforme de développement : Pour la programmation ainsi que le test des applications dévéloppées, l’outil Choregraphe(version 2.4) a été utilisé. Voir le document à l’nnexe 1 : Choregraphe pour plus de détails
* Framework : Pour la programmation des comportements dans Pepper, le framework NAOqi (version 2.4) a été utilisé. Voir le document àl’nnexe 2 : Naoqi pour plus de détails
* Système de versions : pour la gestion des versions, le logiciel Git a été utilisé.
* Diagrammes : Création des diagrammes des cas d’utilisation à l’aide du logiciel ArgoUML.

## Plan

La section suivante présente les différentes applications qui ont été dévéloppées. Pour chaque application, le plan suivant a été élaboré afin de mieux le présenter :

1. Objectif : il s’agit de décrire de manière succinte l’objectif de l’application.
2. Description du scénario : description de manière détaillé du scénario de l’application. Il s’agit de dire à travers des mots, ce qui a été codé dans l’application. Dans certains cas,des diagrammes UML ont été utilisés en guise d’ illustration
3. Analyse : Cette section a pour but de trouver le(s) problème(s) du scénario précédenment décrit et de proposer des éléments(modules, fonctions, méthodes etc.) qui permettraient de les résoudre. Il s’agit donc de répondre aux questions : pour satisfaire ce scénario, qu’est-ce-qui devrait être implémenté ? avec quoi peut-on l’impémenter ?
4. Code : Il s’agit dans cette section de présenter le programme dévéloppé pour cette application
5. Résultat(s) : « screenshot » des résultats obtenus.

## Applications

### Application 1 : Pepper donne le nombre de personnes en face de lui

##### Objectif

L’objectif de cette application est de programmer Pepper pour qu’il puisse d’une part détecte la présence humaine et d’autre part donner les informations sur ce qu’il voit.

##### Description du scénario

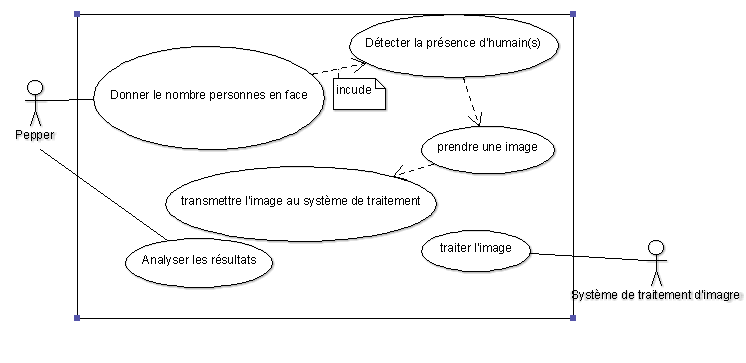


Tableau 1-diagramme UML des cas d'utilisation - Application 1

* Préconditions : Le robot doit être allumé, l’application chargé dans le robot, l’environnement bien éclairé
* Scénarii :

Scénario principal :

1. Pepper regarde en face de lui .
2. Détecte une présence humaine. Pour cela, il doit prendre une image de ce qui est en face de lui et la transmettre au système de traitement d’image pour y reconnaître une ou plusieurs personnes
3. Reçoit les résultats et les analyses
4. Donne le nombre de personnes présentes

Scénario alternatif :

2.a Si aucune personne n’est détecté, Pepper le dit

Scénario d’exception :

Si un évenement survient comme par exemple, le « timeout » atteint, Pepper indique la fin de l’interaction.

Performance attendue :

* Le temps de réponse doit être au plus 20 sécondes après le démarage.

Compléments :

Les cas d’utilisation « prendre une image », « traiter une image », « transmettre l’image » ne sont pas étudiés plus en détail dans ce scénario. Il seront décrits dans la section analyse suivante

##### Analyse

Pour réaliser le scénario décrit précédenment, il faut implémenter les éléments suivants :

1. La détection des personnes

Comme l’a montré le diagramme des cas d’utilisation de la figure 2, pour que Pepper donne le nombre de personnes en face de lui, il doit tout d’abord arriver à détecter la présence d’humains. Dans Naoqi, le module qui permet de faire ceci est ALFaceDetection et les informations sur les personnes détectées sont dans la variable FaceDetected. Pour en savoir plus sur le module ALFaceDection et la détection de personnes, se rendre au document à l’annexe 4 : ALFaceDetection. Il est également possible d’utiliser un autre module de Naoqi pour la détection de personne. Il s’agit de ALPeoplePerception, un module proche de ALFaceDetection. Il a été choisi de le présenter dans d’autres applications. Pour en savoir plus sur ce module, se rendre à l’annexe 5 : ALPeoplePerception

1. Analyse des résultats

Une fois les personnes détectés, il faut déterminer le nombre de personnes trouvées. Il faut donc s’intéresser à la variable FaceDetected dans laquelle a été stockée les informations sur les personnes vues par le robot.

1. Donner le nombre de personnes en face du robot

Le nombre de personnes présentes doit être donné oralement. Pour faire parler Pepper, le module ALTextToSpeech de Naoqi et sa méthode « say » sont suffisantes. Voir le document à l’Annexe 6: Fonctionnement de AltextToSpeech pour comprendre le fonctinnement de ce module. Cependant, il est également possible d’utiliser le module ALAnimatedSpeech qui permet au robot de dire des textes accompagner des gestes. Pour en savoir plus sur ce module, aller à l’Annexe 7 : ALAnimatedSpeech

1. Gestion du « timeout »

Il a été indiqué qu’en cas de « timeout » atteint, le robot doit l’indiquer . Ce timeout doit être implémenté. Le framework qi propose une méthode « async » qui permet de faire des appels asynchrones. Ceci peut être utilisé pour faire un appel asynchrone après un temps défini sur la fonction timeout afin de stopper l’application. Voir l’annexe Annexe 8: Le framework Qi afin d’en apprendre plus sur ce module et cette méthode.

##### Code

Pour voir le code de cette application, se rendre à l’annexe Annexe 8 : code de l’application1 :Pepper donne le nombre de personnes en face

##### Résultat(s)

### Application 2 : Pepper dialogue avec ses interlocuteurs

##### Objectif

L’objectif de cette application est d’apprendre à programmer des dialogues dans le robot Pepper. Ceux-ci lui permettent d’entretenir des conversations.

##### Description du scénario

Le scénario de cette application a été décrit sous la forme d’un échange entre l’homme et le robot ; c’est-à-dire, ce que l’homme dit et le robot répond et vis-versa.

Humain : Bonjour Pepper

Pepper : Bonjour

Humain : Comment vas-tu ?

Pepper : Je vais très bien et vous ?

Humain : Bien merci !

Pepper : Je me rejouis de le savoir

Humain : Parle moi de tes origines

Pepper : J’ai été conçu par la société française Aldebaran en 2014. La connaissez-vous ?

Humain : Non

Pepper : C’est la même qui a conçu les robots Nao et Roméo

Humain : Oui

Pepper : Ok !

##### Analyse

Pour implémenter ce dialogue comme tout autre dans Pepper il faut utiliser le module ALDialog de Naoqi qui permet de créer des dialogues. Il définit des règles de syntaxe et d’écriture qu’il faut connaître pour créer et enrichir les dialogues. Au document annexe Annexe 9 : Créer des dialogues, il est expliqué de manière détaillée comment créer et écrire des dialogues avec ALDialog

##### Code

Le code de cette application est disponible à l’annexe Annexe8 : Code de l’application 2 : Pepper dialogue

##### Résultat(s)

### Application 3 : Pepper détermine l’âge , le genre(homme/femme) d’une personne

##### Objectif

L’objectif de cette application est de programmer Pepper pour analyser les caractéristiques faciales des personnes.

