Département des Technologies de l’information

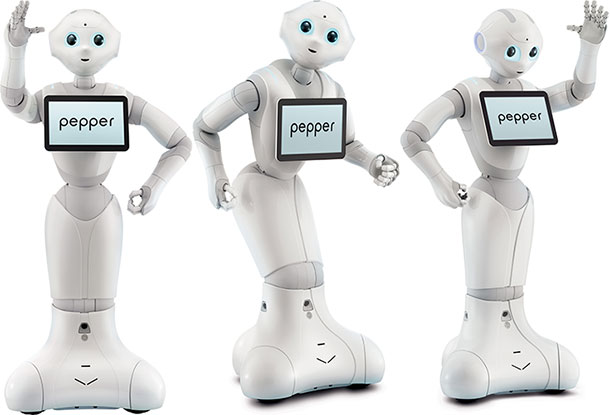
et de la communication

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 : Alexandre Grillon

# Remerciements

Je tiens de tout cœur à remercier les personnes ayant influencées 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 passionnant, 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 Mr 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](#_Toc478025309)

[Table des figures 5](#_Toc478025310)

[Liste des tableaux 6](#_Toc478025311)

[Table des abréviations 7](#_Toc478025312)

[Table des mots clés 8](#_Toc478025313)

[Résumé 9](#_Toc478025314)

[Chapitre 1 : Introduction 10](#_Toc478025315)

[1.1. Introduction 10](#_Toc478025316)

[1.2. Contexte 10](#_Toc478025317)

[1.3. Objectifs 10](#_Toc478025318)

[1.4. Cahier des charges 11](#_Toc478025319)

[Chapitre 2 : Recherche et documentation 12](#_Toc478025320)

[2.1. Présentation du robot Pepper 12](#_Toc478025323)

[2.2. Caractéristiques techniques 12](#_Toc478025324)

[2.3. Fonctionnalités offertes par Pepper 13](#_Toc478025327)

[2.3.1 Répertoire des fonctionnalités de Pepper 13](#_Toc478025328)

[2.3.2 Certaines fonctionnalités de Pepper vues plus en détail 15](#_Toc478025329)

[2.4. Cas pratiques 22](#_Toc478025330)

[2.4.1 Connecter Pepper à internet 22](#_Toc478025331)

[2.4.2 Changer la langue de dialogue de Pepper 24](#_Toc478025332)

[2.4.3 Rendre Pepper autonome 24](#_Toc478025333)

[Chapitre 3 : Recherche et développement 25](#_Toc478025334)

[3.1 Objectif 25](#_Toc478025335)

[3.2 Technologies utilisées 25](#_Toc478025336)

[3.3 Plan 25](#_Toc478025337)

[3.4 Applications 25](#_Toc478025338)

[Application 1 : Pepper donne le nombre de personnes en face de lui 25](#_Toc478025339)

[Application 2 : Pepper dialogue avec ses interlocuteurs 27](#_Toc478025340)

[Application 3 : Pepper détermine l’âge, le genre(homme/femme) d’une personne 28](#_Toc478025341)

[Application 4 : Pepper détermine la couleur de vêtements avec Open CV 30](#_Toc478025342)

[Application 5 : Pepper apprend et reconnaît des visages 32](#_Toc478025343)

[Chapitre 4 : Projet 34](#_Toc478025344)

[4.1 Description 34](#_Toc478025345)

[4.2 Phase d’analyse 34](#_Toc478025346)

[4.3 Phase de conception 35](#_Toc478025347)

[4.4 Phase de développement 39](#_Toc478025348)

[4.5 Phase de tests 41](#_Toc478025349)

[Chapitre 5 : Conclusion 43](#_Toc478025350)

[5.1 Les résultats obtenus 43](#_Toc478025351)

[ Phase 1 : Recherche et documentation 43](#_Toc478025352)

[ Phase 2 : Recherche et développement 43](#_Toc478025353)

[ Phase 3 : Projet 43](#_Toc478025354)

[5.2 Les difficultés rencontrées 44](#_Toc478025355)

[5.3 Les expériences acquises 44](#_Toc478025356)

[Annexe 1 : Chorégraphe 45](#_Toc478025357)

[Annexe 2 : Les frameworks Naoqi et Qi 52](#_Toc478025358)

[Annexe 3 : Prise et traitement d’image par Pepper 52](#_Toc478025359)

[Annexe 4 : Module ALFaceDetection 52](#_Toc478025360)

[Annexe 5: Module ALPeoplePerception 53](#_Toc478025361)

[Annexe 6 : Module ALTextToSpeech 53](#_Toc478025362)

[Annexe 7: Module ALAnimatedSpeech 53](#_Toc478025363)

[Annexe 8: Module ALMotion 53](#_Toc478025364)

[Code-Annexe 54](#_Toc478025365)

[Code de l’application 1 : Pepper donne le nombre de personnes en face 54](#_Toc478025366)

[Code de l’application 2 : Pepper dialogue 55](#_Toc478025367)

[Code de l’application 3 : Pepper détermine le genre et le sexe de ses interlocuteurs 56](#_Toc478025368)

[Code de l’application 4 : Pepper détermine la couleur du vêtement avec openCV 58](#_Toc478025369)

[Code de l’application 5 : Pepper apprend et reconnaît des visages 60](#_Toc478025370)

# Table des figures

Figure 1-Pepper avec les leds allumés 16

Figure 2-angle de vision du capteur 3D de Pepper 17

Figure 3-angle de vision des caméras 2D de Pepper 17

Figure 4-Articulations principales de Pepper 18

Figure 5-représentation spatiale pour les mouvements des articulations 19

Figure 6-rotation d'un avion autour des axes yaw, pitch et roll 19

Figure 7-angles min et max de rotation de l'articulation du cou autour des axes y et z 20

Figure 8-Base du robot Pepper 20

Figure 9-zone de sécurité de Pepper pour éviter les obstacles 21

Figure 10-zones sans système de détection d'obstacle 21

Figure 11-port Ethernet du robot Pepper (source2, pp. http://doc.aldebaran.com/2-4/family/robots/connectivity\_nao.html) 22

Figure 12-port USB du robot Pepper (source2, pp. http://doc.aldebaran.com/2-4/family/robots/connectivity\_nao.html) 23

Figure 13-page internet du robot Pepper 23

# Liste des tableaux

Tableau 1-diagramme des cas d'utilisation - Application 1 26

Tableau 2-diagramme des cas d'utilisation - Application 3 28

Tableau 3-diagramme des cas d'utilisation - Application 4 30

Tableau 4-diagramme des cas d'utilisation - Application 5 32

Tableau 5-diagramme des cas d'utilisation-Projet 35

Tableau 6-diagramme de collaboration-programme principal du projet 36

Tableau 7-diagramme d'activité du contrôleur-projet 37

Tableau 8-diagramme de séquence du programme "H-M Interaction « -Projet 38

# Liste des abréviations

HEIG : Haute École d’ingénierie et de gestion du canton de Vaud

# Résumé

Dans le cadre du travail de fin d’études de Bachelor[[1]](#footnote-1) à la Haute École d’ingénierie et de gestion du canton de Vaud(HEIG-VD[[2]](#footnote-2)), un projet a été mené sur le robot Pepper de l’entreprise Softbank Robotics[[3]](#footnote-3). Il s’agit d’ un travail de recherche et de développement sur le robot Pepper. Ainsi, ce projet a été réaisé en trois phases distinctes mais complémentaires : Une première phase où des recherches ont été menés afin de comprendre le fonctionnement du robot et de produire une documentation sur ses fonctionnalités, une deuxième phase où quelques applications ont été dévéloppées afin de comprendre comment programmer le robot et y produire une documentation et en fin, une dernière phase plus pratique où un projet à été mené pour les portes ouvertes de la HEIG-VD [[4]](#footnote-4)2017 avec pour but de programmer Pepper pour accueillir et distraire les personnes présentes. Ces trois phases sont expliquées et détaillées dans le corps de ce rapport.

# Préliminaire

## Introduction

Depuis la création des premiers robots au début du 20ème siècle, le domaine de la robotique [[5]](#footnote-5)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 [[6]](#footnote-6)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[[7]](#footnote-7). 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, c’est le dernier robot conçu par l’entreprise française Aldebaran 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 le robot Nao a été utilisé pour des projets tels que l’exposition à « la maison d’ailleurs » d’Yverdon-les-bains[[8]](#footnote-8), aux portes ouvertes de la HEIG-VD de 2015. En 2016, lorsque a commencé la commercialisation de Pepper en Europe, la HEIG-VD s’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 « Institut for information and Communication Technologies »(IICT) [[9]](#footnote-9)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’ institut IICT 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 projet sont :

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

## Cahier des charges

Le but principal 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.

# Recherche et documentation



Le but de cette section est de présenter les résultats de recherches qui ont été menés sur Pepper. Il a été question de se documenter sur le robot tant sur la partie matériel que logiciel et aussi sur tous les éléments qui composent son environnement

## Présentation du robot Pepper

Pepper est un robot à l’apparence humaine, développé en 2012. 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 il 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 de l’entreprise 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 de renseigner la clientèle.
* À la une 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 en mai 2016, au « Disrupt SF » à SanFrancisco en septembre 2016 ; salon organisé par Tech Crunch, le média américain de référence sur les nouvelles technologies. 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 qui la conçoit, développe elle-même plusieurs applications disponibles sur son « App Store ». Pepper est vendu actuellement au prix très abordable de 1500euros soient près de 1600 francs suisses mais il est également possible de souscrire à un abonnement et payer mensuellement. Déjà près de 10000 exemplaires de Pepper 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 d’une tablette ordinaire ; navigation sur internet, accès à des contenus multimédias, consultation et envoie des courriers électroniques, installation d’applications etc.
* Audition : Pepper possède 4 microphones situés sur le haut de la tête, 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 qui lui font office d’oreilles
* Vision : Pepper dispose d’une caméra 3D pour la profondeur située dans les deux yeux(l’émetteur dans un œil et le recepteur dans l’autre) et de deux autres HD identiques(l’un situé sur le front et l’autre sur le menton) utilisé pour la couleur. Ses cameras lui permettent de 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.
* Dégré de liberté : Pepper a un degré de liberté égal à 17 ; ce qui lui donne la possibilité de faire plusieurs mouvements et des plus complexes.
* 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, il possède un centrale d’inertie qui lui permet de garder un bon équilibre.
* 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 sur la tête et 2 dans les mains) qui lui permettent d’être sensible au toucher, de 2 sonars et 6 lasers placés sur sa base et un capteur gyroscopique pour éviter les obstacles, un capteur incorporé dans sa batterie qui permet d’avoir les informations sur son état de charge et sa température.
* Connexion 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 de 12h.
* Système d’exploitation : Le système d’exploitation installé sur Pepper est Naoqi conçu également par la société Aldebaran.

## Fonctionnalités offertes par Pepper

Il s’agit dans cette section, de présenter dans un premier temps les principales fonctionnalités du robot et ensuite les exposer de manière plus détaillée.

### Répertoire des fonctionnalités de Pepper

Pepper et le robot Nao partage de nombreuses fonctionnalités en commun. Comparer ces deux robots est donc un bon moyen pour resortir les spécificité du robot Pepper et comprendre l’intérêt qu’il succite.

Nao et Pepper possèdent tous deux les capacités suivantes :

* 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 choisie après l’anglais est gratuite et le choix définitif et les autres langues installées par la suite sont payantes. En outre, ils communiquent aussi avec des gestes et des leds placés sur la partie supérieure de leur corps.

* Bouger les membres ou se déplacer

Ils peuvent bouger 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.

* Danser

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

* Détecter et reconnaître un visage, une voix, des objets

A l’aide de leurs cameras haute définition 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 présence d’une personne, d’objets ou d’images, de les mémoriser et ensuite les reconnaître lorsqu’ils les perçoivent ou voient pour une seconde fois.

* Se connecter à internet

Chaque robot de Softbank Robotics possède une page internet accessible 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.

* 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 développement ses propres applications et de les installer dans les robots. En effet, les robots Pepper et Nao sont en fait des interfaces programmables. Ils sont open source et supportent de nombreux langages et plateformes de programmation.

Mais ce qui distingue Pepper de Nao est :

* Les émotions

Pepper est capable d’analyser le visage ainsi que la voix de ses interlocuteurs pour déceler les cinq émotions suivantes : la joie, la tristesse, l’étonnement, la neuter, la colère; il peut ainsi modifier son comportement en conséquence et miner en retour des émotions.

* La communication

Pepper dispose en plus sur son torse d’une tablette tactile lui permettant d’interagir avec l’homme. Il peut ainsi y faire tourner des contenus multimédias, afficher des menus, faire tourner des applications sur sa tablette etc.

* La vision

Pepper possède contrairement à Nao une caméra 3D qui lui permet d’avoir un bon champ de vision sur son entourage.

* 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.

