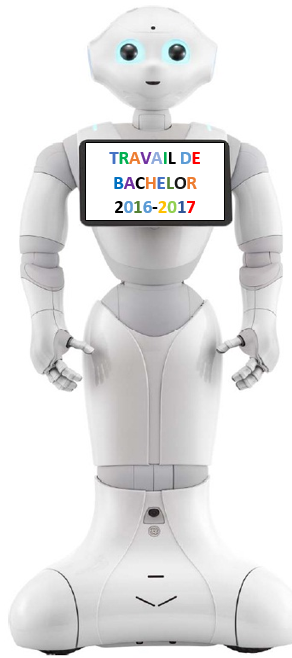
 

ROBOT HUMANOÏDE POUR L’ACCUEIL ET LE DIVERTISSEMENT

30 MARS 2017





**AUTEUR :**

**Michelle Vanessa MEGUEP**

**SUPERVISEUR :**

**Andres PEREZ-URIBE**

**Expert :**

# Remerciements…

# Cahier des charges

Le but de ce travail est de programmer les robots humanoïde Nao et Pepper afin d’interagir avec le public. Il s’agira spécifiquement de penser des scénarios où le robot accueil et amuse le public aux moyens 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 des robots
* Une deuxième phase où l’étudiante programme de nouveaux comportements dans les robots 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.

Table des matières

[Remerciements… 1](#_Toc471790606)

[Cahier des charges 2](#_Toc471790607)

[Résumé 4](#_Toc471790608)

[Introduction 5](#_Toc471790609)

[Chapitre 1 : Apprentissage 6](#_Toc471790610)

[1.1. Présentation des robots 6](#_Toc471790611)

[1.1.1. NAO 6](#_Toc471790612)

[1.1.1.2. Caractéristiques techniques 6](#_Toc471790613)

[1.1.2. Pepper 6](#_Toc471790614)

[*1.1.2.1.* *Fonction* 6](#_Toc471790615)

[1.1.2.2. Charactéristiques techniques 6](#_Toc471790616)

[1.1.3. Comparaison NAO et Pepper 6](#_Toc471790619)

[1.2. Technologies 7](#_Toc471790620)

# Résumé

# Introduction

Depuis la création des premiers robots dans les années 70, le domaine de la robotique ne cesse de connaître des innovations, surtout avec l’évolution sans cesse croissante de l’électronique et de l’informatique. Aujourd'hui, il n’y a rien de plus normal que de trouver un robot relayant l’homme dans presque tous les domaines, aussi variés soient-ils. L’une des branches de la robotique qui nous intéresse est celle des robots humanoïdes. Conçus à l’image de l’homme, leurs concepteurs ont un seul but : les amener à reproduire à la perfection les gestes et les comportements de l’homme. C’est ce à quoi s’adonne l’entreprise française Softbank Robotics autrefois appelée Aldébaran Robotics. Il s’agit d’une société de robotique considérée comme l’un des leaders mondiaux dans le domaine de la robotique humanoïde, qui a déjà créé plusieurs robots à grand succès dont NAO, Pepper et Roméo avec des objectifs un peu différents les uns des autres. Nao est plus orienté vers la programmation, la recherche et l’enseignement, pendant que Roméo est destiné à l’aide aux personnes et Pepper au relationnel. Dans ce projet, réalisé dans le cadre de notre travail de diplôme pour le cycle de Bachelor en ingénierie logiciel à la HEIG-VD(Haute École d’ingénierie et de gestion du canton de Vaud), nous allons nous intéresser aux robots Nao et Pepper. Notre travail consistera à explorer les capacités de ces deux robots pour ensuite les utiliser pour créer des programmes d’interaction homme-machine.

# Chapitre 1 : Apprentissage

## Présentation des robots

### NAO

#### Fonction

NAO est un robot de la classe des robots humanoïdes dévéloppé en 2006 par l’entreprise française Softbank Robotics dont le but est d’aider dans l’enseignement et la recherche. Depuis sa création, cinq de ses versions sont disponibles ; ceci du aux besoins des laboratoires et des autres utilisateurs.

