

APP ROBOTIQUE DE SERVICE

BENHIMA Mehdi
DINDELLI Dylan-Anthony
JENNY Camille
LÉGLISE Cloé
MISON Jules
RUIZ Florian
SANGOUARD Marine

Introduction.....	2
Objectifs du semestre.....	2
Travail et réalisation.....	3
Algorithme de plus court chemin.....	3
Déplacement de Pepper.....	4
Serveur.....	6
Interaction orale avec Nao.....	7
Site internet.....	8
Organisation.....	9
Conclusion.....	9
Annexes.....	10
Site web.....	10
Organisation.....	13

Introduction

Dans le cadre de notre semestre 8 dans la filière “Systèmes Numériques - Instrumentation”, nous poursuivons notre apprentissage par projet centré sur la robotique de service. Notre projet consiste à utiliser les robots Pepper et NAO pour réaliser une visite guidée de Polytech. Le processus de notre projet comprend la conception jusqu'à la réalisation et la validation de celui-ci.

Le premier semestre nous a permis de définir nos objectifs du projet et de nous familiariser avec notre environnement, c'est-à-dire découvrir les robots Pepper et Nao, le logiciel Choregraphe, et faire des recherches sur des serveurs, principalement. Nous avons également pu nous habituer au travail en groupe au sein de notre équipe. Notre second semestre de travail était centré sur le cahier des charges, que nous avons donc défini plus précisément, en le découplant par niveaux de difficulté. Ces deux premiers semestres de travail nous ont permis de commencer l'implémentation du code de notre projet, de réaliser de nombreux tests et d'avancer significativement dans le développement de notre travail. A la fin du semestre 7 de notre cycle ingénieur, nous étions parvenus à remplir le premier niveau de notre cahier des charges.

L'objectif de ce troisième semestre de travail sur ce projet était de terminer l'implémentation de notre code en remplissant le plus de niveaux de notre cahier des charges possible. Les principaux axes de travail étaient le déplacement de Pepper, et le lien entre les différentes parties du projet. Ce semestre est également censé être notre dernier semestre de travail axé sur l'implémentation du projet en elle-même.

Objectifs du semestre

Comme il a été dit en introduction de ce rapport, l'objectif de ce semestre était plus ou moins de terminer l'implémentation de notre projet. Pour ce faire, nous nous sommes prioritairement concentrés sur le déplacement de Pepper et le serveur. Si nous avions, au semestre précédent, correctement codé un algorithme de suivi de mur pour le déplacement du robot, le but était à présent d'utiliser notre algorithme de suivi de mur afin de le faire se déplacer suivant une trajectoire définie. Pour ce qui est du serveur, il nous fallait terminer de faire le lien entre toutes les parties du projet, par exemple utiliser la tablette et l'algorithme de plus court chemin pour afficher un plan avec trajectoire sur la tablette et faire suivre ladite trajectoire à Pepper.

Nous avons également avancé sur d'autres axes un peu moins cruciaux pour le fonctionnement basique du projet. Ainsi, nous avons travaillé sur l'interaction orale avec le robot Nao, et commencé à mettre à jour le site internet du projet, qui servira principalement à l'aspect mise en valeur de notre travail du semestre prochain.

Travail et réalisation

Algorithme de plus court chemin

Au semestre précédent, nous avons créé un algorithme qui permet à partir d'un point de départ et d'arrivée, de donner la liste des coordonnées des points par lesquels il faut passer pour aller au point d'arrivée à partir du point de départ. Nous sommes partis de cette liste pour en faire une suite d'instructions compréhensibles par le robot Pepper.

Pour que le programme Chorégraphe fonctionne, il faut trois informations pour le mode de fonctionnement, la distance à parcourir et le mur à suivre. On crée donc un programme qui va dans un premier temps détecter tous les croisements dans la liste de points. Pour cela, on compare une liste de tous les croisements de Polytech avec notre liste de points et on note quels sont les croisements dans cette liste. Ensuite, on retire tous les points inutiles (sauf si c'est un croisement, par exemple si on a une liste de points [[12,6],[15,6],[18,6]], le point 15,6 n'est pas nécessaire, car c'est sur la même ligne). Ensuite, on crée des sections. Les sections sont une liste de deux points qui sont sur la même ligne. À partir de ces sections, on va pouvoir calculer la distance que le robot doit parcourir. Une fois ceci fait, on ajoute une information à la section qui sera le mur à suivre. On veut que le robot suive le mur qui correspond au sens du prochain virage à effectuer. Dans le cas où il n'y aurait aucun virage à faire, le robot suit le mur de droite. On choisit le mode en fonction de la situation, par défaut s'il y a un virage à prendre, on est en mode 1, si on est en fin de parcours et qu'il n'y a plus de virage, on sera en mode 0. Un troisième mode existe pour les cas particuliers où le robot a besoin d'aller tout droit sans suivre de mur. Il ne nous reste plus qu'à écrire dans un fichier texte ces trois informations. Une fois ce code fini, on le fusionne avec le plus court chemin.

Cependant, Polytech est un bâtiment qui n'est pas régulier. C'est pourquoi on a dû introduire des cas particuliers. En effet, le robot n'est pas en capacité de détecter les barrières sur les passerelles. De plus, Polytech a des portes coupe-feu qui coupent les couloirs. C'est donc difficile pour Pepper de contourner cet obstacle.