* 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 où se déplace. C’est une caractéristique qui a été amélioré chez Pepper. Il est plus souple et ses actions moins mécaniques

* L’équilibre

Pepper tient très bien en équilibre contrairement à Nao qui peut tomber en essayant certains déplacements ou en restant trop longtemps débout. Lorsque Pepper est fatigué, il se repli sur son torse, baisse la tête et stop le programme en cours. Les leds et sa tablette reste toutefois allumés indiquant qu’il n’est pas éteint.

* 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.

* Autonomie

Pepper a une batterie d’une autonomie de 12h contrairement à NAO dont la batterie ne peut faire que 90 min hors charge.

* Fonctionnalités logicielle

Dans les APis du Framework Naoqi utilisés par les robots Pepper et Nao des modules supplémentaires spécifiques ont été ajoutés uniquement pour le robot Pepper. Ceci offre plus de possibilités dans la programmation du robot Pepper

### Certaines fonctionnalités de Pepper vues plus en détail

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

* Communication

1. Communication corporelle
2. 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. Dépendant 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 accueillir. 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 Leeds

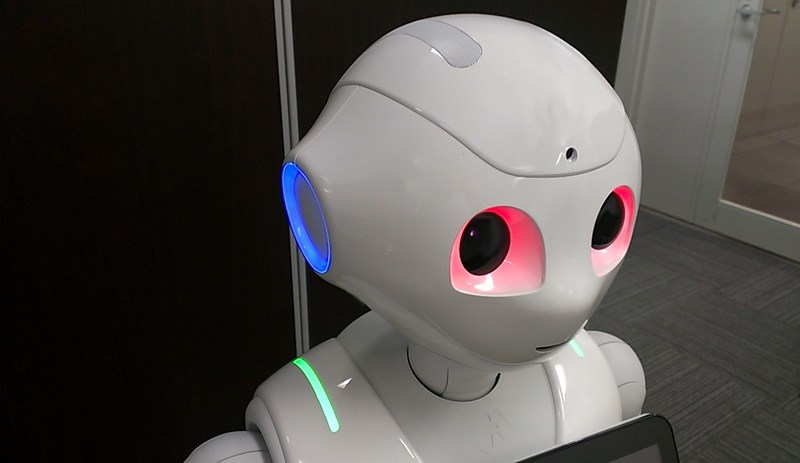


Figure 1-Pepper avec les leds allumés

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 a réussi et rouge sinon
  + Lors d’un dialogue avec Pepper, lorsque les leds de ses yeux deviennent bleues et que la lumière de ses leds 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.

1. Communication vocale

En plus de l’interaction avec l’homme à travers « la parole », Pepper émet aussi des sons dans le but d’informer sur son état. Par exemple, 1) Lorsqu’on l’allume, pour indiquer la fin du processus de démarrage, Pepper émet le son « Ognak Gnouk » ; dérivé du japonais qui signifierait « jamais je ne t’abandonnerai ». De même, lorsqu’on presse sur son bouton d’arrêt, il émet le son « Gnouk » qui signifierait également « Je ne vais jamais te laisser tomber ». 2) Lors d’un dialogue, lorsque Pepper ne perçoit pas bien ce qui est dit, ou bien que le mot ou la phrase ne lui est pas familier, il émet un son « bilips » deux fois ; le premier, pour indiquer qu’il n’écoute plus et le second, qu’il recommence à écouter.

* Vision

Pepper possède pour trois cameras dont une 3D et les deux autres 2D.

1. Propriété des éléments de vision
2. La camera 3D : La camera 3D de Pepper est situé dans ses deux yeux. Il s’agit d’un capteur de profondeur de modèle ASUS XTION et de type SOC Image Sensor. Elle produit une image de 1280x1024 pixels. Celle-ci lui permet de voir à une distance située entre 40 cm et 8 m avec un champ de vision de 70° pour 45 ° verticalement et 58° à l’horizontale.

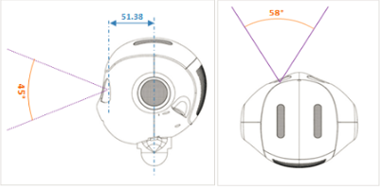


Figure 2-angle de vision du capteur 3D de Pepper

Pour avoir plus d’informations sur ses caractéristiques, se rendre à l’adresse suivante : <http://doc.aldebaran.com/2-4/family/pepper_technical/video_3D_pep.html>.

1. Les cameras 2D : Pepper possède deux cameras identiques hautes définitions, une située dans sa bouche et l’autre sur son front qu’il utilise pour. Ces caméras sont du modèle OV5640 et de type SOC Image Sensor. Elles sont capables d’avoir une résolution allant jusqu’à 2560x1080 pixels pour 5 trames par secondes. L’image qu’elles produisent à quant à elle une résolution de 2592x1944. Leur angle de vision est de 68,2° pour 44.3° verticalement et 57.2° horizontalement. Elle confère au robot un champ de vision allant de 30 cm à l’infini.

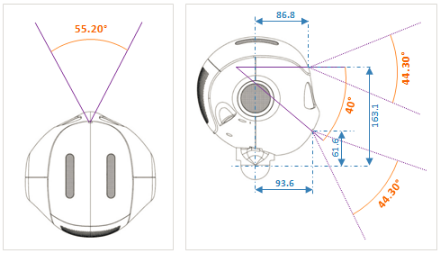


Figure 3-angle de vision des caméras 2D de Pepper

Pour en savoir plus sur les paramètres de ses caméras, les caractéristiques des images produites, les couleurs, frames ou résolutions supportées, suivre ce lien : [caractéristiques des caméras 2D de Pepper](http://doc.aldebaran.com/2-4/family/pepper_technical/video_pep.html)

Pepper possède aussi une camera 3D située sur son front. Elle a une résolution allant jusqu’à 320x240 pixels pour 20 trames par seconde.

Pepper utilise ses cameras 3D et 2D pour l’identification des mouvements et l’analyse des caractéristiques faciales.

1. Influence sur la vision

Grâce aux propriétés de ses caméras, Pepper a une bonne qualité de vision. Mais celle-ci peut être atténuée lorsque l’environnement trop éclairé ou moins éclairé.

1. Conditions idéales de vison

Pour que Pepper puisse avoir une bonne vision sur son entourage, il faudrait :

* Un éclairage compris entre 100 et 500 lux
* Se tenir à une distance de Pepper comprise entre 1 et 2 m. Lorsqu’une personne se rapproche trop près de Pepper (moins de 50 cm), il adapte sa distance de vision en penchant en arrière.
* Mouvements

Pepper possède plusieurs articulations qui lui permettent d’effectuer toute une variété de mouvements et de prendre des postures différentes. Les plus importantes sont répertoriées sur la figure ci-dessous :

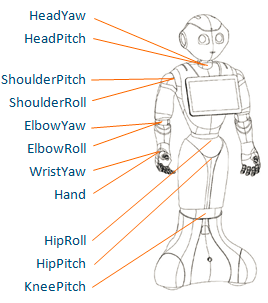


Figure 4-Articulations principales de Pepper

Les mouvements de ces articulations sont définis par des axes x, y et z et des angles et axes de rotations qu’elles prennent dans l’espace.

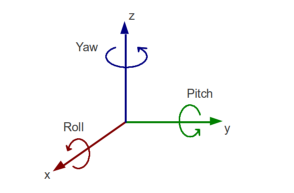


Figure 5-représentation spatiale pour les mouvements des articulations

Sur la figure ci-dessus, on distingue deux représentations :

* Représentation cartésienne : C’est la représentation traditionnelle, celle qui définit un repère cartésien avec ses trois axes x, y, z. Elle permet de représenter tout objet ou point dans l’espace à l’aide des coordonnées x, y et z.
* Représentation autour des axes yaw, pitch et roll : Ces axes font références aux angles d’Euler utilisé pour définir l’orientation d’un objet dans l’espace. Ils sont par exemple utilisés pour définir et décrire la trajectoire d’un avion, les mouvements d’une caméra dans un environnement 3D

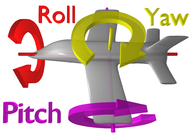


Figure 6-rotation d'un avion autour des axes yaw, pitch et roll

Se sont donc ces six axes qui sont utilisées pour définir les mouvements du robot autour de ses articulations.

La séquence de rotation adoptée pour l’exécution des mouvements est la suivante : roll ->pitch🡪yaw. Donc d’abord une rotation autour de l’axe x, puis une rotation autour de l’axe y et en fin autour de l’axe z. Des valeurs min et max ont été définies pour les rotations autour des différents axes. La figure ci-dessous montre les valeurs définies pour les rotations autour des axes y et z de l’articulation du cou.

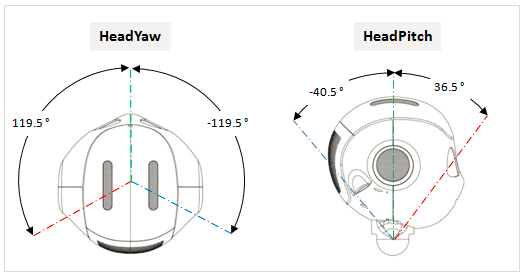


Figure 7-angles min et max de rotation de l'articulation du cou autour des axes y et z

Pour en savoir plus sur les articulations de Pepper, se rendre à l’adresse suivante : [Les articulations de Pepper](http://doc.aldebaran.com/2-4/family/pepper_technical/joints_pep.html). Ce sont ces articulations qui lui confère ses 20 degré de liberté ; c’est-à-dire, 20 différents mouvements indépendants qu’il peut effectuer. Ce degré de liberté est reparti ainsi : tête-2, épaules-2 de chaque côté, coudes-2 de chaque côté, poignet-1 de chaque côté, main-1 de chaque côté, torse-2, gênoux-1, base-3.

Sur les articulations, sont installés des moteurs pour permettre d’effectuer ses mouvements. Il en existe en tout 20, de 5 types différents. Le lien suivant donne sur une page où sont présentés les 5 types de moteurs et leurs caractéristiques, ainsi que le type auquel appartient chaque articulation. Lien : [les moteurs des articulations de Pepper](http://doc.aldebaran.com/2-4/family/pepper_technical/motors_pep.html)

* Déplacement

Pepper ne possède pas de jambes mais une base rigide posée sur trois roues omnidirectionnelles. Cette base de forme triangulaire, fait 42 cm de long.



Figure 8-Base du robot Pepper

Avec ces roues omnidirectionnelles, Pepper est capable de tourner sur lui-même sur 360°, de se déplacer à une vitesse maximale de 3km/h et avoir une accélération maximale de 1.7 m/s2. Pendant son déplacement, Pepper évite les obstacles à l’aide des capteurs anticollisions qu’il possède. Pour cela, il possède une zone dite « de sécurité » qui lui permet de s’arrêter lorsqu’un objet est détecté à l’intérieur de cette zone. Les limites de cette zone sont définies par deux distances : la distance tangentielle qui définit un rayon pour la zone de sécurité et une distance orthogonale qui définit la distance par rapport à un objet situé en face du robot

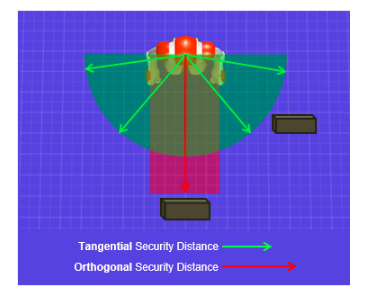


Figure 9-zone de sécurité de Pepper pour éviter les obstacles

(source1, pp. http://doc.aldebaran.com/2-4/naoqi/motion/reflexes-external-collision.html)

Ces distances tangentielle et orthogonale sont par défaut respectivement égales à 0.1 m et 0.4 m mais elles peuvent être modifiée par l’utilisateur du robot.

Il existe toutefois des zones dans l’entourage du robot où il ne peut détecter la présence d’obstacles. Ces zones dites « blind zones » sont situées entre les limites des champs de vision des lasers et des capteurs que Pepper utilise pour la détection d’obstacles. Le schéma suivant montre ces zones.

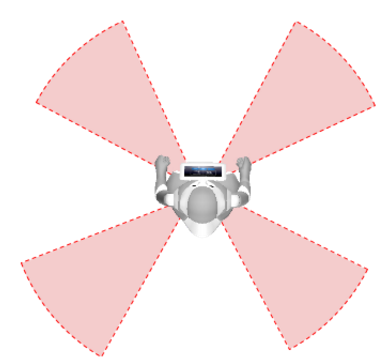


Figure 10-zones sans système de détection d'obstacle

(source1)

Pepper utilise pour détecter les obstacles son capteur 3D situé dans l’œil, ses deux capteurs sonars situés dans sa base.

## Programmation

Le robot Pepper est une plateforme de programmation avec la particularité que tout le monde peut le programmer. En effet, il a été mis pied par l’entreprise Softbank Robotics, un ensemble d’outils et moyens de programmation permettant aux experts comme au novices en robotique de programmer le robot Pepper.