### Caractéristiques techniques

Nao est un robot à l’apparence humaine faisant 58 cm pour un poids de 4.8 kg.Il dispose de quatre microphones pour la reconnaissance vocale et la localisation de la source sonore, deux hauts parleurs pour la synthèse vocale, deux cameras HD de 1280X960 pixels et un capteur sur la tête pour la localisation et la reconnaissance de visages et d’objets, d’un accéléromètre sur trois axes, de deux sonars, 8 capteurs de pression, 2 bumpers(capteurs de contact) pour sa locomotion. Il a une autonomie de 90 mins et un dégrè de liberté de 14 à 25.

### Pepper

### *Fonction*

Pepper est un robot humanoïde lancé en 2014 par l’entreprise Softbank Robotics. Il a été conçu dans le but de percevoir et reproduire certaines émotions humaines. Depuis juin 2014, date de son premier déploiement dans les boutiques Softbank, Pepper a rejoint de nombreux magasins, entreprises, aéroports, foyers etc. On le retrouve dans plusieurs lieux où il est généralement chargé d’accueillir, informer et distraire les visiteurs. Dans les foyers, Pepper fait partir comme membre intégrante et sert de compagnon.

### Charactéristiques techniques



Pepper est un robot a l’apparence humaine avec ses 1m21 de hauteur pour un poids de 28 kg.Il porte sur son torse une tablette à écran tactile de 10.1 pouces, se déplace grâce à trois roues omnidirectionnelles et est muni sur la tête de 4 microphones qui lui permettent d’entendre. Egalement muni d’une camera 3D et de deux autres HD qui lui permettent de voir et d’identifier les mouvements. Il possède également 4 microphones pour parler et de nombreux capteurs qui lui permettent de bien gérer sa motion et éviter les obstacles. Avec sa batérie de Lithium, il a une grande autonomie de près de 12h.

### Comparaison NAO et Pepper

Que se soit NAO ou Pepper, ces deux robots sont capables de faire les actions suivantes :

* Entendre
* Parler avec la possibilité de choisir la langue d’expression
* Bouger
* Danser
* Détecter un visage ou une voix
* Reconnaître un visage ou une voix
* Communiquer
* Rester en équilibre débout ou assis
* Éviter des obstacles
* analyser
* Se connecter à internet par Ethernet ou Wi-Fi
* Ils sont aussi open source

Mais les points qui les distinguent sont les suivants :

1. 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 un aspect plus proche de l’homme et rend l’interaction homme-machine plus réaliste.
2. La mobilité : Pepper dispose pour sa mobilité de trois roues omnidirectionnelles contrairement à Nao qui tient sur deux jambes.Il est vrai que ceci rapproche plus NAO de l’homme mais avec ses roues, Pepper peut se déplacer plus vite, a plus d’équilibre et un meilleur appuis sur le sol.
3. 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 porter ou à déplacer.
4. La vision : Pepper possède une camera 3D qui peut produire une image d’une résolution allant jusqu’à 320x240 pixels pour 20 trames par seconde et deux autres cameras 2D d’une résolution maximale de 2560X1080 pixels pour 5 trames par séconde Contrairement à NAO qui ne possède que deux cameras HD d’une résolution de 1280X960 pixels pour 30 trames par séconde. La résolution des caméras de Pepper permet de mieux simuler la vision humaine.
5. Les émotions : Pepper est capable d’analyser le comportement ainsi que la voix de ses interlocuteurs pour décéler des émotions ; il peut ainsi modifier son comportement en conséquence.
6. La communication : Pepper dispose en plus sur son torse d’une tablette tactile lui permettant d’interagir avec l’homme. Il peut ainsi faire de l’affichage ou proposer des ménus à sélectionner.
7. Autonomie : Pepper a une baterie d’une autonomie de 12h contrairement à NAO dont la baterie ne peut faire que 90 mins hors charge.

Le point essentiel et qui est d’ailleurs ce qui nous intéresse le plus est que ces robots de l’entreprise Softbank Robotics sont tous programmables.

## Developpement

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

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

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

#### Les différents modes de programmation

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

#### Les langages de programmation

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

* + 1. Les plateformes
       1. Choregraphe

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

En effet, choregraphe a été conçu pour aider les programmeurs sans expérience en robotique à developper des applications sans trop s’encombrer de code. Il permet alors à l’aide des boîtes préprogrammés de créer des circuits et exécuter des comportements. Il permet en même temps de se connecter à un robot réel ou virtuel afin de tester les comportements programmés. Cependant, il est possible pour des programmeurs plus expérimentés, 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 l’API NAOqi. Il est également possible de connecter des librairies externes à choregraphe comme par exemple la librairie OpenCV pour ce citer que celle-ci, qui permet de faire du traitement d’images.