Nous avons rencontré plusieurs problèmes lors de la conception de ce système. Pour commencer, nos points d'intérêt sont placés à la main sur le plan, ce qui crée une erreur humaine si on se trompe dans les coordonnées du point. Ce problème a été récurrent, car dans le développement, nous avons eu à déplacer des points pour les faire mieux coller aux emplacements des salles ou pour diverses autres raisons. Ce déplacement de point a pour conséquence que le programme ne trouve plus de solution au chemin, car pour lui le point n'existe pas ou n'est pas accessible.

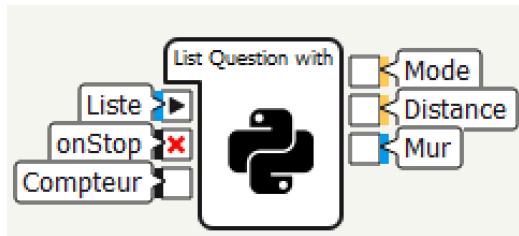
Nous avons rencontré différentes difficultés durant la mise en place de cet algorithme. Dans nos tests de l'algorithme et de la transmission de données au robot, on a rencontré des problèmes avec des instructions erronées à cause de cas particuliers omniprésents dans Polytech. On a donc dû réaliser beaucoup de tests dans différentes situations pour faire apparaître ces problèmes. Un autre problème rencontré, est un problème d'encodage. En effet, le programme qui écrit la donnée pour Pepper et l'envoie au

robot n'étaient pas encodés tous les deux de la même façon, et donc, il y avait des problèmes de messages incompréhensibles. On a donc dû spécifier l'encodage et le désencodage en UTF-8 pour éviter ces problèmes.

Déplacement de Pepper

Pour que Pepper puisse connaître le chemin par lequel elle doit passer, le serveur envoie une liste d'instructions qu'a générée l'algorithme de plus court chemin. On crée un bloc qui a pour but de récupérer cette liste sous forme de chaîne de caractères et de la décomposer pour les envoyer sur la bonne sortie au bon moment. Comme le robot doit récupérer les informations par triplet en fonction de la portion de chemin qu'il fait, le bloc va envoyer un triplet d'informations à chaque fois qu'il y a un "bang" sur l'entrée compteur. Grâce à cela, le robot va récupérer les bonnes informations pour sa portion de chemin. Une fois fini, il envoie un "bang" au bloc qui lui donne les nouvelles instructions.

Ce bloc a deux entrées et un stop. La première prend en entrée la liste de toutes les informations pour aller au bout du chemin sélectionné. La deuxième est une entrée bang qui sert à envoyer le prochain triplet pour la prochaine étape du chemin. En sortie, il renvoie 3 informations, le mode de fonctionnement (0,1,2), la distance en mètres à effectuer et le mur que pepper doit suivre (droit ou gauche).



Le but de notre travail est que le robot Pepper se déplace dans les couloirs et jusqu'à destination choisie par l'utilisateur à travers sa tablette. Il faut donc s'assurer de la bonne connexion entre le robot et le serveur externe afin que le robot puisse recevoir les informations de déplacement. En effet, le serveur traite l'algorithme du plus court chemin et envoie au robot une suite d'instructions telles que le choix du mode de déplacement (en suivie de mur ou pas), la distance à parcourir et le mur suivi (droit ou gauche).

Pour traiter différents cas de figure possibles, nous avons réfléchi à différents choix de déplacement. Nous les avons regroupés sous 3 keyframes :

- Le déplacement par suivi de mur pour une distance donnée :

Nous demandons au robot de suivre un mur (droit ou gauche) sur une distance donnée. Le capteur du côté du mur suivi est tenu à une certaine distance fixée. La distance parcourue par le robot est gardée en mémoire et est comparée en

temps réel avec la distance demandée. Lorsque cette distance est atteinte, le prochain déplacement peut être effectué.

- Le déplacement par détection d'intersection :

Nous demandons au robot de suivre un mur (droit ou gauche) mais de s'arrêter lorsqu'il détecte une intersection. Le mur à suivre est choisi en fonction de la direction vers laquelle le robot devra aller après son prochain virage. Ceci va permettre la détection de l'intersection car le capteur du côté du "trou" détectera une distance trop grande par rapport au mur. Nous avons fixé cette limite pour qu'il détecte l'intersection. Le robot effectuera donc une rotation de 90° du côté de la direction où il veut aller. Pour finir le virage et que le robot puisse reprendre son chemin, nous le faisons avancer un peu pour qu'il se retrouve entre les deux murs du couloir qu'il veut emprunter, et donc puisse reprendre son chemin.

- Le déplacement sans suivi de mur pour une distance donnée :

Nous demandons au robot de se déplacer d'une certaine distance sans prendre en compte où il se situe par rapport aux couloirs. Comme pour le déplacement par détection d'intersection, le robot effectue ensuite une rotation de 90° du côté de la direction où il veut aller et avance avant de reprendre son chemin. Ce moyen n'est pas utilisé fréquemment lors de nos déplacements, il est programmé pour des cas particulier où des petits déplacements ou des déplacements dans les coursives (où les barrières ne sont pas détectables comme un mur par le robot) doivent être effectués.

Ces déplacements sont effectués les uns à la suite des autres en traitant le fichier texte envoyé par le serveur après son calcul du plus court chemin. Le chemin est considéré comme terminé lorsqu'il n'y a plus d'indications à suivre du fichier texte.