### Système d’exploitation

L’entreprise Sofbank a dévéloppé pour tous ses robots un système d’exploitation unique appelé Naoqi. Ils’agit donc de l’OS intégré dans Pepper. C’est un système linux basé sur Gentoo. Toutes fois, il est compatible avec les systèmes Windows et Mac OS. Il peut être utilisé aussi bien sur une machine réelle que sur une machine virtuelle. Il permet la programmation du robot en deux modes : en mode production et en mode développement.

### Langages de programmation

Le robot Pepper peut être programmé en les différents langages de programmation suivants : python, c++, javascript, java.

### Technologies

Des technologies performantes ont été crées en même temps que les robots de la société afin de mieux répondre aux besoins des utilisateurs. Il s’agit de :

* Naoqi et Qi

Il s’agit de deux Frameworks permettant de programmer les robots de l’entreprise. Ils se composent de modules ayant chacun des fonctionnalités précises, qui offrent des méthodes, événements, signaux pour créer des comportements dans les robots. Exemple, le module ALTextToSpeech de Naoqi permet de manipuler la parole du robot. Sa méthode « say » permet de faire dire un texte au robot.

* Choregraphe, une plateforme à la fois de programmation et de simulation qui offre tout un éventaille d’outils et de possibilité de programmer Pepper. Il a été conçu de manière à être un outil de base à l’apprentissage de la programmation sur Pepper. Le document en annexe Choregraphe apporte plus de détails sur cet outil.
* Webot : C’est une autre plateforme de simulation qui permet de créer un monde virtuel dans lequel évolue le robot Pepper en mode virtuel également. Il permet d’effectuer ainsi des tests de situations réelles. Ce outil n’a pas été étudié dans le cadre de ce travail. La préférence a été porté sur Choregraphe, plus utilisé par les developpeurs d’applications sur Pepper. Néamoins, le lien suivant permet d’apprendre d’avantage sur cet outil. <http://doc.aldebaran.com/2-4/software/webots/webots_index.html>
* Les SDK

IL en existe pour tous les langages supportés par Naoqi : le python SDK, c++ SDK, etc. . Se sont des kits de developpement crée par Aldebaran pour permettre le deloppement de programmes sur des machines distantes. Ces programmes peuvent aussi bien s’éxécute à distance et interagir avec le robot ; par exemple accéder à sa mémoire pour y stocker ou préléver des éléments, qu’être chargées dans le robot et exécuté avec les outils tels que Choregraphe. La programmation de Pepper avec ces SDK trouve son advantage dans l’exécution rapide des programmes, le traitement d’image. L’usage des SDK d’Aldebaran est prévu pour les utilisateurs avec des connaissances plus avancées. Le lien suivant permet d’obtenir plus d’informations sur ces SDK. <http://doc.aldebaran.com/2-4/dev/programming_index.html>

## Documentation

La société Aldebaran a également prévu une documentation pour aider dans l’utilisation et la programmation de ses robots. Il s’agit d’un site dynamique disponible à l’adresse suivante : <http://doc.aldebaran.com/2-4/>. On y retrouve la newsletter sur les robots de l’entreprise, l’AppStore où sont téléchargeables des applications commerciales ou non, la communauté de dévéloppeurs, la documentation sur les Frameworks Naoqi et Qi. Il faut noter que cette documentation évolue avec les version des outils dévéloppés par l’entreprise. Par exemple, le lien précédent donne sur la version 2.4.3 de la documentation ; la dernière(2.5) étant reservé aux partenaires uniquement. Les versions précédentes restent disponibles pour les utilisateurs qui utilisent encore les versions précédentes.

1. Documentation des Frameworks Naoqi et Qi

Une documentation détaillée est disponible sur les Frameworks Naoqi et Qi. Il s’agit de bibliothèques sur les modules qui les composent. Ces bibliothèques commencent par présenter la fonctionnalité principal du module puis à donner les méthodes, évenements et signaux qui permettent de l’utiliser. Des cas pratiques d’utilisation de ces modules sont présentés dans les applications dévéloppées plus bas.

1. La communauté Aldebaran

Il existe une communauté de developpeurs qui partage leurs expériences sur les robots de l’entreprise. Ainsi, des projets réalisés sur les robots y sont partagés et un forum de discussion permet d’exposer les problèmes rencontrés lors de la programmation et d’obtenir le soutien des autres programmeurs. Cependant, cette communauté ne comporte pas encore suffisament de membres pour servir de bonne base de documentation.

## Cas pratiques

Le but de cette section est de présenter quelques procédures utiles.

### Connecter Pepper à internet

Lorsque le robot Pepper est allumé pour la première fois, une série de configurations est à effectuer ; parmi lesquelles configurer le wifi. Le choix d’un réseau implique que à chaque démarrage du robot, c’est le wifi auquel il pourra se connecter ; à moins de configurer un nouveau wifi. La procédure est expliquée à la page <http://doc.aldebaran.com/2-4/family/pepper_user_guide/connecting_pep.html>

Un autre cas possible est de connecter le robot Pepper à un switch à travers un câble Ethernet afin d’accéder à internet. En effet, un port Ethernet est disponible à l’arrière de la tête de Pepper, comme le montre la figure suivante :



Figure 11-port Ethernet du robot Pepper (source2, pp. http://doc.aldebaran.com/2-4/family/robots/connectivity\_nao.html)

Il est aussi possible de connecter le robot Pepper à travers un câble USB à son ordinateur ou un autre dispositif externe tel qu’une caméra Kinect, une caméra etc. Ce mode est cependant généralement utilisé pour effectuer la mise à jour du système du robot. Le port USB de Pepper est situé au même endroit que le port Ethernet, comme le montre la figure suivante :

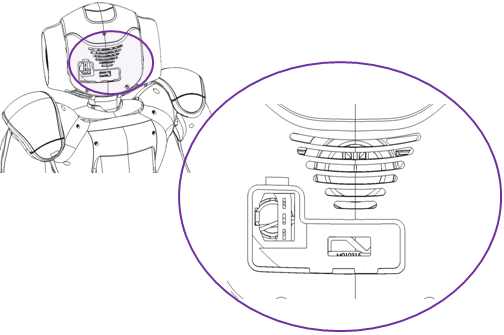


Figure 12-port USB du robot Pepper (source2, pp. http://doc.aldebaran.com/2-4/family/robots/connectivity\_nao.html)

Une fois connecté à internet, il est possible d’accéder à la page internet du robot. Pour cela, il faut connaître son adresse IP. Pour connaître l’adresse IP à laquelle le robot Pepper est connecté, il suffit de presser le bouton sous sa tablette légèrement et une seule fois. Pepper indique son adresse IP dans la langue actuellement installée. Pour réentendre l’adresse IP, il faut presser de nouveau le bouton sous la tablette et Pepper épèle à nouveau.

Pour accéder donc à la page di robot, il faut dans le navigateur saisir l’adresse IP du robot. Se connecter en entrant le nom d’utilisateur « Nao » et le mot de passe celui du robot défini lors du premier démarrage de Pepper. Il faut savoir que les données personnelles du robot telles que son mot de passe ne peuvent plus être modifier à moins de réinitialiser tout le système. La procédure de réinitialisation du mot de passe est expliquée à la page de cette adresse <http://doc.aldebaran.com/2-4/family/pepper_user_guide/erase_memory_pep.html#erase-memory-pep>.

La page internet du robot se présente ainsi :

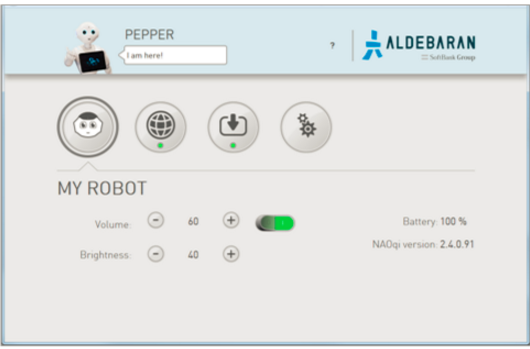


Figure 13-page internet du robot Pepper

Il s’agit d’une page qui répertorie toutes les paramètres systèmes du robot telles que le nombre d’application installée, la version utilisée, le réseau configuré, son état etc. avec la possibilité de modifier ces paramètres. C’est sur cette page que se trouve les boutons « reboot » et « reset factory » qui permettent de réinitialiser tout le système.

### Changer la langue de dialogue de Pepper

La langue par défaut installer sur Pepper est l’anglais. C’est celle avec laquelle est livré le robot. Cependant, il est possible d’installer jusqu’à 7 autres langues dans Pepper. La seconde langue choisie est gratuite mais pour installer les autres, il faudrait les acheter. Comment installer une seconde langue dans Pepper ?

Pour installer une seconde langue, il faut au préalable avoir un compte de la communauté Softbank. Pour en créer un se rendre sur l’adresse suivante : <https://developer.softbankrobotics.com/> et cliquer sur « Sign in » puis « sign up ». Ce compte est nécessaire pour accéder à de nombreux paramétrages du robot tels que l’installation de nouvelles applications, l’accès au forum de discussion, à la page de téléchargement des outils, et bien évidement pour changer la langue du robot. Il faut aussi que le robot soit enregistré à ce compte. Il s’agit ainsi de définir le propriétaire du robot, lequel peut être modifié à tout moment. Un robot ne peut être connecté qu’à un seul compte. Pour enregistrer un robot à un compte, il suffit d

Après connexion au compte de la communauté Softbank, se rendre à la page ….

### Accéder au robot par SSH

# Développement

## Objectif

Le but de cette partie est de présenter quelques applications qui ont été dévéloppées en respectant des scénarios décrits préalablement.

## 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 plateforme de programmation Chorégraphe dont le langage de support est le python, le niveau d’abstraction du langage et aussi par le désir d’apprendre un nouveau langage.
* Os : toutes les applications ont été développées et testées uniquement sur Windows 7
* Plateforme de développement : Pour la programmation ainsi que le test des applications développées, l’outil Chorégraphe (version 2.4) a été utilisé. Voir le document Choregraphe en annexe pour plus de détails
* Framework : Pour la programmation des comportements dans Pepper, le Framework NAOqi (version 2.4) a été utilisé.
* Système de versions : pour la gestion des versions, le logiciel Git a été utilisé.
* Diagrammes : Création des diagrammes UML à l’aide du logiciel ArgoUML.

## Plan de présentation

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

* Objectif : il s’agit de décrire de manière succincte l’objectif de l’application.
* Description du scénario : description 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.
* Analyse : Cette section a pour but de faire ressortir les éléments clés de l’application décrite dans le scénario et de présenter comment ils ont été implémentés. Aussi, les modules de Naoqi utilisés seront présentés.

## Applications

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

##### Objectif

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

##### Description du scénario

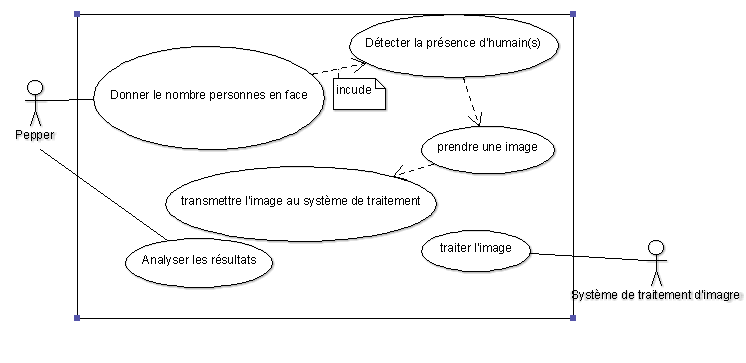


Tableau 1-diagramme des cas d'utilisation - Application 1

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

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 :

1.a Si aucune personne n’est détectée, Pepper le dit

Scénario d’exception :

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

##### Analyse

Pour réaliser le scénario décrit précédemment, 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. Pour implémenter cette fonctionnalité, le module AlFaceDetected de Naoqi a été utilisé. Les étapes suivantes permettent de comprendre comment :

* Premièrement, il a été instancié . Lorsqu’on instancie ce module, le système souscrit à l’événement « FaceDetected » et fait appel aux services développés par la société OKI. Il s’agit d’une société d’électronique 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. Une fois le traitement d’image terminé, les résultats sont stockés dans la mémoire de Pepper avec la clé « Facedetected ».
* La deuxième étape de l’impémentation a donc consisté à récupérer en mémoire les données stockées sous cette clé et à les analyser. 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 effectuer la détection. La première valeur est le temps en séconde et la deuxième, en minisécondes.
* 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.
* La troisième étape a consisté à éffectuer une comparaison à zéro de la taille du tableau FaceInfo[N] (voir le deuxième élément de la liste ci-dessus) pour déterminer si des personnes ont été détectés ou pas.

Il est 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.