##### Description du scénario

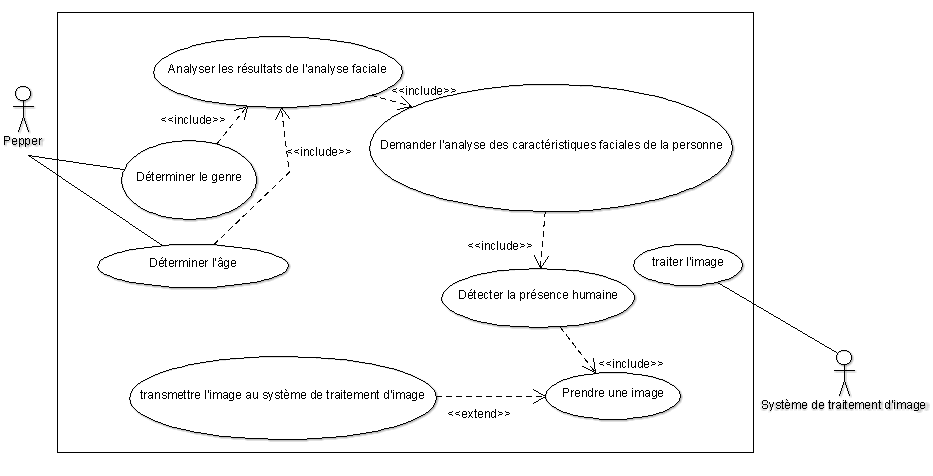


Tableau 2-diagramme UML des cas d'utilisation - Application 3

* Préconditions : Le robot doit être allumé, l’application chargé dans le robot, l’environnement bien éclairé. Aussi, la personne dont il doit être déterminer le genre et l’âge doit être la plus proche du robot et établir un contact visuel avec lui.
* Scénarii :

Scénario principal :

1. Pepper regarde en face de lui .
2. Détecte une présence humaine. Pour cela, il doit prendre une image de ce qui est en face de lui et la transmettre au système de traitement d’image pour y reconnaître une ou plusieurs personnes
3. Demande une analyse des caractéristiques faciales
4. Reçoit les résultats et les analyse
5. Donne le genre et l’âge de la personne en face

Scénario alternatif :

2.a Si aucune personne n’est détecté, ou plusieurs, Pepper le dit.

5.a Si la personne en face a été déjà été analysée, le robot fait une nouvelle analyse et la compare avec la précédente.

Scénario d’exception :

Si un évenement survient comme par exemple, le « timeout » atteint, Pepper indique la fin de l’interaction.

Performance attendue :

* Le temps de réponse doit être au plus 20 sécondes après le démarage.
* Le robot doit reconnaître une personne déjà analysée.

##### Analyse

Pour réaliser le scénario décrit précédenment, il faut implémenter les éléments suivants :

1. La détection des personnes

Il est important de détecté la présence d’une personne pour pouvoir analyser ses caractéristiques. Dans l’application 1 ci-dessus, il a été mentionné un autre module qui permet également de faire la détection de personnes. Il s’agit de ALPeoplePerception. Ce module permet d’obtenir la liste des personnes visibles à chaque changement de la population en face du robot ; ce qui peut donc être utilisé par Pepper pour savoir si des personnes sont présentes. Pour d’amples informations sur le module ALPeoplePerception, se rendre à l’annexe Annexe 5 : ALPeoplePerception

1. Demander l’analyse des caractéristiques faciales

Pour effectuer l’analyse ds visages qu’il voit, Pepper soit explicitement demander l’analyse. Pour ce faire, il existe dans Naoqi le module ALFaceCharacteristics qui permet de donner cet ordre d’analyse et de récolter les informations attendues. Il faut aussi savoir que ce module appelle automatiquement le module ALPeoplePerception. Il n’est donc plus nécessaire de l’appeler explicatement. Voir l’annexe Annexe 6 : ALFaceCharacteristiques pour en apprendre plus.

1. Analyse des résultats

Les résultats de l’analyse des caractéristiques faciales sont stockés en mémoire sous "PeoplePerception/Person/id/ " où id est celui de la personne dont on a demandé l’analyse faciale.

1. Donner le genre et l’âge de la personne

Pepper doit donner oralement le genre et l’âge de la personne. Il faut donc une fois de plus ici faire appel au module ALTextToSpeech et sa méthode say. Voir l’Annexe 6: Fonctionnement de AltextToSpeech pour en savoir plus.

1. Reconnaître une personne dont l’analyse a déjà été faite

Il existe une fonctionnalité de Pepper qui consiste à reconnaître les personnes qu’il a déjà vu une fois. Il ne s’agit pas dans ce cas d’exploiter cette fonctionnalité. Elle le sera dans une autre application. Pour reconnaître ici les personnes déjà analysée, il est possible de stocker dans un tableau, pour chaque id de personne, son genre et son âge. Lors du prochain passage, utiliser l’id pour reconnaître la personne et comparer sa nouvelle analyse avec la précédente.

1. Gestion du timeout

Il a été indiqué qu’en cas de « timeout » atteint, le robot doit l’indiquer . Ce timeout doit être implémenté. Le framework qi propose une méthode « async » qui permet de faire des appels asynchrones. Ceci peut être utilisé pour faire un appel asynchrone après un temps défini sur la fonction timeout afin de stopper l’application. Voir l’annexe Annexe 8: Le framework Qi afin d’en apprendre plus sur ce module et cette méthode.

##### Code

##### Résultat(s)

### Application 4: Pepper détermine la couleur de vêtements avec OpenCV

##### Objectif

L’objectif de cette application est de programmer Pepper pour qu’il interagisse avec le librairie openCV la libairie de traitement d’image en temps réel afin de donner la couleur des vêtements des personnes en face de lui. Il s’agit ici de montrer la possibilité d’ utiliser des programmes externes avec Pepper

##### Description du scénario

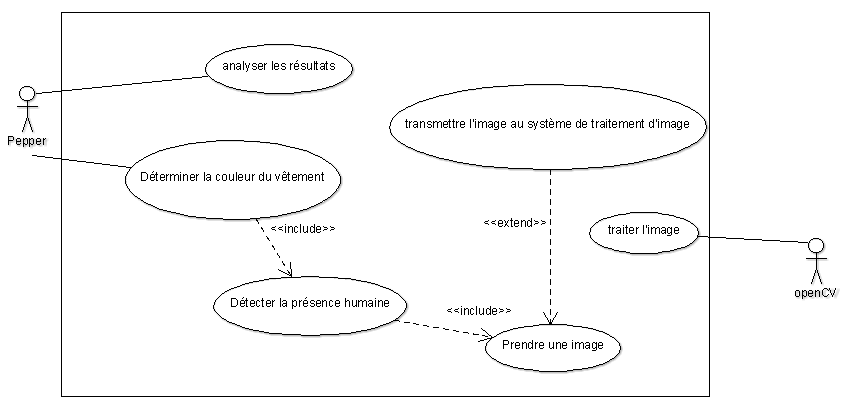


Tableau 4 - diagramme UML des cas d'utilisation - Application 4

* Préconditions : Le robot doit être allumé, l’application chargé dans le robot, l’environnement bien éclairé. Aussi, la personne dont il doit déterminer la couleur du vêtement doit être la plus proche du robot et établir un contact visuel avec lui.
* Scénarii :

Scénario principal :

1. Pepper regarde en face de lui .
2. Détecte une présence humaine. Pour cela, il doit prendre une image de ce qui est en face de lui et la transmettre au système de traitement d’image openCV
3. Reçoit les résultats et les analyse
4. Donne la couleur du vêtement de la personne

Scénario alternatif :

2.a Si aucune personne n’est détecté, ou plusieurs, Pepper le dit.

Scénario d’exception :

Si un évenement survient comme par exemple, le « timeout » atteint, Pepper indique la fin de l’interaction.

Performance attendue :

* Le temps de réponse doit être au plus 20 sécondes après le démarage.