Pour représenter ces trois modes, nous avons trois blocs disposés dans trois keyframes distinctes. Ces keyframes sont appelées par un bloc switch case qui agit en fonction du numéro de mode envoyé par le serveur. Ces différents types de déplacement sont essentiels pour pouvoir s'adapter au mieux à différents types de situation.

Nous avons donc deux blocs qui sont basés sur le suivi de mur développé au semestre précédent. Le suivi de mur a été optimisé pour l'utilisation que nous en faisons, et nous avons maintenant quelques paramètres importants. La distance du mur demandée est importante pour indiquer à Pepper à quelle distance du mur se placer, et quel est l'intervalle d'erreur autorisé pour cette distance avant de la corriger. C'est important car un intervalle trop faible poussera Pepper à se rapprocher du mur jusqu'à en être trop proche pour pouvoir avancer, et une trop grande amènera Pepper jusqu'à l'autre côté du couloir, ce qui n'est pas viable non plus. En plus de cela il est aussi important de fixer la vitesse du déplacement selon l'axe y. Une correction de déplacement en y trop forte fait avancer le robot par à-coups et le fait dévier rapidement. Pour contrer cet effet tout en laissant une correction satisfaisante, nous fixons cette valeur à 1 au maximum. Ensuite, nous paramétrons la correction d'angle maximale que Pepper peut réaliser. Dans le cas du suivi de mur classique, on fixe la vitesse maximale de rotation à 0.8 pour qu'il puisse tourner efficacement, mais sans faire de virage trop violent qui le ferait dériver de sa trajectoire. Pour le suivi jusqu'à une intersection, on fixe la rotation à 0.2 au maximum pour éviter que

Pepper réagisse trop violemment à la vue d'un virage. En effet, Pepper est incapable d'effectuer un virage correct avec un bloc de suivi de mur car elle tourne trop vite, donc nous choisissons de réaliser les virages dans un bloc indépendant. La vue de l'intersection la fera sortir du bloc de déplacement pour entamer son virage.

Une des difficultés majeures rencontrées lors de ce semestre a été la difficulté à fermer nos boucles while qui permettent au robot d'avancer jusqu'au remplissage d'une condition. Pour pallier ce genre de problème, nous avons programmé des fermetures de boucles de sécurité à l'appui long sur un bumper. Cela permet de sortir d'une boucle avec certitude et d'éviter les résidus de boucle tournant en fond sur les processeurs de Pepper.

Un autre aspect compliqué rencontré a été l'architecture générale de Polytech, chaque trajet est très différent car le bâtiment n'est pas complètement uniforme, et programmer Pepper pour être efficace dans des types de virages différents, ou avec différents types de murs ou barrières peut s'avérer très compliqué voire impossible sans alternative dans le cas des barrières. Nous avons donc passé beaucoup de temps à essayer de s'adapter à notre environnement. C'est la raison pour laquelle nous avons prévu plusieurs types de déplacement, mais aussi de trajectoire avec l'algorithme de plus court chemin, pour éviter les endroits dangereux.

Une dernière difficulté que nous avons rencontré est : les variables à partager à tout le programme.

Serveur

Pour la partie serveur, après avoir codé les différents clients nécessaires pour notre projet, le but de ce semestre était de réaliser la connexion entre nos différents clients afin d'établir la liaison entre les différents acteurs de notre projet.

De manière générale, la connexion entre nos différents clients consistait à récupérer l'information de la tablette de Pepper qui va jouer le rôle de l'interface homme-machine de notre projet. Le visiteur va cliquer sur la salle qu'il souhaite visiter sur la tablette, et la tablette envoie l'information reçue par le visiteur au serveur. Le serveur récupère le message de la tablette et le traite en l'envoyant à l'algorithme de plus court chemin qui va traiter le message reçu et renvoyer le bon chemin au serveur. Une fois le message reçu par le serveur, il envoie le bon chemin avec les bonnes instructions au robot Pepper, et envoie les images du plan correspondant au chemin demandé par le visiteur, ainsi qu'un QR code contenant l'image de ce chemin permettant au visiteur de garder le chemin au cours de son déplacement. En plus de ça, pour le robot Nao qui avait comme mission d'accueillir les clients, nous avons aussi codé un client Nao qui permet la connexion avec le serveur, et qui permet d'échanger et de récupérer des messages avec le robot Pepper et informer les visiteurs sur la position et la disponibilité de Pepper.

Pour cela, la première tâche à effectuer était d'établir la connexion entre la tablette qui contiendra notre site et le serveur, de telle sorte qu'un clic sur la salle génère l'envoi d'une commande au serveur contenant le numéro de la salle cible. Pour ce faire, nous avons créé un code php pour chacune de nos salles qui permet la création d'un client Web

associé à chaque salle qui permettra la liaison avec le serveur, ensuite nous avons transformé le clic généré par le client à une chaîne de caractère envoyée au robot.

L'étape suivante était de traiter le message reçu par le serveur par l'algorithme de plus court chemin. Pour ce faire, il fallait établir tout d'abord la liaison entre notre serveur codé en Java et notre algorithme de plus court chemin codé en Python. Le transfert des messages entre ces 2 clients nous a présenté un grand problème d'encodage qui nous a empêché de traiter l'information reçue par le serveur. Pour résoudre ce problème, nous avons créé un client serveur local pour le serveur codé en Python qui permet de jouer le rôle du serveur local pour nos clients codés en Python et qui corrigera l'erreur d'encodage avec notre serveur principal.