1. Dire 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 ses méthodes « say » et « setLanguage » ont été utilisés. Ce module fonctionne ainsi : 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 moteur de parole. Selon la langue choisie, le logiciel est différent. 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 du module ALTextToSpeech(changer la vitesse, mettre des pauses entre les mots, etc.)**

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. C’est cette méthode qui a donc été utilisée pour faire des appels asynchrones tous les 20 secondes sur une fonction de timing. La fonction de timing a pour rôle unique d’indiquer oralement que le timing est atteint.

### Application 2 : Pepper dialogue avec ses interlocuteurs

##### Objectif

L’objectif de cette application est progammer des boîtes de dialogues entre Pepper et l’humain.

##### 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 réjouis de le savoir

Humain : Parle-moi de tes origines

Pepper : J’ai été conçu par la société française Aldebaran en 2012. Le saviez-vous ?

Humain : Non

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

Humain : Oui

Pepper : Ok !

##### Analyse

C’est le module ALDialog de Naoqi qui a permis de créer le dialogue décrit ci-dessus. C’est un module qui définit des règles de syntaxe et d’écriture permettant de créer et enrichir les dialogues. Le lien suivant donne plus de détails sur ce module. <http://doc.aldebaran.com/2-4/naoqi/audio/altexttospeech.html>. Pour créer le dialogue de cette application, il a suffit de respecter ces règles d’écriture et de syntaxe.

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

##### Objectif

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

##### Description du scénario

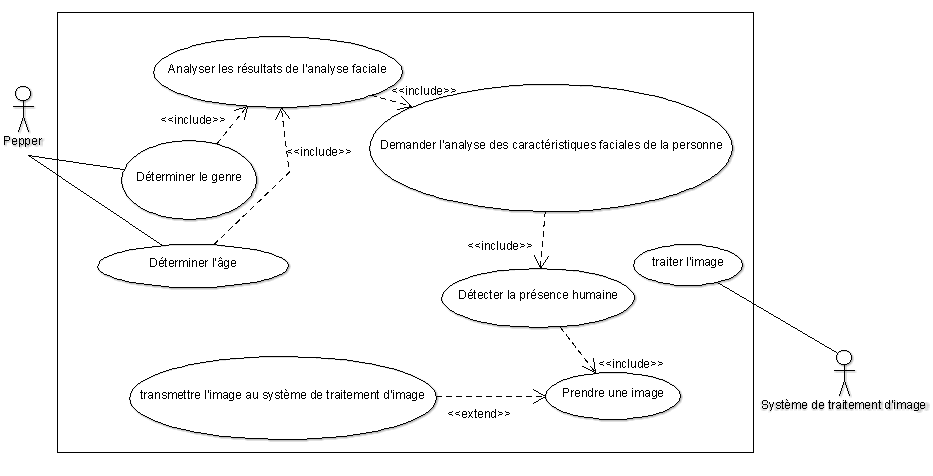


Tableau 2-diagramme des cas d'utilisation - Application 3

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.

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 détectée, ou plusieurs, Pepper le dit.

3.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 évènement survient comme par exemple, le « timeout » atteint, Pepper indique la fin de l’interaction.

Performance attendue :

Le robot doit reconnaître une personne déjà analysée.

##### Analyse

Pour réaliser le scénario décrit précédemment, 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, le module ALFaceDetection a été utilisé. Pour cette application, C’est le module ALPeoplePerception qui a été permis d’ implémenter la détection de personnes. Il s’agit d’un module qui permet d’obtenir la liste des personnes visibles à chaque changement de la population en face du robot ; ce qui peut donc être utilisé pour savoir si des personnes sont présentes ou pas. Pour obtenir cette liste de personnes visibles, le module fait appel aux données recueillies par les cameras 3D et 2D de Pepper et utilise la technique « visual cues » pour identifier les personnes et les stocker en mémoire sous forme d’une liste d’identifiant pour chaque personne trouvée. Pour disposer de cette liste d’id, il a suffit d’aller en mémoire chercher cette liste sous la clé « PeoplePerception/PeopleList ».

1. Demander l’analyse des caractéristiques faciales

Pour effectuer l’analyse des visages qu’il voit, Pepper doit explicitement demander l’analyse. Pour ce faire, il existe dans Naoqi le module ALFaceCharacteristics qui permet de donner cet ordre d’analyse en passant en paramètre la liste des personnes à analyser les caractéristiques. Cette liste est celle déterminée au point précédent. Les résultats de l’analyse des caractéristiques de chaque personne détectée est disponible en mémoire sous les clés « PeoplePerception/Person/id /GenderProperties» et « PeoplePerception/Person/id /AgeProperties» où id est l’identifiant de la personne qui a été analyser .

Il faut noter que le module ALFaceCharacteristics, appelle automatiquement le module ALPeoplePerception. Il n’a donc pas été nécessaire de l’appeler explicitement.

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.

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. Cette fonctionnalité n’a pas été exploitée dans cette application. Elle le sera dans une autre application. Pour reconnaître ici les personnes déjà analysée, il a été stocké dans un tableau, pour chaque id de personne, son genre et son âge. Lors du prochain passage, ce tableau a été repris pour comparer avec les données de la nouvelle analyse et déterminer si la personne avait déjà été analysée et si les résultats sont les même.

1. Gestion du timeout

Le Framework Qi a été utilisé comme a l’application 1.

### Application 4 : Pepper détermine la couleur de vêtements avec Open CV

##### Objectif

L’objectif de cette application est de programmer Pepper pour qu’il interagisse avec le librairie open CV la librairie 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

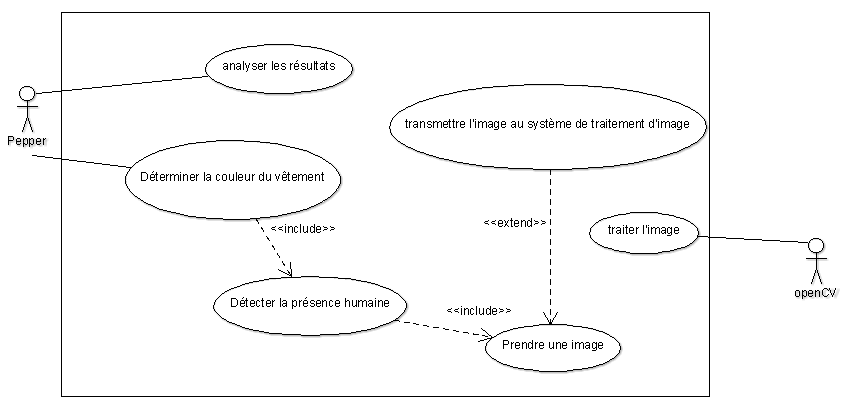


Tableau 3-diagramme des cas d'utilisation - Application 4

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.

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 open CV
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 détectée, ou plusieurs, Pepper le dit.

Scénario d’exception :

Si un évènement 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 secondes après le démarrage.

##### Analyse

Pour réaliser le scénario décrit précédemment, 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. Un 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 mouvements, 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 :

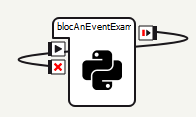


Figure 5-Bloquer événement« ALBasicAwareness/HumanTracked »

1. Prendre une image et la transmettre à open CV

Open CV est une librairie externe à Chorégraphe 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 caméra du robot. Il suffit d’enregistrer cette caméra 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 10 : ALVideoDevice.

1. Analyse des résultats

La librairie open CV renvoie la couleur dominante de l’image qui lui a été transmise. Pour en savoir plus le fonctionnement de la librairie open CV, aller é l’annexe 11 : open CV.

1. Donner la couleur du vêtement

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

1. Gestion du timeout

Pareil que dans les applications précédentes.

### 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

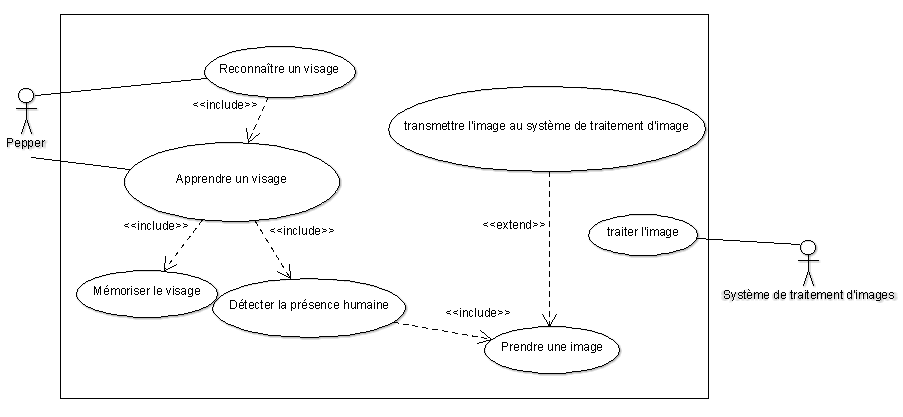


Tableau 4-diagramme des cas d'utilisation - Application 5

Préconditions : Le robot doit être allumé, l’application chargé dans le robot, l’environnement bien éclairé. Pour de meilleures 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)

* Scénario principal : 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
* Reçoit les résultats et les analyse
* Si le visage de la personne lui est familier, il le mentionne
* Sinon, s’il ne reconnaît pas le visage
* Affiche un formulaire sur sa tablette
* Demande à son interlocuteur d’entrer son nom
* Apprend son visage en le mémorisant avec le nom saisi

Scénario alternatif :

* Si l’utilisateur clique sur le bouton supprimer du formulaire, son image n’est pas mémorisée.

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 précédemment, 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 tablette 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 12 : Module ALTabletService

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 qui permet également de mémoriser les visages et de faire la reconnaissance. Pour mémoriser, il suffit d’utiliser la fonction learnFace et reLearnFace pour réapprendre 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 4 : ALFacedetection

# Projet

## Description

Comme spécifié dans le cahier des charges, un projet a été mené afin de clôturer les activités de recherche, documentation et de développement menés au cours des premières phases de ce travail. Il a été question de programmer Pepper afin qu’il interagisse avec les personnes présentes aux portes ouvertes de la HEIG-VD 2017. L’objectif étant de les informer et distraire. Il a donc été mis sur pied un programme informatif et ludique avec les fonctionnalités suivantes : présentation du thème des portes ouvertes, jeu ludique sur les émotions, animations.

* Présentation du thème des portes ouvertes

Le thème des portes ouvertes de cette année 2017 a été « la révolution numérique ». Pour présenter le thème de cette édition, Pepper a été programmer afin de reconnaître et présenter quelques personnages historiques ayant influencé la révolution numérique.

* Jeu ludique sur les émotions

Le jeu « Smiley Game » a été développé dans le but de distraire les visiteurs. Il s’agit d’un jeu où le robot Pepper détermine l’émotion présente sur le visage du joueur et affiche un smiley correspondant sur sa tablette.

* Animations

Une série d’animations exécutées par Pepper ont été prévues en absence de visiteurs.

## Phase d’analyse

Il s’agit dans cette section de présenter les études qui ont été menées afin de garantir de bons résultats pour le projet

* Etude des besoins

Les besoins suivants nécessitent d’être fortement pris en compte dans réalisation du projet :

* Interaction naturelle homme-machine : Il est important non seulement que le robot interagisse avec le public mais aussi que cette interaction se fasse le plus naturellement possible afin de créer une bonne ambiance mais aussi pour présenter la capacité du robot à avoir un comportement proche de l’humain
* Autonomie : Il est important que le robot évolue lors de la phase de présentation du projet le plus indépendamment possible.
* Attrait : Il est important que les programmes développés dans le cadre du projet soient attrayants
* Variation : Il est important que le projet présente des activités variées afin d’attirer constamment les gens.
* Divertissement : Il est important que lors de la présentation des activités du projet, que les personnes présentes s’amusent.
* Etude technique

Un parcours des fonctionnalités de Pepper a été effectuée afin de décider vers lesquels s’orienter pour le projet ; Pour cela, une évaluation des performances des différentes fonctionnalités du robot exploitées lors de la phase de recherche et développement a été menées. Les résultats des limites de certaines fonctionnalités sont présentés dans le document annexe : A quoi faut-il faire attention. Les fonctionnalités présentent dans les programmes de ce projet sont donc celles qui ont parues plus fiables.

* Cadrage du projet

Il est question dans cette partie de présenter comment a été décidé le choix des activités du projet.

En tenant compte des spécifications du cahier des charges au sujet du projet ainsi que l’étude des besoins, il a été décidé de créer un premier programme en rapport avec l’événement des portes ouverts. Il a donc été décidé de créer une application à la fois informative, divertissante et interactive où le robot reconnaîtrait et présenterait quelques personnages de la révolution numérique.