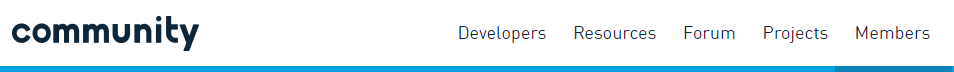
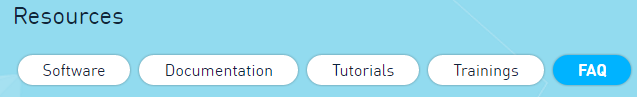
##### Installation

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

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

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

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

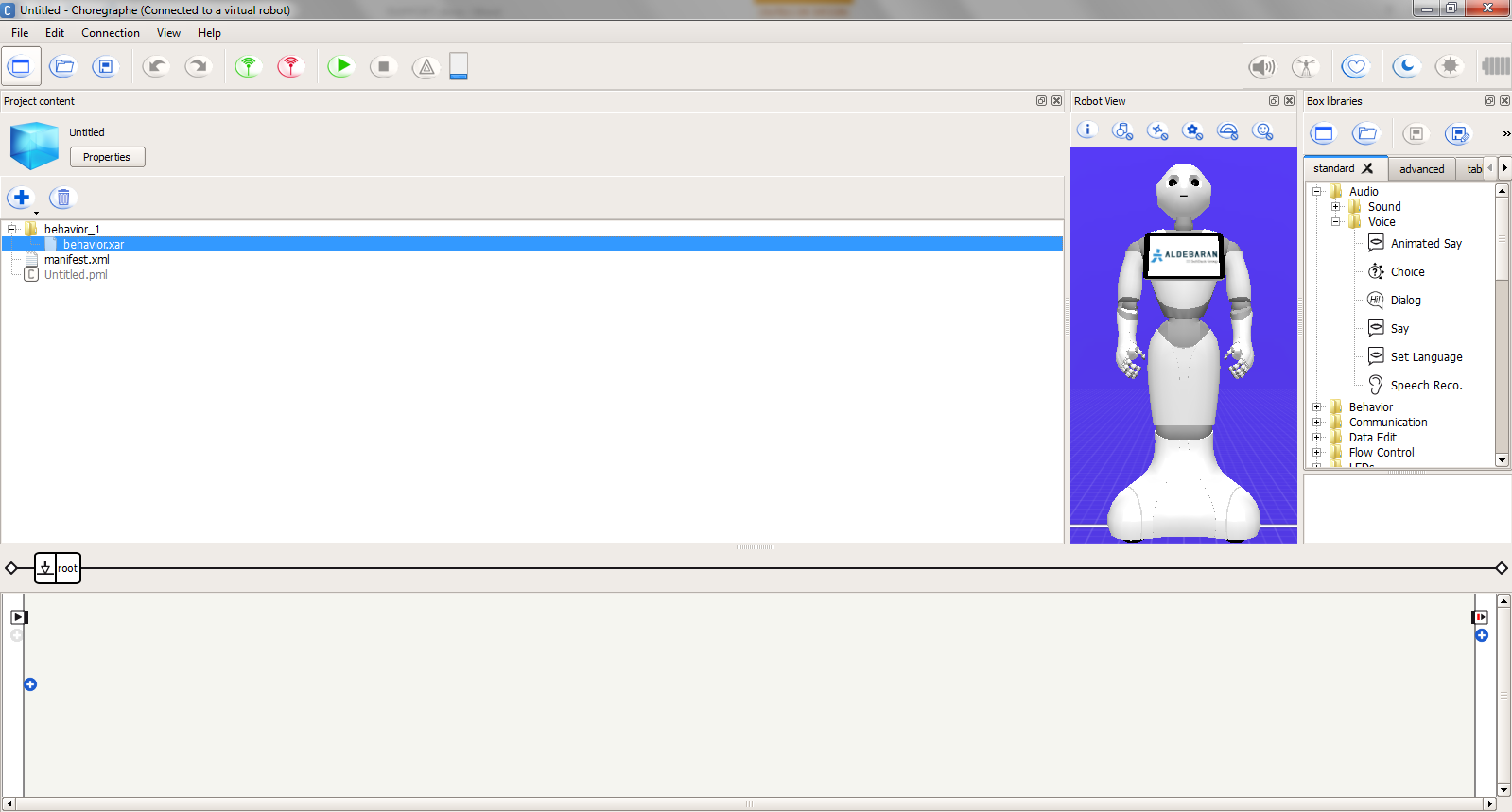
* Rendez vous à l’adresse suivante : <https://community.ald.softbankrobotics.com/>
* Cliquez sur **Resources** dans la barre de menu à droite de l’écran et ensuite sur **Software** dans la liste des resources qui s’affichent
* Connectez vous ou créer un compte si vous n’en avez pas
* Dans la liste des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