##### Analyse

Pour réaliser le scénario décrit précédenment, il faut implémenter les éléments suivants :

1. La détection des personnes

Il est important de détecté la présence d’une personne pour pouvoir analyser son vêtement. Dans les applications 1 et 2 ci-dessus, il a été présenté deux moyens de détecter la présence d’humains. Une autre moyen est d’utiliser le module ALBasicAwaresness de Naoqi. C’est un module qui permet de jouer avec la sensibilité du robot face à son environnement. Cette sensibilité concerne les sons, les movements, les personnes, les touchés. La présence d’une personne est donc considérée comme un stimulus. Il suffit de souscrire à l’événement « ALBasicAwareness/HumanTracked » qui sera appelé lorsqu’une personne est détectée. Pour éviter que cet événement ne soit appelé qu’une seule fois et éviter que le robot ne réagisse plusieurs fois face à la même personne, connecter la sortie *onStopped* de la boîte avec l’entrée *onStop* comme sur l’image suivante :

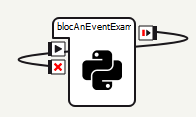


Figure 5-Bloquer l’évenement « ALBasicAwareness/HumanTracked »

1. Prendre une image et la transmettre à openCV

OpenCV est une librairie externe à Choregraphe et au robot. Pour pouvoir profiter de ses services de traitement d’image, il faut lui transmettre l’image à traiter. Heureusement dans Naoqi, il existe un module qui permet de manipuler la prise et la transmission d’images dans Pepper. Ce module est ALVideoDevice. Il permet d’obtenir une image d’une camera du robot. Il suffit d’enregistrer cette camera au module ALVideoDevice et de récupérer l’image renvoyée. Cette image n’est pas encore l’image recherchée mais un buffer contenant d’autres informations sur l’image telles que la hauteur de l’image, la largeur, l’id de la camera utilisée etc. C’est la 7ème case de ce tableau qui contient les données de l’image. Pour en savoir plus sur ce module, aller é l’annexe Annexe 8 : ALVideoDevice.

1. Analyse des résultats

La librairie openCV renvoie la couleur dominante de l’image qui lui a été transmise. Pour en savoir plus le fonctionnement de la librairie openCV, aller é l’annexe Annexe 8 : openCV.

1. Donner la couleur du vêtement

Une fois de plus, le module AltextToSpeech serait necessaire pour dire la couleur trouvée.

1. Gestion du timeout

Pareil que dans les applications précèdentes.

##### Code

##### Résultat(s)

### Application 5: Pepper apprend et reconnaît des visages

##### Objectif

L’objectif de cette application est de programmer Pepper pour qu’il fasse de la reconnaissance faciale. Il s’agit de montrer comment apprendre un visage à Pepper et ensuite le reconnaître

##### Description du scénario

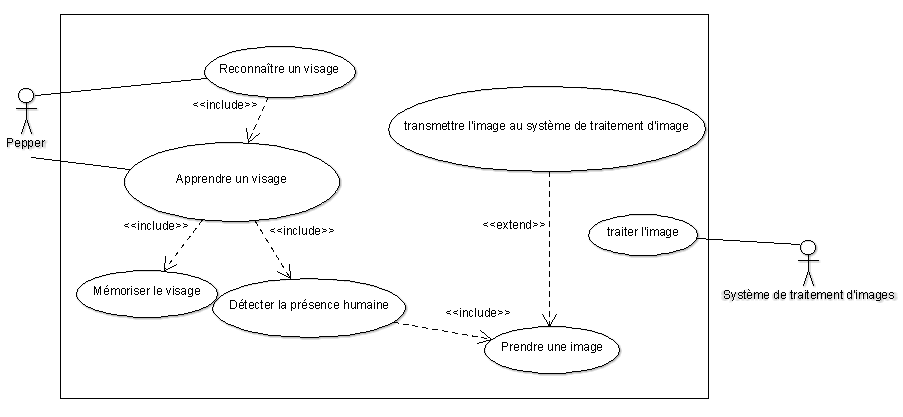


Tableau 6- Diagramme UML des cas d'utilisation - Application 5

* Préconditions : Le robot doit être allumé, l’application chargé dans le robot, l’environnement bien éclairé. Pour de meilleurs performances, éviter le port des lunettes, essayer d’avoir une expression faciale neutre(ne pas sourire ou faire des grimaces), se place à une distance proche du robot(environ entre 1 et 1.5 m)
* Scénarii :

Scénario principal :

1. Pepper détecte une présence humaine. Pour cela, il doit prendre une image de ce qui est en face de lui et la transmettre au système de traitement d’image
2. Reçoit les résultats et les analyse
3. Si le visage de la personne lui est familier, il le mentionne
4. Sinon, si il ne reconnaît pas le visage
5. Affiche un formulaire sur sa tablette
6. Demande à son interlocuteur d’entrer son nom
7. Apprend son visage en le mémorisant avec le nom saisi

Scénario alternatif :

4.2.a Si l’utilisateur clique sur le bouton supprimer du formulaire, son image n’est pas mémorisé.

Scénario d’exception :

Si une erreur est survenue lors du chargement de la tablette, l’interlocuteur est avisé.

De même, si une erreur survient lors de la mémorisation du visage, celui-ci est réappris

Performance attendue :

Le robot doit arriver à reconnaître toutes les personnes déjà rencontrée.

##### Analyse

Pour réaliser le scénario décrit précédenment, il faut implémenter les éléments suivants :

1. La détection des personnes

L’implémentation de la détection de personnes peut se faire à l’aide du module ALFaceDetection comme à l’application1.

1. L’affichage du formulaire sur la tablette

Le formulaire à afficher peut se limiter à un simple champ de texte avec un titre qui demande la saisie du nom. Aussi, la tablettre de Pepper est par défaut en mode de veille. Il faut l’allumer avant de pouvoir effectuer l’affichage du formulaire. Ces deux actions peuvent être implémenter grâce au module ALTabletService de Naoqi qui offre tout un éventail de solutions pour manipuler la tablette de Pepper. Pour en savoir plus sur ce module, aller à l’annexe Annexe 8 : Utiliser la tablette de Pepper

1. L’enregistrement du visage en mémoire

Il est possible de connaître le nombre de visages déjà connus par Pepper. C’est la méthode getLearnFacesList du module ALFaceDetection. C’est ce module module qui permet également de mémoriser les visages et de faire la réconnaissance. Pour mémoriser, il suffit d’utiliser la fonction learnFace et reLearnFace pour reapprendre le même visage.

1. La reconnaissance des visages

Lorsque le module ALFaceDetection est utilisé pour détecter la présence des personnes, le système de traitement d’image stocke les résultats dans la variable FaceDetected. Il s’agit d’un buffer qui contient les informations sur les visages trouvées dans l’image transmise au système de traitement d’image. L’analyse de ce buffer permet de déterminer si une personne est connue par le robot ou pas. Pour en savoir plus sur l’apprentissage et la reconnaissance d’image par Pepper, se rendre au document annexe Annexe 8 : ALFacedetection

##### Code

Pour cette application, la boîte « Face Reco » de Choregraphe a été reprise et le code ajouté afin de respecter le scénario de cette application.

Code initiale de la boîte :

Code de l’application finale après ajout :

##### Résultat(s)

# Chapitre 4: Projet

Les robots Pepper et NAO tourne sur le système d’exploitation appelé NAOqi. Il s’agit d’un système Linux

basé sur Gentoo qui a été dévéloppé par l’entreprise Aldebaran elle-même afin de repondre spécifiquement aux besoins de ces robots et faciliter l’interaction avec eux. Toutesfois, les systèmes Windows et Mac OS sont également compatibles.

Il existe plusieurs manière de devélopper les applications pour les robots Pepper et NAO en utilisant des langages et plateformes différentes.