* Planification du projet

Se référer au document suivant en annexe : journal de bord

## Phase de conception

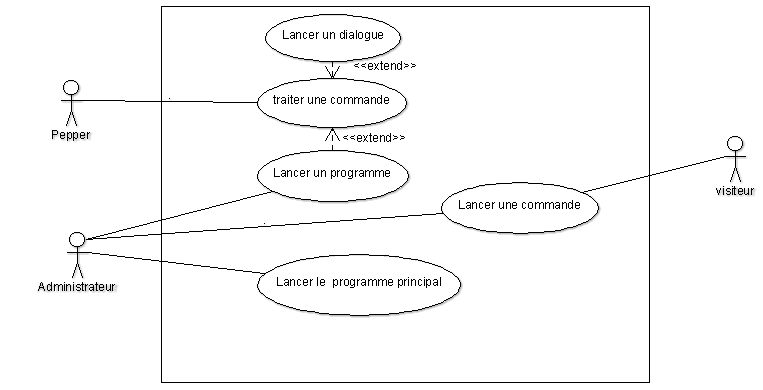


Tableau 5-diagramme des cas d'utilisation-Projet

Terminologie :

* Administrateur : C’est la personne en charge du robot et du projet et qui pendant l’évènement est chargé de coordonner ses activités avec les humains.
* Visiteur : Il s’agit de toute personne attirée par le robot et désireuse d’interagir avec lui.
* Commande : Pepper a été programmé afin de réagir à certains mots ou textes. Une commande est donc tout mot ou texte dit au robot et peut provoquer une action de sa part.
* Programme principal : Il s’agit ici du programme conçu pour l’événement comprenant la présentation du thème des portes ouvertes, le jeu d’émotions ainsi les animations.
* Programme : Il peut s’agir des sous programmes du programme principal ou de tout autre application installée dans le robot.
* Dialogue : Il s’agit des dialogues prédéfinis et enregistrés dans le robot.

Etude du Cas d’utilisation 1 : Lancer le programme principal

Pour lancer le programme principal, l’administrateur doit au préalable le charger dans Chorégraphe, s’assurer que le robot est connecté à un réseau avant de cliquer sur le bouton « Play » de Chorégraphe pour charger le programme dans le robot et l’exécuter. Pour connaître comment utiliser Chorégraphe, se rendre à l’annexe 1 : Chorégraphe.

* Fonctionnement du programme principal

Les fonctionnalités du programme principal ont été développées sous la forme de trois programmes indépendants mais connectés entre eux. On distingue donc un programme « Numerical History » pour la présentation des personnages historiques, un programme « Smiley Game » pour le jeu d’émotions et un programme « animations » pour les animations avec le robot. Les programmes « Numerical History » et « Smiley Game » nécessite l’interaction avec l’homme tandis que le programme « animations » fait intervenir uniquement le robot. Ces deux programmes d’interaction ont donc été regroupés dans le programme « H-M interaction » pour interaction homme machine.

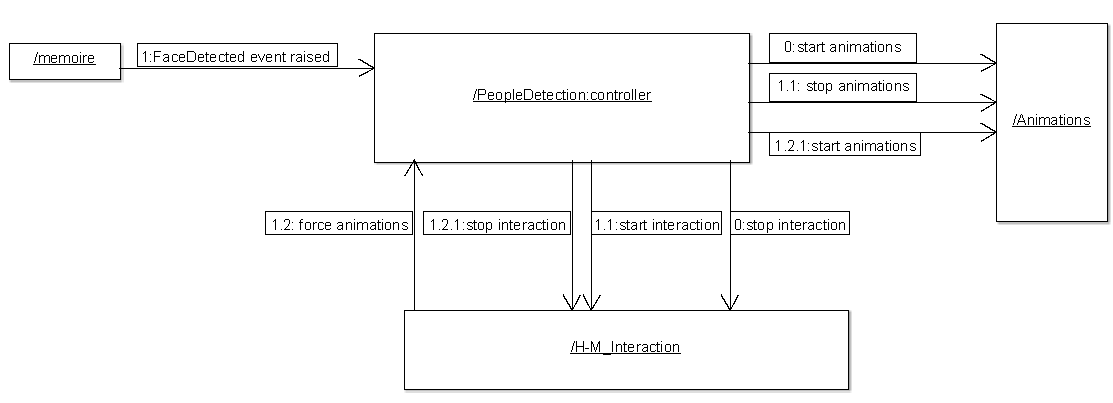


Tableau 6-diagramme de collaboration-programme principal du projet

Sur la figure ci-dessus traduit la connexion entre les programmes du programme principal. Cette connexion est établie et coordonnée par la classe PeopleDetection qui joue le rôle de Controller.

Description :

Au début de l’exécution du programme principal, le contrôleur (classe PeopleDetection) donne la main au programme « animations » où des animations sont exécutées. Lorsqu’une personne est détectée, le contrôleur stop les animations et lance le programme « H-M interaction ». Si pour une raison ou une autre l’interaction est stoppée, le programme « H-M interaction ». Envoie au contrôleur un signal de lancer les animations ; le contrôleur stop donc le programme « H-M interaction » et lance le programme « animations ».

* Fonctionnement du contrôleur

La classe PeopleDetection joue le rôle de contrôleur dans l’exécution du programme principal. En effet, c’est elle qui coordonne l’arrêt ou le démarrage des programmes qui constituent le programme principal. Le diagramme suivant montre son fonctionnement en interne.

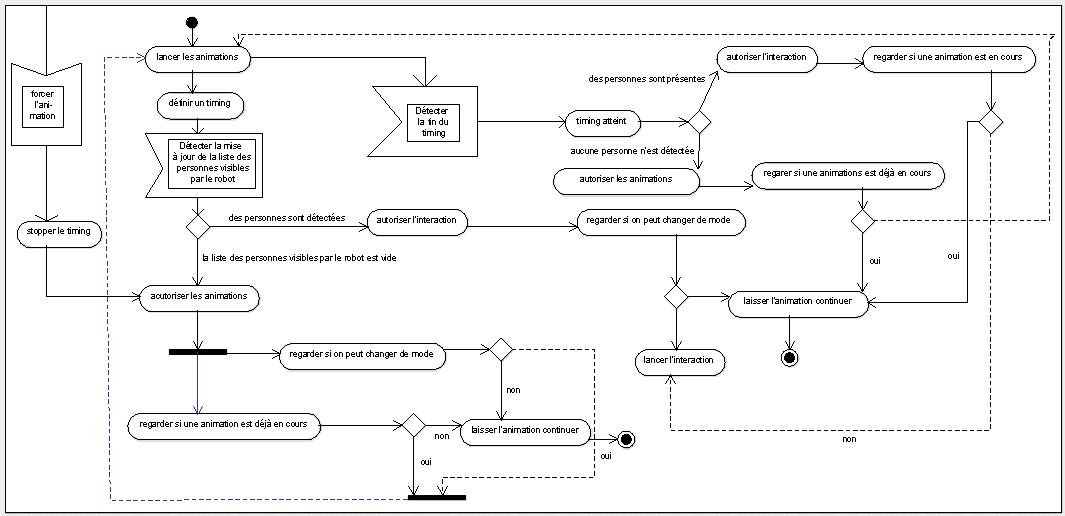


Tableau 7-diagramme d'activité du contrôleur-projet

* Fonctionnement du programme « animations »

Le but de ce programme est de lancer des animations en absence de personnes. Ces derniers pourront ainsi être attiré et se rapprocher.

ALAnimatedSpeech est un module du Framework Naoqi, qui permet au robot de s’exprimer à travers des gestes. L’usage de ce module est expliqué en l’annexe : ALAnimatedSpeech. Ce module met à disposition une base de données d’animations. Les éléments de cette base sont des mouvements que peut faire Pepper auquel sont associés des mots clés sous forme de nom d’animations. Par exemple, le mot clé animations/Stand/Gestures/Please\_1 permet à Pepper d’exécuter un mouvement exprimant une demande.

Le fonctionnement du programme « animations » se base donc sur une table contenant quelqu’une de ses animations choisies aléatoirement à chaque fois qu’une instruction de lancer une animation est effectuée.

* Fonctionnement du programme « H-M Interaction »

Comme mentionné plus haut, « H-M Interaction » est une association de deux programmes nécessitant l’interaction homme-machine « Numerical History » et « Smiley Game ». Ces deux programmes sont indépendants l’un de l’autre mais ont été associés afin de permettre une cohérence du programme principal. Le diagramme suivant permet d’illustrer ce qui précède :

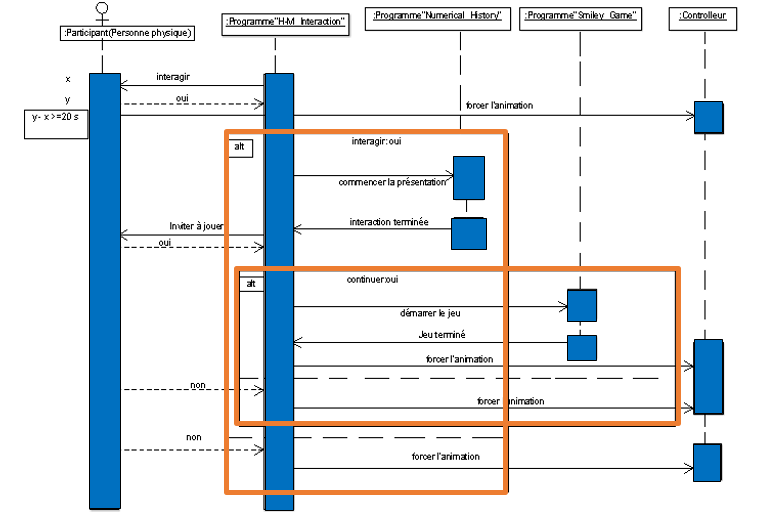


Tableau 8-diagramme de séquence du programme "H-M Interaction « -Projet

1. Fonctionnement du programme « Numerical History »

Ce programme est basé sur la reconnaissance d’images. En effet, Pepper est capable de reconnaître des personnes, des objets, des mouvements et des images. Pour ce programme, le scénario suivant a été mis en place :

Scénario principal :

1. Pepper accueille le joueur et l’invite à saisir son nom sur sa tablette
2. Pepper énonce le thème des portes ouvertes
3. Demande au joueur de lui montrer une image d’un personnage de la révolution numérique
4. Pepper reconnait et présente le personnage
5. Pepper propose d’apprendre de d’autres personnages

Scénario alternatif

1. Si après 30 secondes le joueur n’a pas saisie son nom ou sa saisie est incorrect, retour au point 1
2. Si Pepper ne reconnaît pas le personnage sur l’image, il indique que l’image lui est inconnue
3. Si après 20 secondes Pepper n’a pas donné de résultat, retourne au point 3
4. Le joueur veut apprendre de d’autres personnages, retour au point 3
5. Le joueur veut interrompre la présentation
6. Si parmi les images proposées, celles des personnages Suisse ayant participé à la révolution numérique n’ont pas été tous vus, Pepper l’indique au joueur et lui propose de continuer la présentation
7. Si le joueur accepte de continuer, retour au point 3
8. Sinon, Pepper arrête la présentation et continue avec le jeu « smiley Game »
9. Si après 60 secondes, aucune réponse n’est donnée, Pepper l’indique, stop la présentation et redonne la main au contrôleur
10. Sinon, Pepper arrête la présentation et continue avec le jeu « smiley Game »

1. Si après 60 secondes, aucune réponse n’est donnée, Pepper l’indique, stop la présentation et redonne la main au contrôleur
2. Fonctionnement du programme « Smiley Game »

« Smiley Game » est un jeu d’interaction avec l’homme où le Pepper essaye de déterminer l’émotion sur le visage du joueur et d’afficher un smiley correspondant sur sa tablette. Les émotions trouvées ne peuvent indiquer que joueur soit dans cette liste : neutre, fâché, triste, étonné, surpris. Les autres n’ayant pas été pris en compte.

Le fonctionnement de ce programme est défini à l’aide du scénario suivant :

Scénario principal :

Pepper présente le principe du jeu et demande au joueur s’il veut jouer

1. Si oui

* Pepper indique les instructions du jeu
* Donne oralement l’émotion déterminée et affiche le smiley correspondant sur sa tablette
* Demande à faire une autre grimace et retourne au point 1.a.ii
* Au bout de 3 occurrences, Pepper demande au joueur s’il veut continuer le jeu
* Si oui, retour au point 1.a
* Si non, Pepper demande si quelqu’un d’autre veut joueur
* Si oui, retour au point 1.a
* Si non, Pepper stop le jeu et redonne la main au contrôleur

1. Si non, Pepper retourne redonne la main au contrôleur

## Phase de développement

* Plateforme et langage de programmation utilisé

Le langage de programmation utilisé est le python et l’ensemble des programmes ont été tous développés sur Chorégraphe.

* Technologies utilisées

Les deux Framework utilisés sont Naoqi et qi. Une documentation sur chacun de ses Framework est prévu en annexe.

* Les principales fonctionnalités du projet
* La reconnaissance d’image

Pour effectuer la reconnaissance d’images, Pepper doit dans un premier temps apprendre ses images et les mémoriser puis dans un second temps les reconnaître.