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

##### Apprendre à utiliser Choregraphe

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

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



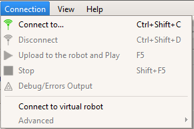
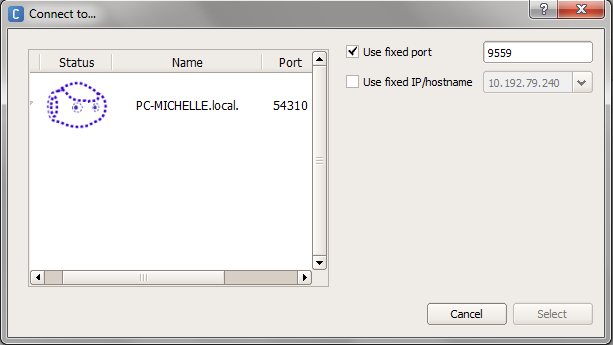
D andrea@cotza.me

E andrea@cotza.me

Candrea@cotza.me

B andrea@cotza.me

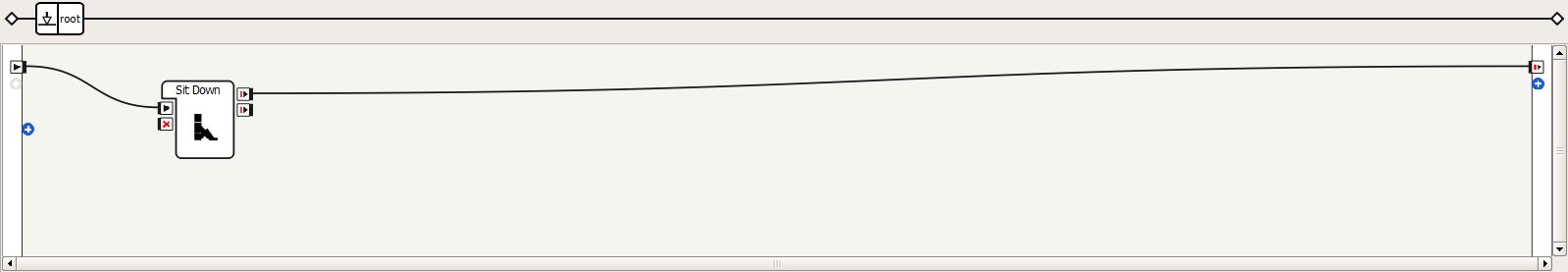
A andrea@cotza.me

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

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

* : Il permet de selectionner les fenêtre à afficher dans l’interface.Par exemple, si vous voulez afficher la fenêtre des logs, il vous suffit de cliquer sur ce bouton puis selectionner **Log viewer.**



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

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

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

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

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

1. Tests Audio

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

1. Tests de la voix(voice test)

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

1. Tests du comportement(behavior)

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

1. Tests de la communication (communication)

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

1. Tests des animations

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

1. Tests de la gestuelle(motion)

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

1. Tests de la vue

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

Légende : textes en noirs : testés avec le simulateur

Textes en vert : a programmer pour voir le comportement puis tester avec le robot

Textes en rouge : A tester avec le robot.

* + - 1. Webot

###### Programmer Pepper

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

* 1. NAOqi

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

1. Fonctionnement

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

1. NAOqi - documentation

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

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

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

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

//décrire le code de cette boxe.

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

1. Scénario 2 : Pepper exécute un dialogue

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

Topic 1 : Accueil

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

Topic 2 : Renseignement

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

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

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

### Utiliser le SDK python pour la programmation

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

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

Exemple :



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

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



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