#### Les différents modes de programmation

Les robots NAO et Pepper peuvent être programmés de deux manières. La première consiste à créer des applications à l’aide des outils dédiés tels que Choregraphe, les SDK etc. et de les installer directement dans la tête du robot qui les interprêtera et exécutera les comportements élaborés. La deuxième quant à consiste à créer des programmes externes ; c’est-à-dire directement sur notre ordinateur, sans utiliser les outils dédié à la programmation des robots et de contrôler ces derniers depuis ces programmes.

#### Les langages de programmation

Pepper et NAO peuvent être programmés dans les langages suivants : C++, C, Python, Java , Urbi,MATLAB,.NET

* + 1. Les plateformes
       1. Webot

###### Programmer Pepper

Dans cette section, nous allons entammer la partie programmation de Pepper. Nous allons utiliser pour cela le langage python, Choregraphe pour créer des scripts et le robot reel pour exécuter notre code.

* 1. NAOqi

Aldebaran met à disposition des utilisateurs le framework NAOqi du même nom que le système d’exploitation qui tourne dans ses robots. NAOqi est un framework programmable utilisé pour programmer les robots de la société. En effet, il est possible d’utiliser le C++, JAVA, Pyhon, javascript etc. C’est également un framework multiplate-forme (Windows, Linux, MacOS).

1. Fonctionnement

NAOqi est basé sur un système de modules. En effet, toute action du robot est basé sur module. Exemple : pour gérer tout ce qui concerne la motion du robot est regroupé dans le module ALMotion. Il contient toutes les méthodes, évenements et signaux en rapport avec la locomotion du robot.

1. NAOqi - documentation

, une plateforme de documentation. On y trouve de la documentation sur les robots, les dernières versions de produits, les APIs pour le developpement des applications, sur l’installation et l’utilisation des outils de developpement et de simulation etc. Pour y accéder, se rendre à l’adresse suivante : <http://doc.aldebaran.com/2-4/>. Il s’agit de la dernière mise à jour qui corespond à la dernière version de l’OS. En effet, il existe

1. Scénario 1 : Pepper détecte un visage et dit : «Bonjour! Je m’appelle Pepper !Bien venu aux portes ouvertes de la HEIG-VD.Comment puis-je vous aider ?» ; tout ceci accompagné des gestes.

Pour ce faire, nous allons programmer une nouvelle boxe du nom de « bienvenue » dont le but sera d’exécuter ce scénario. Nous avons besoin de deux modules de l’API NAOqi : ALFaceDetection et ALAnimatedSpeech. Le premier a pour but de permettre au robot de détecter des visages et le second de dire le texte animé.

Le module ALFaceDetection fonctionne sur un système de callback. L’idée ici est de lancer l’évenement qui permet de dire le texte animé lorsque des visages sont reconnus.

//décrire le code de cette boxe.

Le robot ne prononce pas bien HEIG-VD. Il faudrait écrire dans le script H E I G V D pour qu’il arrive à bien prononcer.

1. Scénario 2 : Pepper exécute un dialogue

Le but ici est d’améner le robot à dialoguer avec les personnes en face de lui.Les dialogues peuvent être classés par topics.Nous allons par exemple définir les topics suivants :

Topic 1 : Accueil

* : scénario 1
* :Bonjour Pepper. Comment allez-vous ?
* : Bien et vous ?
* : Bien merci !

Topic 2 : Renseignement

* : Vous pouvez me dire de quel évenement il s’agit ?
* :Biensûr !il s’agit des portes ouvertes organisées comme chaque année par la HEIG-VD.
* :Quel est le thème ?
* : Cette année, les portes ouvertes portent sur le thème du bien-être. Il s’agit notament de découvrir nos nombreux projects dont celui :
* du fauteuil qui capte l’activité physique de la personne assise
* Un suivi temps réel de courses cyclistes
* Une écoute de routes silencieuses
* Un suivi d’une pilule dans le corps humain
* Des solutions domotiques orientées utilisateurs
* Un parcours découverte sur la face cachée des objets

Nous allons créer une nouvelle boxe appelée « Dialogue » pour ce scénario.

Nous avons besoin dans cette partie du module «AlDialog » qui permet de créer des topic et des dialogue

### Utiliser le SDK python pour la programmation

Il existe plusieurs avantage à utiliser le SDK ici celui de python pour programmer les robots NAO et Pepper :

* Pour faire le traitement d’image
* Les fonctions du framework NAOqi sont bloquantes ; c’est-à-dire qu’il faudrait attendre la fin d’une méthode pour exécuter la deuxième. Avec le SDK, ce problème est résolu. La méthode posst permet de créer un nouveau thread et de le joindre à celui existant afin que les deux s’exécutent en même temps.

Exemple :



Dans cette image où nous souhaitons que le robot marche en parlant, après avoir instancier le module ALMotion qui permer au robot de marcher, lui avons appliqué laméthode post afin de créer un nouveau thread et exécuter les deux actions. Le robot va donc marcher en disant le texte entre guillemets.

Si toutefois nous souhaitons qu’il exécute une action après une autre, il faut dans ce cas utiliser la méthode wait. L’exemple suivant illustre bien ce que nous venons de dire.



Nous avons stocké dans la variable threadMove le thread crée dans l’exemple précédent. En applquant au module « motion » la méthode wait avec en paramètre le nom du thread threadMove dont on doit attendre la fin de l’exécution et le temps, le robot va finir d’avancer avant de dire le texte qui indique qu’il est arrivé à destination.

Lien utile pour rappot : https://web.wpi.edu/Pubs/E-project/Available/E-project-011415-130837/unrestricted/NAO\_Robots\_MQP\_2014.pdf

## Annexe 1 : Choregraphe

Choregraphe est une plateforme crée par le groupe SoftBank Robotics pour aider les programmeurs dans le developpement d’applications pour leurs robots humanoïdes. Il est à la fois une plateforme de programmation et de simulation. C’est d’ailleurs un très bon outil quand on a jamais programmer un robot.

En effet, choregraphe a été conçu pour aider les programmeurs sans expérience en robotique à developper des applications sans trop s’encombrer de code. Il permet alors à l’aide des boîtes préprogrammés de créer des circuits et exécuter des comportements. Il permet en même temps de se connecter à un robot réel ou virtuel afin de tester les comportements programmés. Cependant, il est possible pour des programmeurs , d’enrichir les boîtes existantes avec leur propre code ou de partir de rien et de programmer soit même de nouvelles boîtes en faisant appel à des modules et fonctions offertes par le framework NAOqi. Il est également possible de connecter des librairies externes à choregraphe comme par exemple la librairie OpenCV pour ce citer que celle-ci, qui permet de faire du traitement d’images.

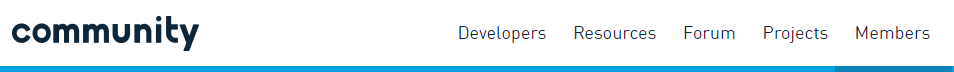
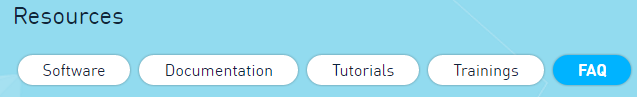
##### Installation

Pour installer Choregraphe, vous devez vous assurer que votre ordinateur dispose des éléments suivants :

* Une connection à internet
* Au 1.5 GHZ de CPU
* 512 MB de RAM disponible
* La carte graphique OpenGl installée

La marche à suivre pour l’installation de ce programme sur Windows est la suivante :

Télécharger la dernière version sur le site web de la communauté Softbank Robotics. Il faut avoir un compte et s’y connecter pour pouvoir télécharger le logiciel.