L’apprentissage ne se fait pas via l’écriture du code mais par un ensemble de manipulations qui consistent à capturer l’image vue par Pepper et à l’enregistrer dans une base de données intégrée au robot à l’en local sur l’ordinateur.

La reconnaissance quant à elle consiste à capturer l’image présenter lors de l’exercice de reconnaissance et de la comparer aux images de la base de données. La comparaison découle d’un processus de traitement d’image.

Les processus de capture d’image, de traitement d’image et de comparaison sont effectués à l’intérieur du robot par des algorithmes performants mais inaccessibles. Pour en savoir plus sur ces processus de capture et de traitement d’image, se rendre au document Prise et traitement d’image par Pepper situé en annexe. L’implémentation de cette fonctionnalité dans ce projet a donc consister à effectuer les manipulations nécessaires pour l’apprentissage puis à utiliser l’outil de Visio reconnaissance présent dans Chorégraphe. Des éléments ont été mis en place afin de permettre le bon fonctionnement de ce caractère fonctionnel. Il s’est agi de programmer une posture à adopter par le robot lorsque débute la reconnaissance de l’image. En effet, il a été constaté que Pepper bouge beaucoup la tête lors de la phase de reconnaissance et donc le temps d’analyse est prolongé. En outre, il a été également important de s’attarder sur la qualité et le format des images utilisées pour la reconnaissance. Certaines propriétés ont permis d’accélérer également le processus de reconnaissance. Les résultats de ses expériences sont présentés en annexe dans le document A quoi faut-il faire attention.

* L’interaction avec la tablette

La tablette de Pepper a été utilisée afin de favoriser l’interaction avec le public. En effet, il a été question de la programmer premièrement dans le programme « Numerical History » pour l’identification et deuxièmement dans le jeu « Smiley Game » afin d’afficher le bouton de démarrage du jeu ainsi que les smileys correspondants aux émotions trouvées. Pour effectuer cela, le module ALTabletService du Framework Naoqi dont la description est donnée en annexe a été utilisé. Il fournit dans son API une liste de méthodes, d’événements et de signaux qui permettent de manipuler la tablette de Pepper et d’y charger des applications. Pour l’identification sur la tablette dans le programme de présentation des personnages historiques, des méthodes de ce module ont permis d’afficher un champ de saisie de texte et d’y créer un évènement qui se déclenche lorsqu’une saisie est effectuée. Il a ensuite été mis sur pied une fonction de callback associée à l’évènement crée et permettant de gérer la saisie des utilisateurs. Pour le jeu sur les émotions, il a été développé une application en JavaScript permettant de lancer le jeu en cliquant sur un bouton sur la tablette. Cette application a été développé à l’extérieur de Chorégraphe. La raison est présentée dans le document A quoi faut-il faire attention située en annexe. Pour la charger dans le robot, il a de nouveau eu besoin d’utiliser le module ALTabletService et le module ALMemory toujours de Naoqi afin de créer une communication entre l’application JavaScript et le programme python tournant sur Chorégraphe. Pour afficher les smiley, ALTabletService propose des méthodes qui permettent d’afficher des images sur la tablette du robot.

* La détermination des émotions.

L’idée principal du jeu « Smiley » est d’amener Pepper à déterminer les expressions faciales. En effet, Pepper est capable à l’aide des programmes de traitement d’images de déterminer les caractéristiques d’un visage et ensuite les analyser à l’aide d’algorithmes performants pour déterminer les expressions du visage, le genre, le sexe, le sourire et les stocker dans la mémoire sous forme de valeur. Pour les expressions du visage, seules les expressions suivantes peuvent être déterminées : neutre, content, surpris, en colère, triste. La valeur stockée est comprise entre 0 et 1 et représente la probabilité que l’expression ait été émise. Comme déjà mentionné, les algorithmes qui permettent de traiter les images ne sont pas accessibles ; ainsi que ceux qui permettent d’analyser les caractéristiques du visage. Cependant, il existe dans Naoqi le module ALFaceCharactéristique qui définit un ensemble de méthodes, évènements et signaux qui permettent de manipuler ses données. C’est ce module qui a été utilisé pour implémenter cette fonction. Puisque les données sur les expressions stockées en mémoire se présente sous la forme d’une valeur de certitude comprise entre 0 et 1 pour chacune des 5 expressions, il a aussi été mis sur pied un algorithme permettant de déterminer précisément l’expression émise. Cet algorithme consiste à demander six fois l’analyse d’un même visage et à récupérer l’expression ayant au moins deux fois la plus grande valeur comme l’expression émise.

* L’interaction orale

Pour permettre une interaction naturelle, il a été programmé des dialogues permettant un échange entre le robot et son entourage. Ces dialogues ont été aussi conçus afin de guider la séquence des activités du projet. C’est le module ALDialog de Naoqi qui a été exploité afin de créer ces dialogues. En effet, Il définit un ensemble de règles et de concepts qui permettent de créer de bons dialogues.

* Eléments importants
* Asynchronisation des programmes

Afin de gérer l’exécution les différents programmes du programme principal, il a été impératif de mettre sur pied un système d’asynchronisation. Pour ce faire, il a été programmé une classe contrôleur(PeopleDectection) donnant ou retirant le droit d’exécution aux différents programmes et un système d’envoi de communication entre les différentes entités. C’est en se basant sur les diagrammes établis lors de la phase de conception que des fonctions ont été implémentées dans le contrôleur gérant chacun des programmes. Pour la communication entre les différentes entités, les ports d’entrée et sortie des boîtes de Chorégraphe ont été utilisées.

* Gestion des timings

Pour favoriser une interaction naturelle entre le robot et l’homme, il a été implémenté des fonctions de timing permettant de surveiller les différentes activités. Par exemple, des fonctions de timing ont été définies pour vérifier que le robot ne reste pas bloqué, en attente d’une réponse de son interlocuteur. Pour implémenter ses fonctions, il a été exploité la méthode « async » fournie par le Framework qi, permettant de créer des activités asynchrones. Avec cette méthode, il a été possible de créer des activités de manière asynchrone, permettant l’exécution des fonctions de timing après un temps définis lorsque certaines conditions n’ont pas été remplies mais aussi de les stopper lorsqu’elles l’ont été.

## Phase de tests

Le but de cette partie est de présenter les tests qui ont été menés. L’objectif des tests a été de vérifier les points suivants :

* Les programmes d’exécutent correctement
* Les programmes respectent les spécifications énoncées lors de la conception telles que l’identification sur tablette, la reconnaissance des images, la reconnaissance des émotions etc.
* L’asynchronisme et la collaboration fonctionne bien entre les différents programmes
* Les timings sont bien gérés
* Les programmes sont distrayants
* Les étapes

Les tests réalisés dans ce projet ont été effectuées en trois étapes bien distingues :

* La première étape de test a consisté a effectué des tests unitaires sur chaque élément du système. Elle a été menée lors du développement des différentes applications. En effet, à chaque implémentation d’une fonctionnalité, un test a été effectué afin de vérifier qu’il fonctionne avant d’être incorporé au reste du système.
* La deuxième phase de test a consisté à effectuer des tests d’intégrations et système.  Elle a été menée sur tout le système, à la fin du développement du projet afin de s’assurer de la cohérence entre les différentes activités (dialogues, jeu, animations) et de déceler les erreurs ignorées jusque-là.
* La troisième et la dernière phase de test a consisté à effectuer un test d’acceptation. Elle a été menée premièrement en présence de l’expert puis sur le site de l’événement des portes ouvertes, afin d’effectuer une mise en situation du robot. Il a donc été testé la connection du robot au réseau, l’éclairage qui agit sur la qualité de vision du robot, le fonctionnement global du projet
* Procédure

Il s’agit de décrire comment ont été effectué les tests.

* Tests unitaires

Les tests unitaires sur les éléments du système ont consisté à valider le fonctionnement des principales fonctionnalités de ces éléments. Par exemple, pour tester le programme « Numerical History », l’accent a été porté sur la reconnaissance des images. Ainsi, plusieurs fois le robot a été confronté à reconnaître les images de ces programmes. Aussi, il a été contrôlé que le temps nécessaire pour la reconnaissance est relativement cours, que le robot ne reste pas bloqué dans un dialogue ; soit que les timing effectionne bien, que les joueurs peuvent joueur autant de fois pour peu qu’il demande à continuer la présentation.

* Test d’intégration et système

Pour effectuer ces tests, différentes situations ont été testées afin de vérifier tous les cas de figure. La liste suivante présente ces situations :

* Test d’acceptation

Pour le test d’acceptation, le robot a été connecté au réseau, le projet exécuté plusieurs fois afin de vérifier le fonctionnement de la reconnaissance sur ces conditions d’éclairage, mais aussi la cohérence des différentes activités.

* Résultats

Les procédures de tests ci-dessus ont laissé transparaitre lors du test d’acceptation sur le site de l’événement, un problème avec le bruit. En effet, il a été constaté que le bruit perturbait la cohérence du système ; en ce sens que Pepper avait du mal à écouter ce qui lui était dit, mais aussi à garder l’attention sur le programme en train d’être exécuté. Pour pallier à ce problème, il a été décidé de placer Pepper lors de l’événement dans un emplacement avec moins de bruit. Toutes fois, les autres tests ont révélés des résultats satisfaisants.

# Chapitre 5 : Conclusion

Au terme du travail qui nous a été confié dans le cadre de notre projet de fin d’études, il est temps d’effectuer un bilan. Ainsi, les sections suivantes présenteront dans l’ordre les résultats obtenus, les difficultés rencontrés et les expériences acquises.

## Les résultats obtenus

### Phase 1 : Recherche et documentation

Les travaux de recherches menées sur le robot nous ont permis de comprendre le fonctionnement du robot Pepper et d’explorer ses fonctionnalités. Les résultats de ses recherches ont été présenté dans le chapitre 2 du corps de ce rapport.

### Phase 2 : Recherche et développement

Les travaux de recherche et de développement menées nous ont permis de comprendre et de nous s’imprégner de la programmation le robot. Les travaux de recherche sur les processus de programmation du robot ainsi que les applications développées sont présentés au chapitre 3 du corps de ce rapport. Par ailleurs, lors de cette phase, quelques limites ont été décelée sur certaines fonctionnalités du robot. Il s’agit de :

* La détection des personnes : Lors des tests impliquant la détection de personnes, il a été remarqué qu’outre l’éclairage qui peut influencer sur les résultats, la couleur de la peau pourrais également avoir une influence. En effet, des tests ont été menés avec des personnes de couleur blanche et noire et le résultat variait beaucoup selon la personne en face du robot. Plus précisément, il faudrait apparemment plus d’éclairage pour une personne de couleur noire et que pour une personne de couleur blanche. Des tests plus poussés n’ont pas été mené. Il s’agit donc que d’hypothèses.
* La reconnaissance des personnes : Il a été remarqué à travers des tests qui ont été menés que le processus de reconnaissance n’est pas très fiable. Il est arrivé dans certains cas que Pepper ne se souviennent pas d’un visage qu’il a appris.
* La gestion des événements et des signaux : Il a été décelé un problème avec la gestion des événements et des signaux. En effet, il a été impossible de stopper certains événements et signaux après qu’ils aient été lancés. C’est le cas de l’événement « FaceDetected » qui est lancé lorsqu’une personne au moins est détectée, « onInputDialog » qui est lancé quand un champ de texte est saisi sur la tablette du robot, pour ne citer que ceux-ci. Il existe un moyen de s’inscrire à ces évènements mais la désinscription ne fonctionne pas. Il faudrait donc créer un court-circuit sur la boîte pour ignorer le reste d’informations qui déclenchées à la chaine par l’événement ; sinon, le robot reste dans une boucle infinie.

D’autres éléments ont été relevé et sont présentées dans le document « A quoi faut-il faire attention » situé en annexe.

### Phase 3 : Projet

* Le projet réalisé a dans l’ensemble fourni des résultats satisfaisants. Les applications développées sont fonctionnelles et permettent l’interaction souhaitée et sont également distrayantes. Toutes fois, quelques points sont à relever :
* Lors des portes ouvertes, il a été constaté que les bruits posaient un réel problème pour Pepper dans la cohérence de ses activités. En effet, malgré le fait qu’on a essayé de l’isolé dans un endroit plus calme, il n’a pas été possible de d’empêcher totalement les bruits. Pepper interrompait donc des activités en cours pour en exécuter d’autres. Un a regard a été porté sur ce problème et des esquisses de solutions sont présentées en annexe au document « A quoi faut-il faire attention »
* Toujours lors des portes ouvertes, nous avons eu des soucis avec le réseau auquel était connecté le robot. En effet, dans un premier temps, Pepper ne parvenait plus à se connecter à réseau wifi auquel il s’était connecté au paravent. Par la suite, nous l’avons connecté à l’aide d’un câble thernet mais il ne cessait de se déconnecter à tout moment ; ce qui a nuit à un partie de la présentation. Il serait peut-être important de voir si la charge du réseau n’a pas été à l’origine de ce problème vu le nombre de personnes connectées en même temps.