Pour rendre la connexion avec l'algorithme de plus court chemin plus fluide, il était modifié de telle sorte qu'il prend en entrée un fichier texte "Commande.txt" contenant la salle de départ du robot et la salle d'arrivée, et renvoie en sortie un fichier texte "DonneeToPepper.txt" contenant le chemin avec les bonnes instructions au robot Pepper. En récupérant la salle ciblée par le visiteur, le serveur modifie le fichier texte "Commande.txt" et modifie la salle de destination avec la nouvelle salle reçue. Une fois le chemin effectué, la salle de destination devient la salle de départ et la salle de destination sera la nouvelle salle demandée par le client. Après la réception de la commande, le serveur exécute le client du plus court chemin. Une fois lancé et exécuté, le fichier "DonneeToPepper.txt" est modifié par le nouveau chemin du robot, et le fichier texte du chemin est envoyé au serveur avec l'image et le QR code associés au bon chemin de déplacement souhaité.

Lorsque le serveur reçoit les données après le traitement de l'algorithme du plus court chemin, les instructions du chemin sont envoyées au client Pepper, et les images du plan et du QR code sont envoyées à notre client Web afin qu'elles soient affichées sur la tablette.

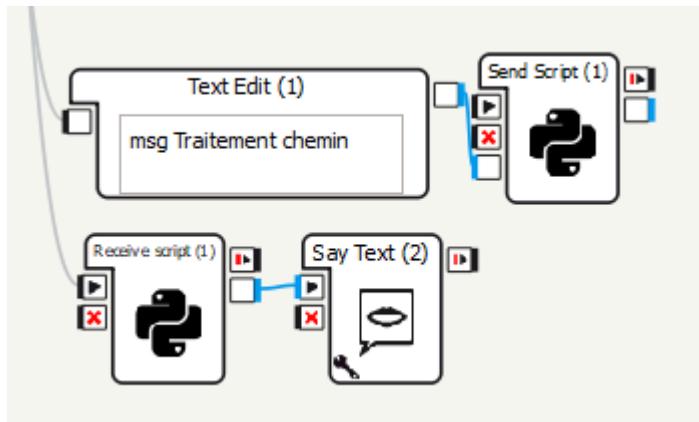
La dernière fonctionnalité du serveur consistait à établir la connexion entre le robot Nao qui s'occupera de l'accueil des visiteurs, et le robot Pepper qui va les guider. Pour rendre cette communication plus fluide, nous avons codé quelques fonctionnalités qui permettent à Nao de récupérer le dernier chemin effectué par le robot, d'où il pourra savoir sa position actuelle, en plus d'une fonctionnalité qui permet à Nao de connaître l'état de Pepper, s'il est connecté, en chemin ou déconnecté via le serveur.

Interaction orale avec Nao

Nao est en mesure de présenter les diverses spécialités de l'école (Annecy uniquement). Pour ce faire, nous avons créé un fichier texte avec toutes les questions que Nao doit et peut comprendre. Ce fichier contient également chaque réponse associée et même une interaction (lire un fichier audio par exemple). Notre but est que le visiteur puisse récolter plus d'informations sur l'école, mais également patienter si Pepper n'est pas encore disponible. Nous avons besoin du serveur pour connaître l'état de Pepper. Nous en avons profité pour ajouter diverses informations détenues par le serveur comme le dernier trajet effectué par Pepper.

Pour ce faire, nous avons repris les blocs de communications codés pour pepper, les 2 robots utilisant le même langage il n'y a eu aucune adaptation nécessaire. La structure de communication avec le serveur se fait de la façon suivante :

- Une impulsion de type bang écrit le message que nous souhaitons envoyer au serveur. En parallèle, l'impulsion lance le bloc de réception du serveur.
- Ensuite le message est envoyé et nous attendons une réponse du serveur pour que Nao parle.



Cela nous permet de demander le dernier trajet effectué ainsi que l'état actuel de Pepper au travers de Nao. De plus, nous avons étoffé les dialogues de Nao comme par exemple des présentations des différentes filières.

Avec ce processus, il est facile de rajouter des nouveaux dialogues à Nao en modifiant seulement le fichier texte. On peut aussi ajouter des actions qui devront par contre être ajoutées dans le programme Chorégraphe. Cependant, si une action est ajoutée une fois, elle peut être utilisée par plusieurs lignes de dialogue. Il suffirait de mettre sur la ligne d'interaction le nom de celui-ci.

Nous n'avons pas rencontré de difficulté majeure hormis quelques difficultés de connexion à Nao et une instabilité de la connexion sur les dernières séances.

Site internet

Dans le cadre de la mise en valeur de notre projet, nous avions à réaliser lors de notre premier semestre de travail un site web avec toutes les informations sur notre projet : notre équipe, nos objectifs, notre avancée... Si nous n'y avons pas touché depuis, nous avons dû nous y remettre ce semestre afin de le mettre à jour. Le but était premièrement d'avoir des informations à jour avec notre avancée au fur et à mesure du temps, et deuxièmement de restructurer notre travail, notamment car nous souhaitions éviter d'utiliser une base de données pour des raisons de facilité d'accès de la part des visiteurs. Cet outil

sera particulièrement exploité lors du semestre prochain, dont le but est de mettre en valeur notre travail.