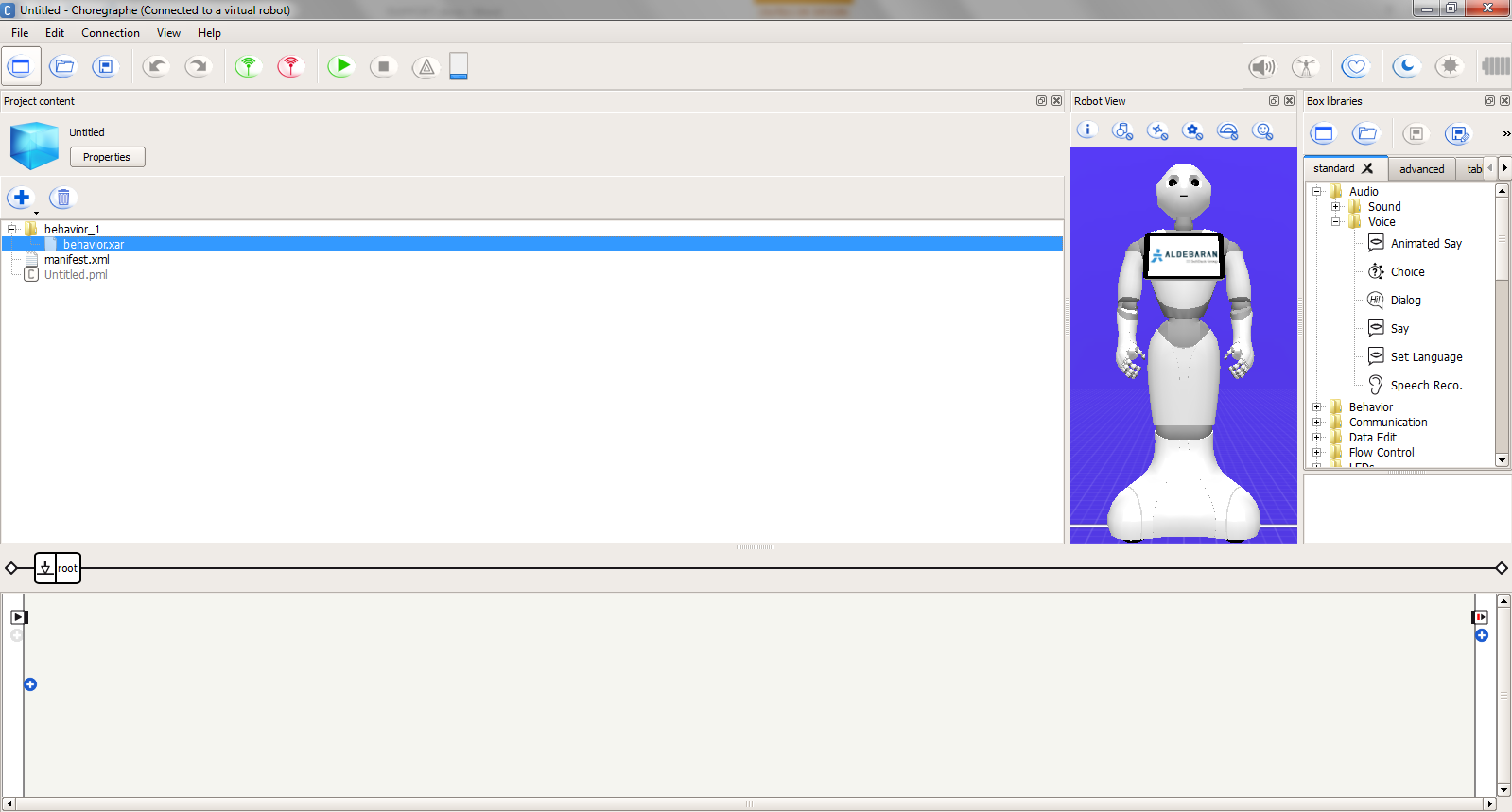
* Rendez vous à l’adresse suivante : <https://community.ald.softbankrobotics.com/>
* Cliquez sur **Resources** dans la barre de menu à droite de l’écran et ensuite sur **Software** dans la liste des resources qui s’affichent
* Connectez vous ou créer un compte si vous n’en avez pas
* Dans la liste des programmes, allez sur **choregraphe suite** et cliquez sur le setup de windows pour le télécharger 
* Ouvrir le fichier d’installation que vous avez télécharger et suivre la procédure d’installation qui est toute simple(de préférence, accepter les choix par défaut)

Pour les autres systèmes d’exploitation, rendez vous à l’adresse suivante <http://doc.aldebaran.com/1-14/software/installing.html> pour voir la procédure d’installation.

##### Apprendre à utiliser Choregraphe

Choregraphe est un outil très simple d’utilisation. Il repose sur un système de boîtes regroupées dans des librairies qui, utilisées seules ou en série, permettent de simuler de nombreuses actions du robot.

Voici ce à quoi ressemble l’interface de Choregraphe une fois que vous l’avez ouverte.



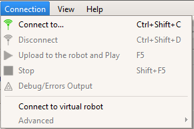
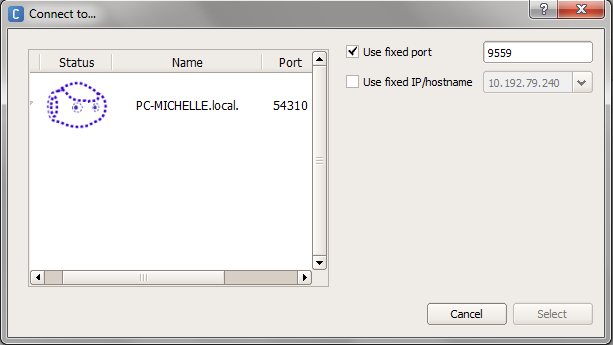
D andrea@cotza.me

E andrea@cotza.me

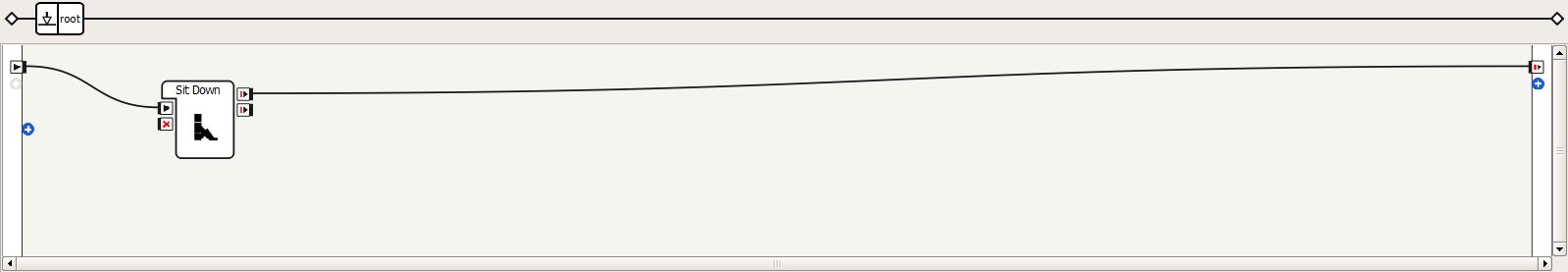
Candrea@cotza.me

B andrea@cotza.me

A andrea@cotza.me

* **A :** Cette zone est constituée éessentiellement de boutons.Parmis elles, les plus importantes sont:
* Le bouton  : Situé dans la barre de menu, il permet de choisir à quel type de robot se connecter. En effet, Choregraphe permet de se connecter à un robot réel et lui faire exécuter des actions mais au cas où vous n’aurez pas de robot de le simuler en vous connectant à un robot virtuel incorporé dans ce logociel. Alors, en cliquant sur ce bouton de connection, le menu déroulant suivant s’affiche . En cliquant sur **Connect to ,** cette fenêtre  s’affiche et vous pouvez selectionner dans la liste des robots réels connectés, celui à qui vous voulez vous connecter ou entrer son adresse IP dans le champ Use fixe IP/hostname.Le bouton **Connect to virtual robot** permet quant à lui de se connecter au robot virtuel
*  :Il s’agit des raccourcis pour la connection au robot réel. Le bouton vert permet de se connecter et rouge de se déconnecter.
*  : Ce bouton est un raccourci qui permet de charger un « behavior » dans le robot et de le lancer. Nous verrons plus bas la notion de « behavior ». Pour l’arrêter, cliquer juste sur le bouton Stop  à côté. S’il y a des erreurs lors du chargement d’un « behavior », le bouton suivant  devient rouge et en cliquant dessus, on peut voir les erreurs survenus.