## Les difficultés rencontrées

La principale difficulté rencontrée dans ce travail est le manque de documentation. En effet, hors mis les documentations officielles sur les API des modules des framework Naoqi et qi, il n’existe pas d’autre source de documentation. Aussi, la communauté des développeurs n’est pas encore assez riche ; ce qui limites considérablement l’aide à la programmation du robot.

## Les expériences acquises

C’est avec beaucoup d’intérêt que nous avons réalisé ce travail et avec satisfaction que nous arrivons à la fin. En effet, ce travail nous a permis d’apprendre comment programmer un robot, à avoir les bases du langage Python, à gérer un projet, à effectuer des travaux de recherche. Nous espérons avoir de nouveau l’occasion d’effectuer un projet portant sur un robot.

## Annexe 1 : Chorégraphe

Chorégraphe est une plateforme crée par le groupe SoftBank Robotics pour aider les programmeurs dans le développement 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 n’a jamais programmé un robot.

En effet, chorégraphe a été conçu pour aider les programmeurs sans expérience en robotique à développer 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 chorégraphe comme par exemple la librairie Open CV pour se citer que celle-ci, qui permet de faire du traitement d’images.

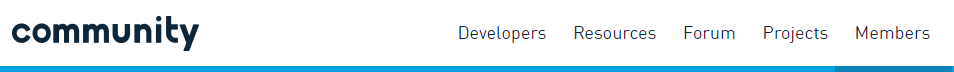
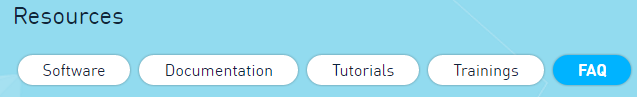
##### Installation

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

* Une connexion à internet
* Au 1.5 GHZ de CPU
* 512 MB de RAM disponible
* Le composant 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.

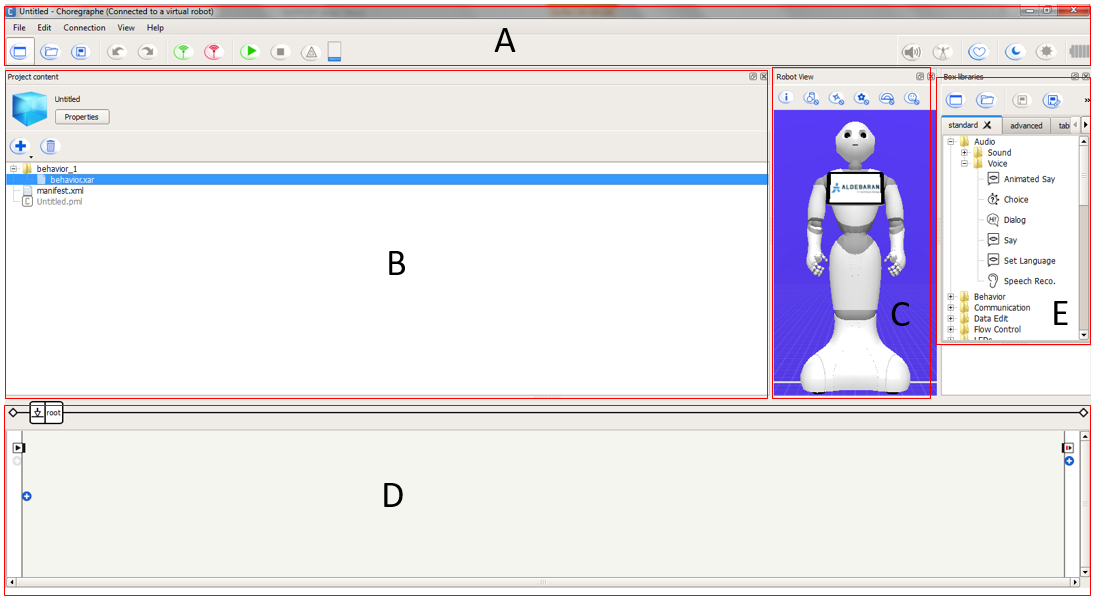
* Rendez-vous à l’adresse suivante : <https://community.ald.softbankrobotics.com/>
* Cliquez sur **Resources** dans la barre de menu à droite de l’écran et ensuite sur **Software** dans la liste des ressources qui s’affichent
* Connectez-vous ou créer un compte si vous n’en avez pas
* Dans la liste des programmes, allez sur **chorégraphe 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.

##### Description

Chorégraphe 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.

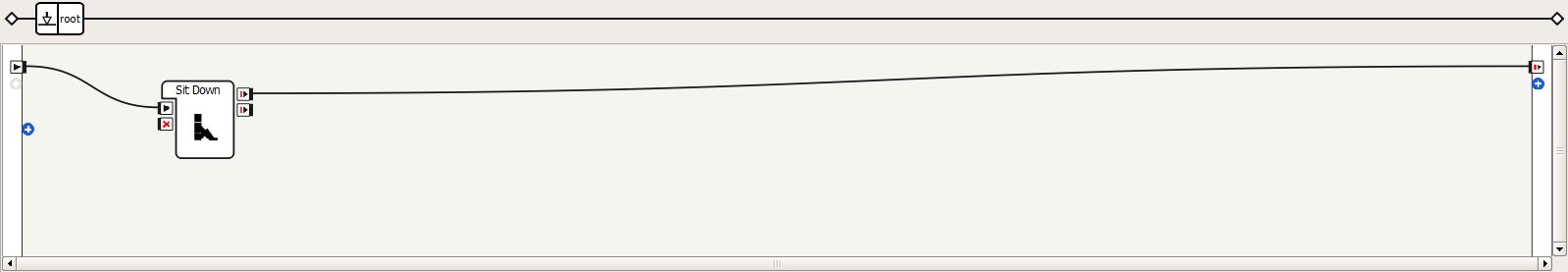
Voici ce à quoi ressemble l’interface de Chorégraphe une fois que vous l’avez ouverte.



* **A :** Cette zone est constituée essentiellement de boutons. Parmi 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, Chorégraphe 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 logiciel. Alors, en cliquant sur ce bouton de connexion, le menu déroulant suivant s’affiche. En cliquant sur **Connect to,** cette fenêtre s’affiche et vous pouvez sélectionner 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 connexion 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 survenues.

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

*  : Il permet de sélectionner les fenêtres à afficher dans l’interface. Par exemple, si vous voulez afficher la fenêtre des logs, il vous suffit de cliquer sur ce bouton puis sélectionner **Log viewer.**
* **B :** Cette zone est réservée 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 :
* Sélectionner une boîte dans la liste des librairies
* La glisser dans cette fenêtre
* Connecter l’entrer « onStart » du robot à celle de la boîte puis la sortie « onStopped » de la boîte à celle du robot 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 réparties 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 créées.

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 | Procédure |
| Play sound | Amener Pepper à jouer une musique. | A l’aide de ce bouton ajouter un fichier wave dans le behavior puis sélectionner le box .Editer ses paramètres et dans le nom du fichier, choisir le fichier wave du behavior. Jouer |
| Record sound | Enregistrer un son extérieur 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 accompagné 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 debout 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 dialogue. | 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 exécutés. |

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 correspondant est : |
| Send email | Permet d’envoyer un email depuis le robot. | Il faut pour cela que le robot ait une adresse email. Le box correspondant est : |
| Remote control | Permet de faire les commandes à distance. Pepper commande un ordinateur ou alors, commandé pepper depuis un autre appareil. | Les box à utiliser sont : |

1. Tests des animations

|  |  |  |
| --- | --- | --- |
| blink | Ce 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 intermittence 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 définir la couleur des leds des yeux | . Utiliser |
| Set leds | Définir l’intensité lumineuse d’un groupe de leds. | Utiliser |
| Set single led | Définir l’intensité lumineuse d’un led en particulier. | Utiliser |

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 s’il est dans le noir. | Utiliser |
| Learn Face | Permet au robot de mémoriser des visages. Ses yeux deviendront verts en cas de succès et rouge sinon. | Ok |
| Record vidéo | Permet d’enregistrer des vidéos à l’aide d’une des caméras du robot. Le temps d’enregistrement est édité dans les paramètres. | Utiliser |
| Take picture | Permet de prendre une photo à l’aide d’une des caméras du robot. | Utiliser |
| Unlearn faces | Permet de supprimer tous les visages de la db. Utiliser |  |
| Vision reco | Permet de reconnaitre les objets, 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 2 : Les frameworks Naoqi et Qi

Le robot Pepper tourne sur le système d’exploitation appelé NAOqi. Il s’agit d’un système Linux

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

Il existe plusieurs manières de développer les applications pour les robots Pepper et NAO en utilisant des langages et plateformes différentes

## Annexe 3 : Prise et traitement d’image par Pepper

## Annexe 4 : Module ALFaceDetection

## Annexe 5: Module ALPeoplePerception

## Annexe 6 : Module ALTextToSpeech

ALTextToSpeech est un module qui permet à Pepper de parler. Le procédé est le suivant :

## Annexe 7: Module ALAnimatedSpeech

## Annexe 8: Module ALMotion

## Code-Annexe

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

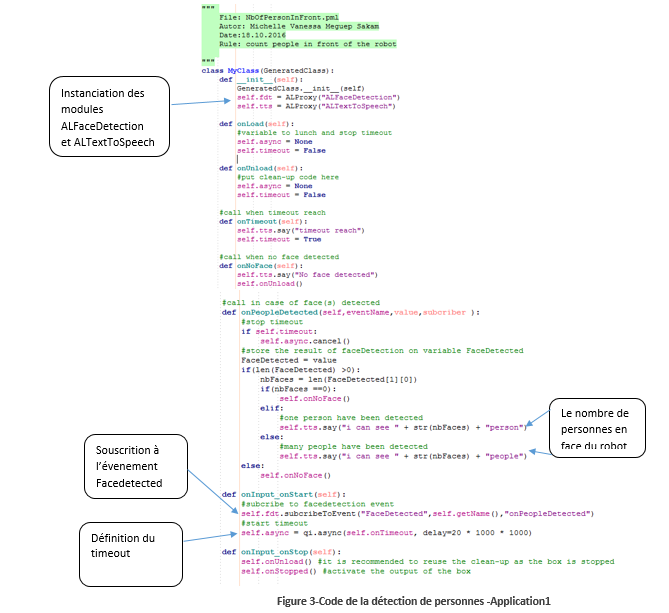
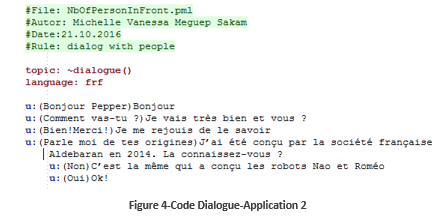


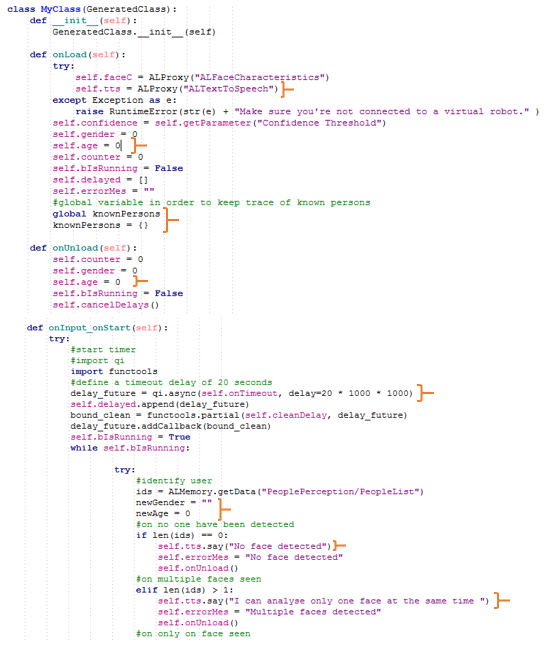
Figure 3 : Pepper donne le nombre de personnes en face

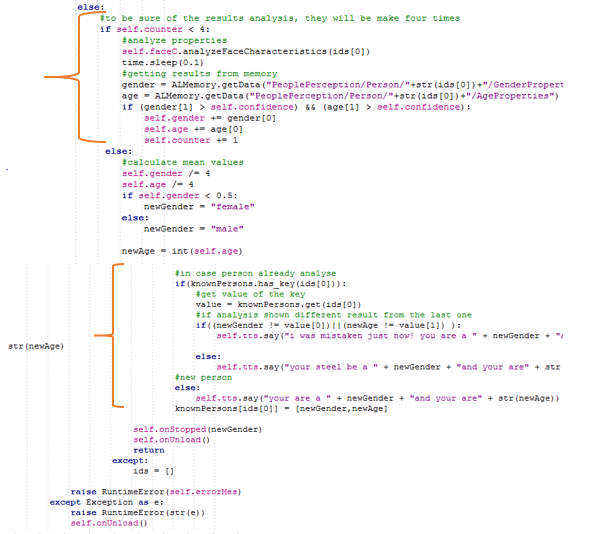
## Code de l’application 2 : Pepper dialogue