En plus de la page d'accueil, qui présente sommairement le principe de l'APP et l'objectif de notre équipe, notre site est constitué de quatre pages différentes. La page *Objectifs* reprend notre cahier des charges par niveaux. La page *Avancement* détaille notre travail selon trois axes principaux : le déplacement de Pepper, l'interaction homme-machine, et le lien entre les différentes parties du projet. Cette page ne reprend pas nos avancées de manière chronologique, mais a pour but d'expliquer clairement ce que nous avons mis en place. C'est là que nous posterons notre vidéo de démonstration. La page *Notre équipe* présente notre équipe, avec des photos et son évolution au fil du temps. Enfin, la page *Archives* recueille tous nos compte-rendus de fin de séance et de semestre, ainsi qu'une introduction et un bilan pour chaque semestre, en plus d'un résumé de travail par séance.

Vous trouverez des copies d'écran de ce site ainsi que son lien en annexe.

Organisation

Après deux semestres passés à travailler ensemble, l'équipe a pris ses marques. Pour ce semestre, Camille Jenny était notre responsable de projet. Nous avons globalement conservé notre fonctionnement général, avec certaines personnes travaillant sur les axes principaux (Mehdi sur le serveur, Jules et Dylan-Anthony sur l'algorithme de plus court chemin et l'interaction orale, Marine et Camille sur le déplacement de Pepper), et d'autres personnes gravitant entre les groupes (Florian et Cloé, qui s'est ensuite concentrée sur le site web). La différence majeure ce semestre-ci est que la communication a pris une place encore plus importante qu'avant puisqu'il fallait faire le lien entre toutes les différentes parties. Les différents groupes qui s'étaient naturellement formés au tout début du projet se sont peu à peu réunis au fur et à mesure que nous avancions dans le projet et que les différents axes de travail devenaient de plus en plus confondus.

Pour ce qui est de l'organisation générale de l'équipe, nous avons continué d'utiliser les outils à notre disposition, c'est-à-dire le git, mais aussi les sites Trello et Gantt (la copie du Gantt créé pour ce semestre est à retrouver en annexe, ainsi que le lien du git).

Conclusion

Lors de ce semestre, nous avons quasiment terminé l'implémentation de notre projet. L'interaction orale de Nao et le déplacement de Pepper selon une destination choisie sur sa tablette sont fonctionnels, à quelques dysfonctionnements près. Nous avons finalement dévié de notre cahier des charges par niveaux, puisque nous n'avons pas défini de boucle à faire faire à Pepper, ce qui était un objectif de notre niveau 2, mais avons passé énormément de temps sur le déplacement du niveau 4, qui est maintenant terminé et presque opérationnel. Ainsi, notre projet ressemble actuellement beaucoup à ce que nous avions imaginé lors de notre toute première séance d'APP au semestre six, et bien qu'il ne soit pas encore tout à fait en état de faire une visite guidée en autonomie lors d'une journée portes ouvertes, il ne restera en théorie pas grand-chose à modifier avant que cet objectif

soit atteint. De même, le site web, qui est plus un objectif du semestre prochain, est complètement terminé et il ne restera plus qu'à le tenir à jour au fil du temps.

Ce semestre marque la fin de l'APP pour une grande partie de notre équipe, et si les difficultés qui se sont présentées à nous et la fin précipitée des séances passées à essayer de terminer rapidement le projet avant la date limite sont encore frais dans nos esprits, nous pouvons globalement être fiers du travail que nous avons accompli jusqu'à présent. Notre projet est aujourd'hui quasiment fonctionnel, et l'équipe a su explorer différents domaines de la robotique et informatique.

Le prochain semestre sera axé sur la mise en valeur du projet. Si l'équipe restante décide de mettre en place un projet complètement fonctionnel, il s'agira principalement de régler les dysfonctionnements, l'implémentation en elle-même étant terminée.