Le bouton  permet de voir l’état de chargement d’un « behavior » dans le robot.

*  : Il permet de selectionner les fenêtre à afficher dans l’interface.Par exemple, si vous voulez afficher la fenêtre des logs, il vous suffit de cliquer sur ce bouton puis selectionner **Log viewer.**
* **B :** Cette zone est reservée à l’affichage de « behavior » et des fichiers qui les constituent. Un « behavior » est un groupe d’instructions que vous chargez dans le robot et qui lui font faire les actions souhaitées.En base de données, un « behavior » correspondrait à un fichier sql qui contient les instructions de création et suppression de la base de données, les triggers etc. Un « behavior » peut être associé à un projet.
* **C :** Cette fenêtre sert à visualiser le robot. Si vous êtes connecté à un robot virtuel, alors l’image du robot qui sera affichée sera celle du robot fourni pour la simulation ; sinon ça sera celle de votre robot auquel vous êtes connecté. Les boutons suivants  situés au dessus de l’image du robot permettent d’activer ou non des options sur la fenêtre. Exemple : le bouton  permet d’afficher ou non sur la fenêtre ceux que voit le robot.
* **D :** Cette fenêtre sert de plateforme où sont assemblées les boîtes pour créer des «  behaviors »(Suite d’instructions). Pour créer un « behavior » il suffit de :
* Selectionner une boîte dans la liste des librairie
* La glisser dans cette fenêtre
* Connecter l’entrer « onStart »du root à celle de la boîte puis la sortie « onStopped » de la boîte à celle du root comme le montre l’image suivante : 

Pour charger le « behavior » dans le robot et l’exécuter, il suffit de cliquer sur le bouton .

* **E :** Dans cette zone, on retrouve les librairies et les boîtes.Une librairie est un ensemble de boîtes. Ces boîtes sont être reparties dans des dossiers selon leur action.Exemple : la librairie standrad contient dans le dossier Sound les boîtes :Play sound, record sound, set speaker volume, sound located, sound peak. Un dossier peut contenir un sous-dossier. Exemple : Le dossier Sound de la librairie standard contient un sous-dossier Audio. Chaque librairie a un onglet comme on peut le voir sur cette image  .
* Le premier onglet est celui de la librairie **standard**, qui contient toutes les boîtes de base telles que , etc.
* Le deuxième est celui de la librairie **advanced** qui contient les boîtes aux fonctionnalités encore plus avancées telles que  qui permet retourne une string avec le nom de la posture prise par le robot, ou encore la boîte  qui permet au robot de marcher en suivant une trajectoire décrite dans un fichier joint. Toutefois, avec les boîtes de base de la librairie standard, il est possible de reproduire le comportement d’une boîte de cette librairie.Exemple : en associant une série de boîte de base , le robot peut suivre une trajectoire donnée puisque cette boîte permet au robot de se déplacer en suivant une direction passée en paramètre.
* Le troisième est celui de la librairie **tablet** qui quant à elle contient les boîtes en rapport avec la tablette montée sur le torse du robot. Exemple : La boîte  permet d’afficher une image sur l’écran de la tablette.
* Le dernier, **Search** permet de rechercher une boîte parmi celles existantes dans toutes les librairies en entrant le nom de la boîte recherchée dans le champ prévu à cet effet.

Cette fenêtre peut contenir d’autres librairies. En effet, en cliquant sur le bouton  « New box library », vous créez une nouvelle librairie qui s’ajoute à la liste de celles existante. Pour le moment elle est vide. Vous pouvez la peupler avec les nouvelles boîtes que vous aurez créés.

Nous avons testé les boîtes de la librairie standard avec le simulateur puis le robot afin de comprendre leur fonctionnement. En effet, il n’est pas possible d’effectuer tous les tests avec le robot virtuel du simulateur.Par exemple, le robot virtuel ne peut s’asseoir, parler(il affiche juste le texte qu’il est censé dire).

1. Tests Audio

|  |  |  |
| --- | --- | --- |
| Nom de la boîte | Fonction | Procedure |
| Play sound | Amener Pepper à jouer une musique. | A l’aide de ce bouton ajouter un fichier wave dans le behavior puis selectionner le box .Editer ses paramètre et dans le nom du fichier, choisir le fichier wave du behavior. Jouer |
| Record sound | Enregistrer un son exterieur dans le robot. A tester avec le robot | ok |
| Set speaker volume | Ajuster le volume.A tester avec le robot | ok |
| Sound loc | Permet de location dans quelle direction un sound est produit.A tester avec le robot |  |

1. Tests de la voix(voice test)

|  |  |  |
| --- | --- | --- |
| Animated say | Pepper dit quelque chose accompagné d’un geste.Exemple : dire bonjour acompagné du geste de la main. | Pour ce faire, choisir le box  et le jouer..le geste avec les mains qui accompagne le bonjour est différent selon qu’il soit débout ou assis. |
| choice | Utilisé pour un test question-réponse, il permet de sauvegarder dans le robot une liste de réponses pour des questions données | Ajouter dans la boîte une série de questions avec leurs réponses. |
| dialog | Contient des exemples de dialogues en plusieurs langues. | choisir  dans la librairie .Uniquement gestuel |
| Say | Pepper peut dire « Bonjour ». | Choisir le box  dans la library et le connecter |
| Set language | Permet de choisir la langue dans laquelle pepper va écouter et parler. | Choisir  dans la librairie et l’écouter |
| Speech reco | Permet de reconnaître un mot dans la liste des mots insérés dans les paramètres du box.Peut être utilisé pour le dialigue. | Choisir  dans la librairie et jouer |

1. Tests du comportement(behavior)

|  |  |  |
| --- | --- | --- |
| Nom de la boîte | Fonction | Procédure |
| Run behavior | Permet à l’aide d’un texte édité dans ses paramètres de faire faire quelque chose à pepper.Exemple, hocher la tête. | Le box correspondant est : . Les comportements inconnus ne seront pas éxécutés. |

1. Tests de la communication (communication)

|  |  |  |
| --- | --- | --- |
| Fetch email | Permet de rechercher un email depuis le server. Peut être utilisé pour recevoir les confirmations lors de l’achat des bien par exemple. | Pour cela, il faudrait créer une adresse mail au robot. Le box correspondnat est : |
| Send email | Permet d’envoyer un email depuis le robot. | Il faut pour ce la que le robot ait une adresse email.Le box correspondant est : |
| Remote control | Permet de faire les commandes à distance.Pepper commande un ordianteur ou alors, commandé pepper depuis un autre appareil. | Les box à utiliser sont : |

1. Tests des animations

|  |  |  |
| --- | --- | --- |
| blink | Cette box permet au robot de clignoter une fois. | utiliser .Il cligne des yeux une fois |
| Random eyes | Permet de changer les yeux du robot aléatoirement | Utiliser .Ses yeux prennent des couleurs aléatoires annoncées par le bouton de son torse. |
| Twinkle | Permet de faire briller des leds par intermitance pendant une durée de temps définies dans les paramètres. | Utiliser |
| Ear leds | Permet d’augmenter l’intensité lumineuse des leds. | Utiliser |
| Eye leds | Permet de definir la couleur des leds des yeux | . Utiliser |
| Set leds | Definir l’intensité lumineuse d’un group de leds. | Utiliser |
| Set single led | Definir l’intensité lumineuse d’un led en particulier. | Uitliser |