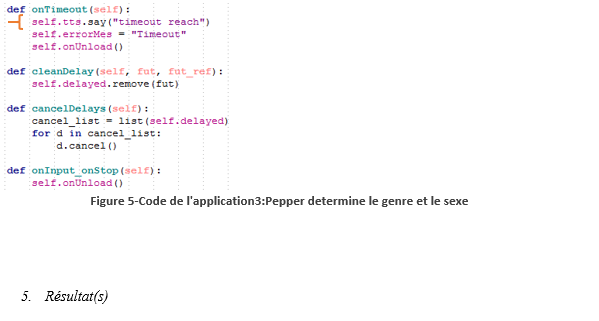


## Code de l’application 3 : Pepper détermine le genre et le sexe de ses interlocuteurs

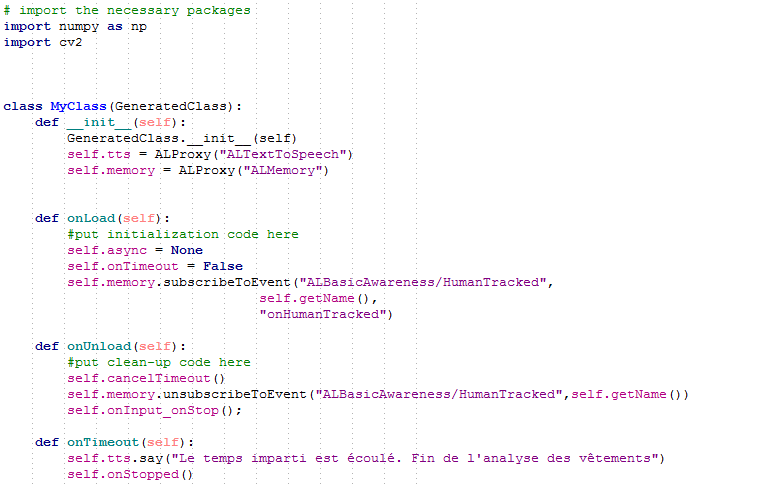
Pour cette application, il a été utilisé une boîte existante de Chorégraphe et le code y a été ajouté afin de correspondre aux éléments du scénario. La boîte de base utilisée est *GetGender* qui permet de donner le genre d’une personne. Le texte ajouté est marqué par les accolades et ci-dessous.

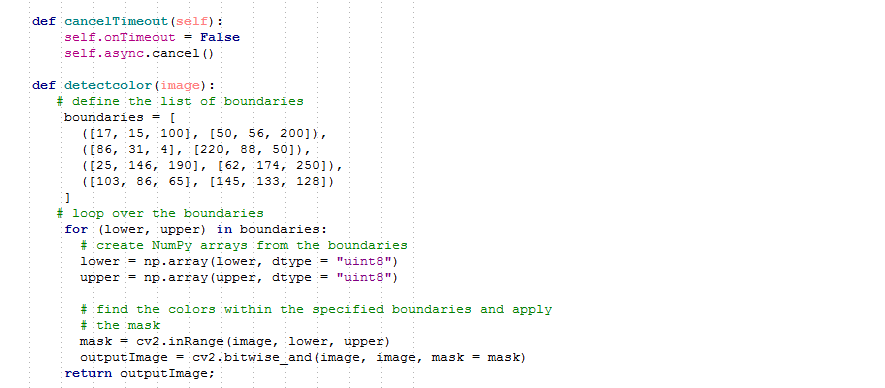


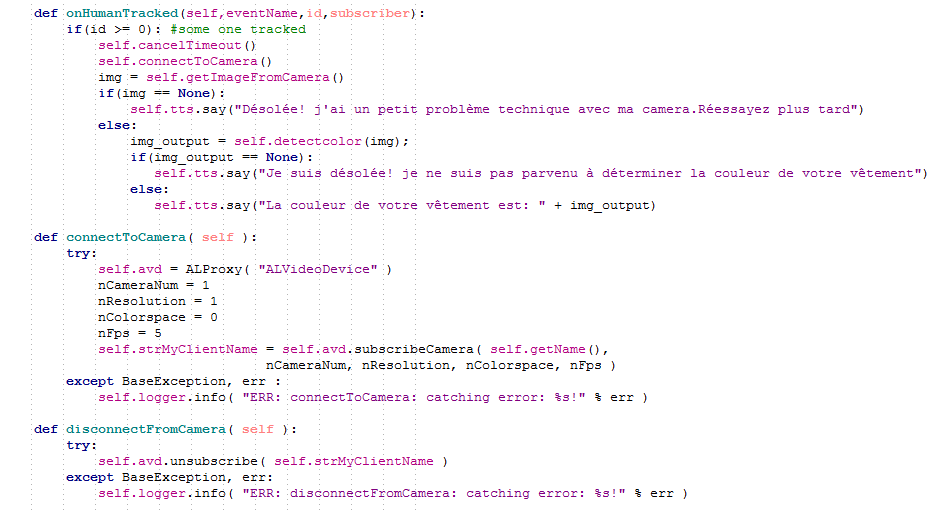


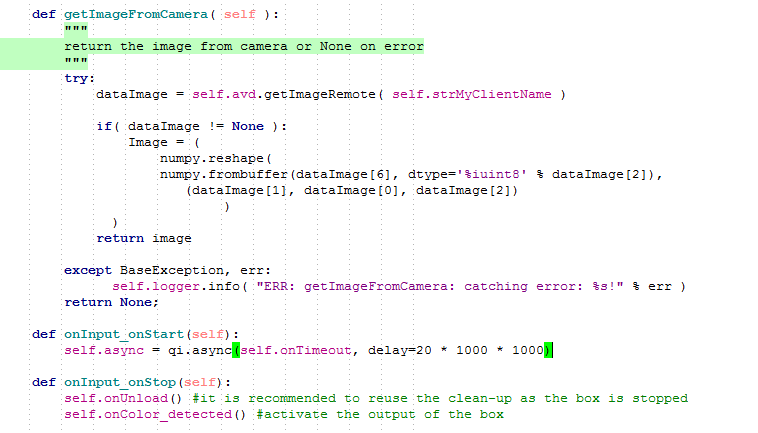


## Code de l’application 4 : Pepper détermine la couleur du vêtement avec openCV



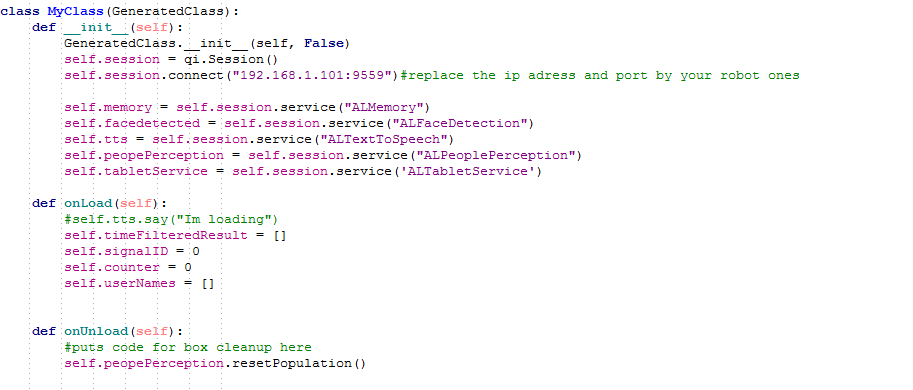


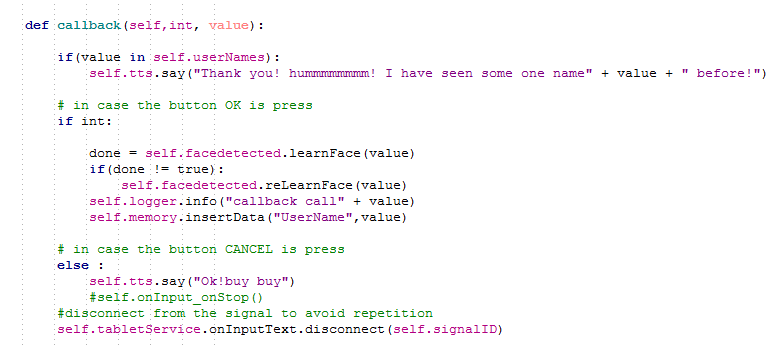


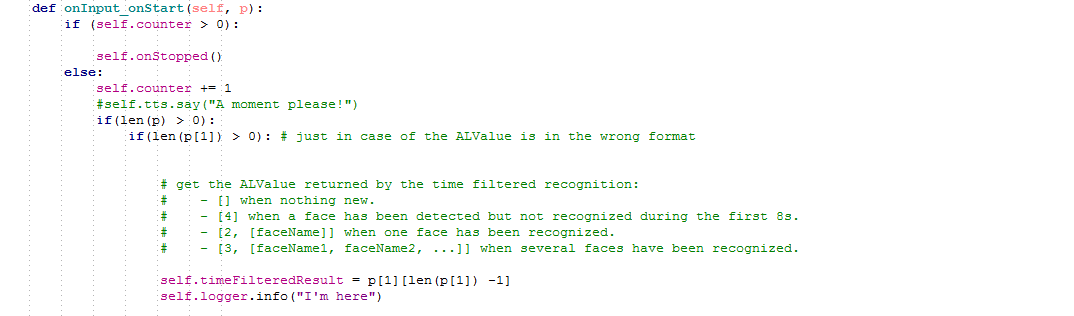


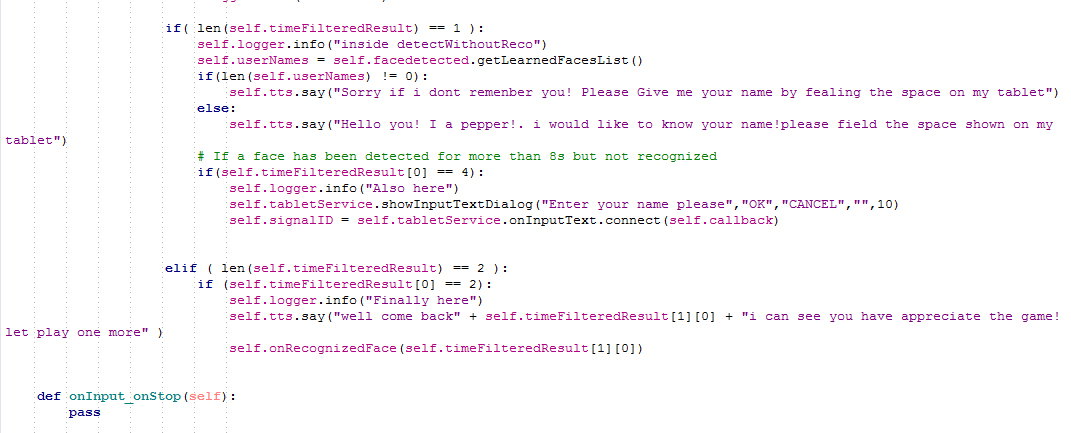
## Code de l’application 5 : Pepper apprend et reconnaît des visages











1. Bachelor, diplôme de couronnement de 3 années d’études [↑](#footnote-ref-1)
2. HEIG-VD, établissement scolaire situé dans le canton de Vaud-Suisse offrant des formations de Bachelor et Master(diplôme de couronnement de 2 années d’études à la suite d’un Bachelor) dans les domaines de l’ingénierie et de Gestion. [↑](#footnote-ref-2)
3. Softbank Robotics, société de robotique spécialisée dans le domaine de la robotique humanoïde, issue du rachat de l’entreprise française Aldebaran Robotics par le groupe Japonais Softbank [↑](#footnote-ref-3)
4. Portes ouvertes de la HEIG-VD, manifestation annuelle organisée par la HEIG-VD et dstinée au grand public, entreprises, futur-e-s étudiant-e-s où sont présentés les filières d’enseignement ainsi que des travaux de recherche menés au sein de l’établissement [↑](#footnote-ref-4)
5. Domaine de la robotique, domaine orienté dans la recherche et l’implémentation des techniques permettant la conception des robots [↑](#footnote-ref-5)
6. Robot, dispositif mécanique, électronique et informatique conçu pour imiter et réproduire les actions de l’hommes [↑](#footnote-ref-6)
7. Humanoïdes : robots dont l’apparence physique rappelle celle de l’homme [↑](#footnote-ref-7)
8. Yverdon-les-bains, ville Suisse du canton de Vaud [↑](#footnote-ref-8)
9. IICT, institut de recherche de la HEIG-VD qui couvre les domaines en informatique et télécommunications. [↑](#footnote-ref-9)