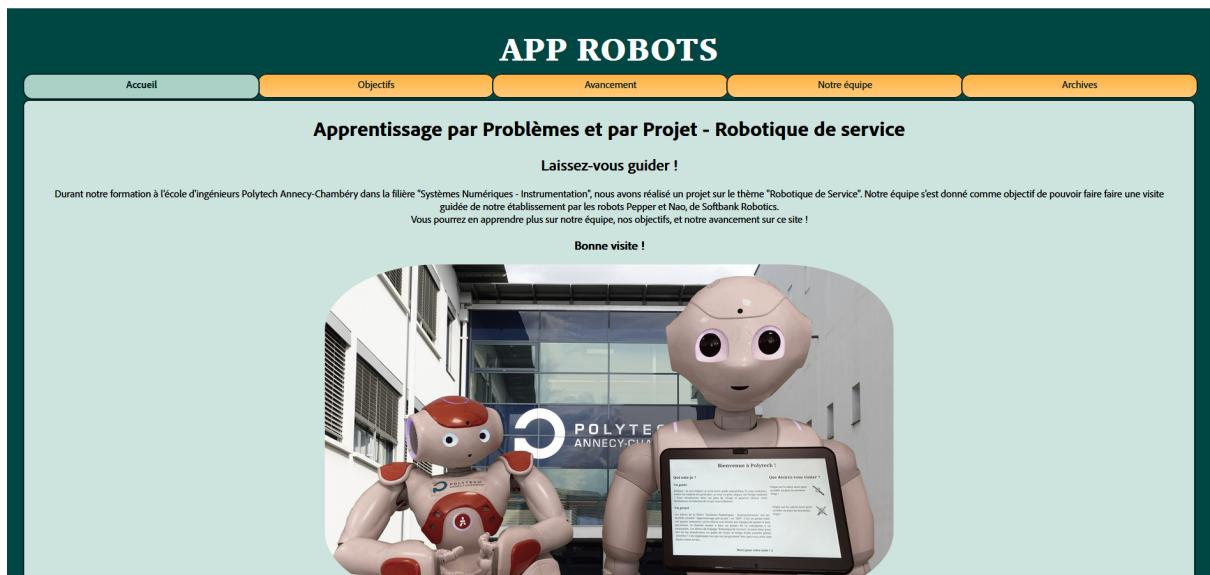
Annexes

Site web

Lien du site (le fait que le site soit à jour au moment où vous regarderez ne dépend pas de nous, nous avons prévenu le professeur responsable que nous avions terminé pour ce semestre et ne savons pas quand il le mettra en ligne) :

https://app-polytech.fr/sites_web_app/SITES_APP/APP2024/ROBOTS/index.html

Voici quelques captures d'écran montrant le visuel général du site :



Haut de la page d'accueil du site

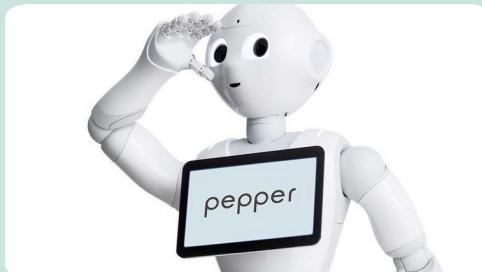
L'Apprentissage par Problèmes et par Projet est un module obligatoire dans la formation SNI de Polytech Annecy-Chambéry. Il s'agit un long projet étalé sur deux ans, par équipes. Chaque élève choisit la thématique qu'il préfère, et c'est ainsi que notre équipe Robotique de Service a été créée. Nous devons travailler sur ce projet sur un total de quatre semestres, respectivement axés sur la découverte, le cahier des charges, l'implémentation et la mise en valeur de notre projet. C'est à nous, les élèves, de déterminer nos objectifs, en rapport avec notre thème.

Notre thème étant la robotique de service, et ayant à notre disposition des robots Pepper et Nao de Softbank Robotics, nous avons décidé de partir sur du guidage, notre but final étant de faire faire une visite guidée de notre établissement aux robots lors d'une journée portes ouvertes, avec Nao à l'accueil et Pepper comme guide. Ce projet nous permet d'explorer de nombreux aspects de notre formation que nous ne voyons pas forcément en cours et qui sont spécifiques à chaque projet. Nous aurons ainsi l'occasion d'approfondir nos connaissances sur les serveurs, le code de sites et applications web, ou encore le travail avec des robots en utilisant le logiciel Choregraphe.

Ce projet est également l'occasion de travailler avec une équipe nombreuse (nous étions 8 au début) que nous n'avions pas choisie. La méthode de notation est particulière, avec une grande importance donnée à des entretiens individuels en plus d'une soutenance et d'un rapport de fin de semestre qui viennent régulièrement résumer notre travail. L'école met peu à peu en place un système de notation par compétences, qui nous force à prendre du recul sur ce que nous sommes capables de faire et notre évolution grâce aux différents projets - et particulièrement l'APP. C'est donc un module très important dans notre cycle ingénieurs.

Sur ce site, vous trouverez un résumé de notre cahier des charges sur la page Objectifs, une explication de notre implémentation sur la page Avancement, une présentation de notre équipe et de son évolution au cours du temps sur la page Notre équipe, et enfin tous nos documents obligatoires, c'est-à-dire les compte-rendus de fin de séances et les rapports de fin de semestres, accompagnés de résumés de notre avancée rédigés au fur et à mesure de notre travail, sur la page Archives.

Robot PEPPER



Robot NAO



Pepper est un robot humanoïde. Nous souhaitons en faire un guide mobile permettant aux visiteurs de découvrir notre bâtiment. Nous utilisons sa tablette tactile comme interface avec un plan des différents étages permettant aux utilisateurs de sélectionner leur destination, avec comme objectif final que Pepper puisse ainsi amener les visiteurs là où ils le souhaitent... Ne nous屏障ez pas !

Nao est un tout petit robot. Pour des raisons de batterie et de stabilité, nous n'avons pas prévu de l'utiliser comme guide. Son rôle sera plutôt d'accueillir les visiteurs. Il sera possible d'interagir avec lui pour lui demander des renseignements sur l'école, ou encore lui demander comment trouver Pepper afin d'essayer la visite guidée. Et qui sait, s'il est de bonne humeur, il vous chantera une belle chanson !

Suite de la page d'accueil du site

APP ROBOTS

Accueil

Objectifs

Avancement

Notre équipe

Archives

Objectifs

Comme indiqué en page d'accueil, notre thème est la robotique de service, et nous avons choisi d'utiliser les robots à notre disposition (Pepper et Nao, de Softbank Robotics) afin d'organiser une visite guidée de notre établissement faite par ces robots. Pour cela, nous avons établi un cahier des charges détaillé. L'objectif principal de notre second semestre de travail était la rédaction de ce cahier des charges. Nous avons décidé d'opter pour un cahier des charges par niveau de difficulté, le premier niveau comprenant des fonctionnalités que nous avons jugées faciles à implémenter, et le dernier niveau (le niveau 4) comprenant un idéal à atteindre, dans le cas où nous aurions plus de temps. Les différents niveaux du cahier des charges sont détaillés ci-dessous, et vous trouverez la version rédigée de notre cahier des charges en annexe de notre rapport de fin de semestre / sur notre page Archives. La page Avancement décrit notre avancement sur ce projet et des explications un peu plus détaillées sur les différents concepts que vous croiserez dans le cahier des charges.