1. Tests de la gestuelle(motion)

|  |  |  |
| --- | --- | --- |
| Breath | Permet au robot d’enchainer des mouvements avec son corps. | utiliser .Il bouge horizontalement le corps et fait un mouvement semi-circulaire avec les mains. |
| Hello | Ce box contient une animation de Hello. | Utiliser .Il salue de la main sans dire mot |
| Wipe forehead | Permet au robot de s’essuyer le front. | utiliser |
| hands | Permet au robot d’ouvrir sa/ses mains. | Utiliser .Il déplie ses doigts |
| Sit down | Asseoir le robot. | Utiliser |
| Stand up | Permet au robot de se mettre debout.on peut éditer le nombre de fois qu’il peut essayer de se mettre débout. | Utiliser |
| Move to | Permet au robot de se déplacer vers un point donné relativement à sa position courante. | Utiliser |
| Move toward | Permet au robot de se diriger vers une direction donnée. | Utiliser |
| Obstacle avoidance | Permet de se déplacer et d’éviter les obstacles. | Utiliser . Mais quand on place l’object subitement devant lui, il marche dessus et puis tombe. |

1. Tests de la vue

|  |  |  |
| --- | --- | --- |
| Face detection | Permet de détecter les visages et retourne le nombre de visages détectés. | Utiliser .affiche le nombre de visages trouvé mais ne le dit pas |
| Face Reco | Permet de détecter et reconnaitre des visages. | Pour cela, il devra reconnaitre des visages avec le box de reconnaissance faciale prévu. |
| Is in darkness | Permet au robot de vérifier si il est dans le noir. | Utiliser |
| Learn Face | Permet au robot de mémoriser des visages.Ses yeux deviendront vert en cas de succès et rouge sinon. | ok |
| Record vidéo | Permet d’enregistrer des vidéos à l’aide d’une des cameras du robot.Le temps d’enrégistrement est édité dans les paramètres.. | Uitliser |
| Take picture | Permet de prendre une photo à l’aide d’une des cameras du robot. | Utiliser |
| Unlearn faces | Permet de supprimer tous les visages de la db. Uitliser |  |
| Vision reco | Permet de reconnaitre les objects, images et endroits |  |
| Look At | Permet au robot de regarder vers une position donnée. | ok |
| Point at | Permet au robot d’indiquer une position | ok |

Annexe 3 : Fonctionnement de AlTextToSpeech

ALTextToSpeech est un module qui permet à Pepper de parler. Le procédé est le suivant : Après avoir spécifier le texte à dire et la langue d’expression à travers ses méthodes « say » et « SetLanguage », ALTextToSpeech envoie une commande à un logiciel de traitement. Selon la langue choisie, le logiciel est différents. Si la langue est le Japonais, le logiciel de traitement est « **microAITalk » conçu par la société AI, Inc ; sinon c’est un logiciel des sociétés « ACAPELA » OU « Nuance ». Ces logiciels sont des programmes qui permettent de traduire des textes en voix. Une fois les voix fournies par les traducteurs de texte, il est possible de les paramétrer à l’aide des tags(changer la vitesse, mettre des pauses entre les mots, etc.)**

**Annexe 4 :** ALAnimatedSpeech

Annexe 5 : ALFaceDetection

Pour éffectuer la detection de personnes, Pepper utilise les services dévéloppés par la société OKI . Il s’agit d’une société d’électonique qui fournit des services pour la détection et la reconnaissance faciale. Ces services consistent à déterminer le visage d’une personne dans une image et à retourner les informations sur celle-ci. Lorsque le module AlFaceDetection est appelé, le système souscrit en même temps à l’événement « FaceDetected ». Une fois le traitement d’image terminé, les résultats sont stockés dans la mémoire de Pepper avec la clé « Facedetected ». Il faut donc aller en mémoire et récupérer la valeur correspondant à cette clé. La valeur retournée par la clé « FaceDetected » est un tableau organisé ainsi qui suit :

* Premier élément du tableau : TimeStamp. Il s’agit d’un tableau à deux éléments indiquant tous deux le temps utilisé pour éffectuer la détection. La première valeur est le temps en séconde et la deuxième, en minisécondes utilisé.
* Deuxième élément du tableau : [FaceInfo[N],Time\_Filtered\_Reco\_Info] . Il s’agit d’un tableau à deux éléments. Le premier est le tableau contenant les informations sur tous les visages détectés. Le deuxième est une variable de type tableau qui permet de faire la reconnaissance Faciale. En effet, selon la taille de ce tableau et celle de ces éléments, il est possible de déterminer si une ou plusieurs personnes ont été reconnu par Pepper ou pas. Le thème de la reconnaissance faciale sera traité dans un autre scénario plus bas.
* Les éléments suivants : CameraPose\_InTorseFrame, CameraPose\_InRobotFrame, CameraPose\_Id donnent les informations sur les cameras utilisés pour prendre les images nécessaires à la détection. Les deux premiers indiquent la position 6D de la camera utilisée et la dernière, l’id de la camera utilisée.

Pour déterminer si Pepper a détecté quelqu’un, seule une comparaison à zéro de la taille du tableau FaceInfo[N](voir le deuxième élément de la liste ci-dessus) est nécessaire.

Annexe 8 : Code de l’application 1 : Pepper donne le nombre de personnes en face

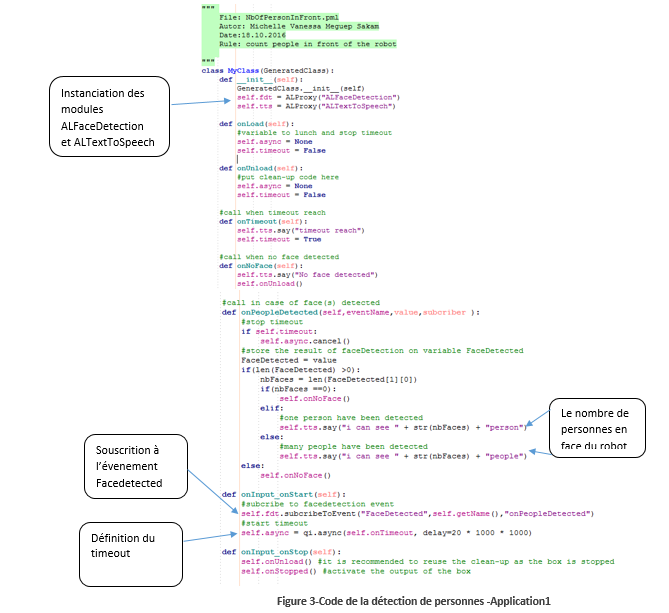
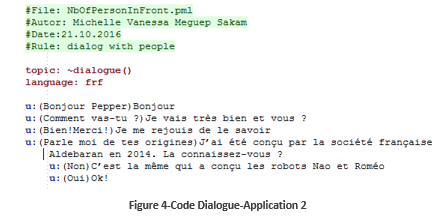


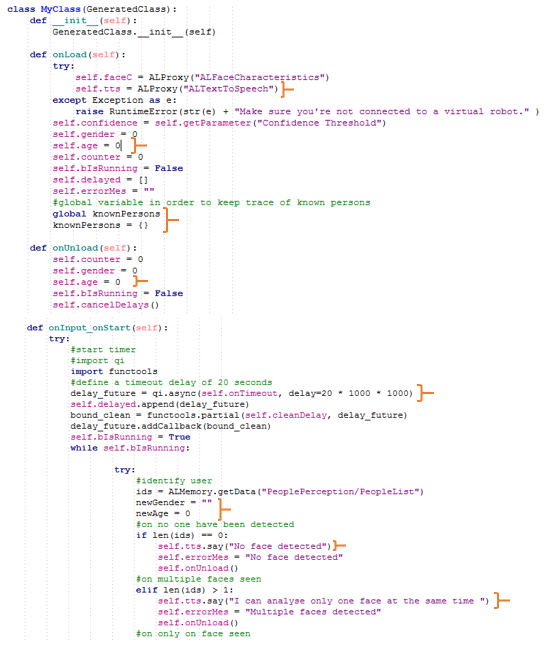
Figure 3 : Pepper donne le nombre de personnes en face