Niveau 1 du cahier des charges (fonctionnalités essentielles, atteint en fin de deuxième semestre)

▼ Déplacement de Pepper

Pas de déplacement prévu pour ce niveau, simple affichage du plan sur la tablette.

- Interaction humains - robots
- Tablette tactile
- Serveur
- Sécurité

Niveau 2 du cahier des charges (fonctionnalités qu'on aimerait mettre en place avant la fin)

▼ Déplacement de Pepper

Faire faire un chemin prédéfini en boucle à Pepper.

- Interaction humains - robots
- Tablette tactile
- Serveur
- Sécurité

Niveau 3 du cahier des charges (fonctionnalités qu'on espère mettre en place avant la fin)

▼ Déplacement de Pepper

Affichage du plan et éventuellement indications à l'oral sur le chemin à suivre. Essayé avorté d'un système où Pepper doit suivre un humain en le guidant comme un GPS.

Page objectifs du site reprenant notre cahier des charges

APP ROBOTS

Accueil
Objectifs
Avancement
Notre équipe
Archives

Suivi du projet

Sur cette page seront expliqués les aspects techniques de l'avancement de notre projet. Dans la page des archives, vous trouverez tous nos compte-rendus de séances ainsi que nos rapports de fin de semestres. L'objectif de cette page est d'expliquer clairement ce que nous avons fait, sans noyer notre lecteur dans les détails ! Les concepts seront donc simplifiés pour la plupart, et vous pourrez trouver plus d'informations dans nos compte-rendus sur la page dédiée aux archives, et particulièrement dans les rapports de fin de semestres.

Durant ce projet, nous avons utilisé les robots Pepper et Nao de Softbank Robotics. Nous les avons programmés à l'aide du logiciel Choregraphe, qui permet de faire faire des actions simples aux robots par programmation graphique, mais également d'approfondir la programmation en utilisant le langage Python Aldébaran. Les boîtes d'action du logiciel sont programmées dans ce langage, et nous en avons modifié et écrit quelques-unes pour notre utilisation personnelle.

- Déplacement de Pepper

Pour rappel, notre but ultime est que Pepper soit capable d'effectuer une visite guidée en autonomie. Depuis notre deuxième semestre de travail, nous disposons de deux robots Pepper, que nous avons l'intention de positionner respectivement au premier et deuxième étage de Polytech (le robot se déplace sur roulettes et ne peut pas franchir les escaliers). Pour cela, nous avons besoin de mettre une méthode de déplacement sûre et efficace, et de faire en sorte que le robot sache quel chemin emprunter pour aller d'un endroit à un autre. Nous avons donc mis en place des algorithmes de suivi de mur pour le déplacement, et un algorithme de plus court chemin pour la trajectoire générale.

- Algorithme de suivi de mur
- Algorithme de plus court chemin

- Interaction homme - machine

Pour que le projet fonctionne, il ne suffit pas que Pepper puisse se déplacer, il faut également que le robot et le visiteur puissent communiquer. Il a été dit plus haut et les deux robots Pepper seront positionnés pour faire visiter les étages 1 et 2 de notre bâtiment Polytech Annecy. Comme mentionné sur notre page d'accueil, le robot Nao n'est pas assez stable ni rapide pour faire office de guide, et sa batterie ne tient pas très longtemps. Il a donc été décidé de le positionner dans le hall d'entrée, au rez-de-chaussée, afin d'accueillir le visiteur et éventuellement les rediriger vers les Pepper pour la visite des étages. Nous avons donc mis en place deux méthodes de communication : une communication orale, destinée principalement à Nao et accessible à tous les robots ; et l'utilisation de la tablette tactile intégrée sur le robot Pepper, qui servira uniquement pour les visites.

- L'interaction orale
- La tablette tactile

- Lien entre les différentes parties du projet

Toutes les fonctionnalités mentionnées plus haut ne font pas le projet en lui-même si elles ne peuvent pas communiquer entre elles. Nous avons mis en place un système de serveur multi-thread pour faire le lien entre les différentes parties du projet. De même, les

Début de la page avancement du site

• Interaction homme - machine

Pour que le projet fonctionne, il ne suffit pas que Pepper puisse se déplacer, il faut également que le robot et le visiteur puissent communiquer. Il a été dit plus haut et les deux robots Pepper seront positionnés pour faire visiter les étages 1 et 2 de notre bâtiment Polytech Annecy. Comme mentionné sur notre page d'accueil, le robot Nao n'est pas assez stable ni rapide pour faire office de guide, et sa batterie ne tient pas très longtemps. Il a donc été décidé de le positionner dans le hall d'entrée, au rez-de-chaussée, afin d'accueillir le visiteur et éventuellement les rediriger vers les Pepper pour la visite des étages. Nous avons donc mis en place deux méthodes de communication : une communication orale, destinée principalement à Nao et accessible à tous les robots ; et l'utilisation de la tablette tactile intégrée sur le robot Pepper, qui servira uniquement pour les visites.