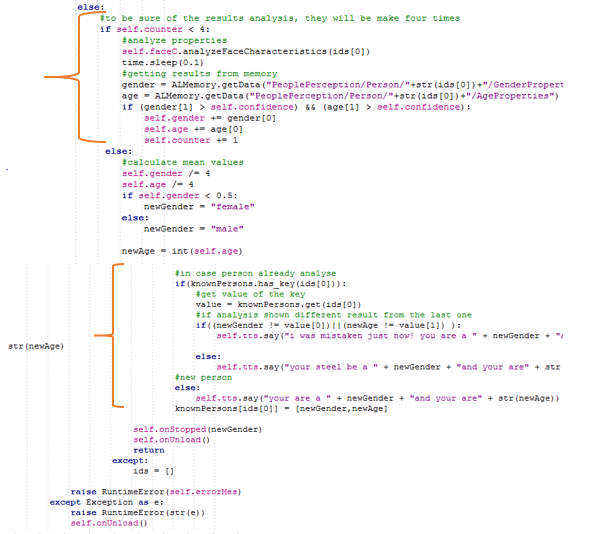
Annexe 8 : Code de l’application 2 : Pepper dialogue

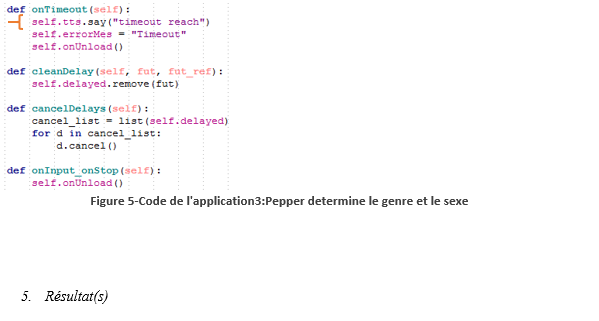


Annexe 8 : Code de l’application 3 : Pepper détermine le genre et le sexe de ses interlocuteurs

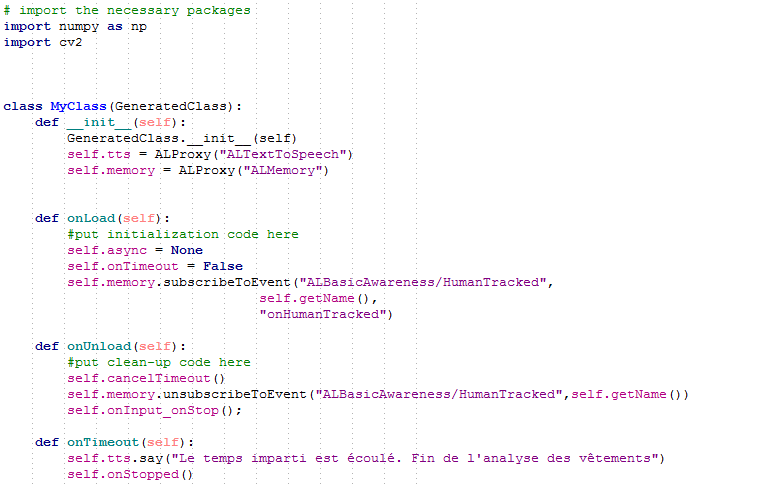
Pour cette application, il a été utilisé une boîte existante de Choregraphe et le code y a été ajouté afin de correspondre aux éléments du scénario. La boîte de base utilisée est *GetGender* qui permet de donner le genre d’une personne. Le texte ajouté est marqué par les accolades et ci-dessous.

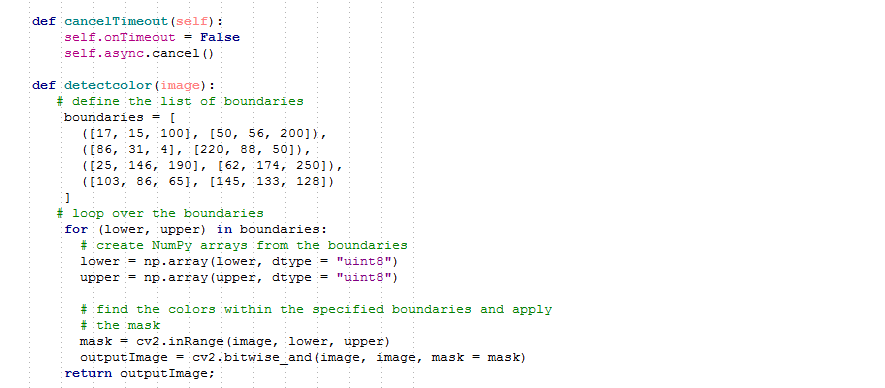


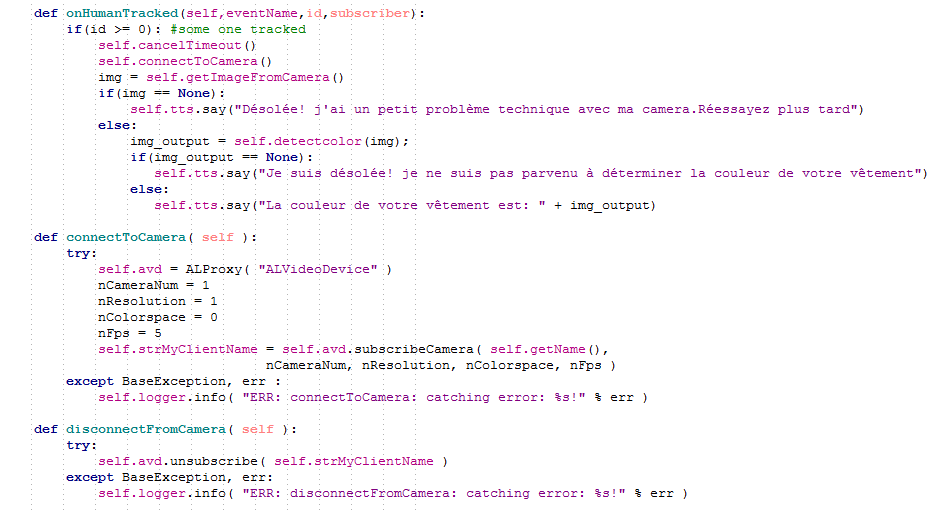


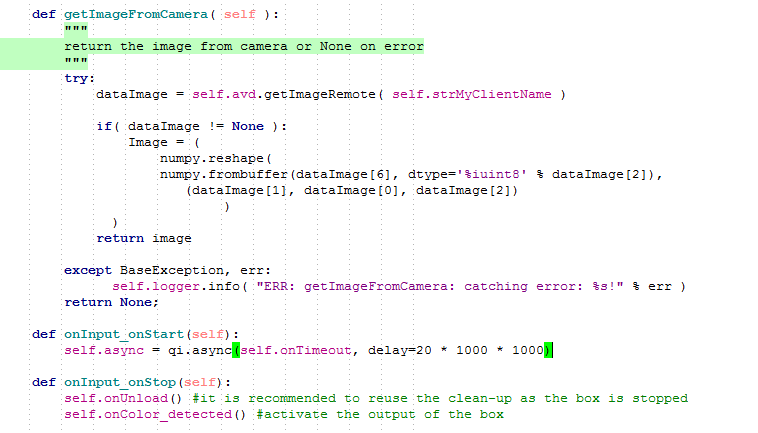


Annexe 8 : Code de l’application 4 : Pepper détermine la couleur du vêtement avec openCV









**Annexe 8 : Code de l’application 5 : Pepper apprend et reconnaît des visages**