- L'interaction orale
- La tablette tactile

Les robots Pepper ont une tablette tactile intégrée et indépendante du robot. Nous avons codé une application web à afficher sur cette tablette, avec une page d'accueil permettant de naviguer sur les plans des différents étages. Pour chaque étage, un plan s'affiche avec une légende expliquant les différents types de salle. L'utilisateur peut ainsi sélectionner la salle qu'il souhaite visiter, et en lire une courte présentation. S'il le souhaite, il peut ainsi afficher le plan avec la trajectoire générée par l'algorithme de plus court chemin, et même scanner un QR code lui permettant d'obtenir ce plan sur son téléphone portable. C'est en utilisant la tablette tactile que l'utilisateur peut communiquer avec le robot et lui demander de le guider à un endroit en particulier. Cette application web a été codée uniquement en HTML/CSS et Javascript, et utilise également du PHP pour la partie serveur à retrouver plus bas. Des copies d'écran montrant cette application sont présentées en rapport de fin de semestre 7 à retrouver dans la page dédiée aux archives.

Voici des images de la page d'accueil de l'application web, de la page du premier étage avec la présentation de notre salle de travail (passez la souris sur l'image pour l'agrandir) :

• Lien entre les différentes parties du projet

Toutes les fonctionnalités mentionnées plus haut ne font pas le projet en lui-même si elles ne peuvent pas communiquer entre elles. Nous avons mis en place un système de serveur multi-thread pour faire le lien entre les différentes parties du projet. De même, les différents membres du groupe ont besoin de communiquer entre eux pour se mettre d'accord (par exemple sur la manière de communiquer les instructions de l'algorithme de plus court chemin à Pepper), et nous avons utilisé différents outils pour cela.

- Serveur
- Communication au sein de l'équipe

Bas de la page avancement du site, avec pied de page et exemple de partie détaillée

12

APP ROBOTS

Accueil Objectifs Avancement **Notre équipe** Archives

L'équipe

Au début de notre formation, nous avons tous choisi quel projet était celui qui nous intéressait le plus. De cette manière, l'équipe Robotique de Service est née - et les autres aussi. Au fil du temps, cette équipe a connu de nombreux changements, avec les semestres d'échange à l'étranger et les redoublements. Voici son histoire.

L'équipe du premier semestre

Il se trouve que notre premier semestre de travail sur l'APP correspond en réalité au sixième semestre de notre cycle ingénieurs, durant lequel Florian était notre chef de projet. À ce moment-là, notre équipe était composée de 8 membres :



Début de la page de notre site présentant l'équipe de travail

Archives

Bienvenue sur la page des archives ! Vous trouverez ici un résumé de chaque semestre et de chaque séance, ainsi que les liens vers les PDF des archives à proprement parler, c'est-à-dire les différents comptes-rendus et rapports. Cette page sera mise à jour au fur et à mesure de l'avancement du projet.

- **Semestre 6**

Le semestre 6 de notre cycle ingénieur marque le premier semestre de travail sur notre APP. C'est donc un semestre essentiellement lié à la découverte des différents outils à notre disposition afin de pouvoir apprendre à les maîtriser. Si ce travail de découverte est essentiel pour avancer dans notre travail, le gros de notre projet, c'est-à-dire l'établissement d'un cahier des charges et une progression plus réfléchie du travail, n'interviendra qu'aux semestres suivants, quand nous saurons déjà nous servir correctement de notre environnement de travail. Notre apprentissage sur ce semestre se concentrera essentiellement sur la maîtrise et la compréhension du fonctionnement de Choregraphe, le logiciel codé en python pour utiliser les robots Pepper et Nao ; l'apprentissage et la maîtrise du langage JavaScript pour créer une application web ; et enfin la compréhension et la maîtrise du principe de serveur web dans le but d'en créer un lors des semestres suivant (son but étant de lier entre eux tous les différents éléments de notre projet).

Pour ce premier semestre de travail, notre équipe était constituée de BENHIMA Mehdi, JENNY Camille, LEGLISE Cloé, MISON Jules, ROY Nicolas, SALH Hamza, SANGOUARD Marine, et notre chef de projet RUIZ Florian.

 - ▶ Première séance - 09/02/2022
 - ▶ Deuxième séance - 02/03/2022
 - ▶ Troisième séance - 16/03/2022
 - ▶ Quatrième séance - 29/03/2022
 - ▶ Cinquième séance - 13/04/2022
 - ▶ Sixième séance - 26/03/2022
 - ▶ Septième séance - 04/05/2022
 - ▶ Huitième séance - 10/03/2022
 - ▶ Bilan du semestre
- **Semestre 7**

Le semestre 7 de notre cycle ingénieur est le second semestre de travail lié à notre projet. L'objectif principal de ce semestre est la mise en place d'un cahier des charges détaillé. Maintenant que nous avons pris en main les différents outils à notre disposition et que nous comprenons un peu mieux les difficultés face à nous, nous allons rédiger un cahier des charges par niveaux et commencerons l'implémentation de celui-ci.

Pour ce deuxième semestre de travail, l'équipe était constituée de BENHIMA Mehdi, DINELLI Dylan-Anthony, JENNY Camille, MISON Jules, RUIZ Florian, SANGOUARD Marine, et notre chef de projet LEGLISE Cloé.

 - ▶ Première séance - 14/09/2022
 - ▶ Deuxième séance - 29/09/2022

Partie de la page des archives avec la totalité de nos documents livrables

Organisation

Lien du git :

https://github.com/PolytechAnnecy-RobotiqueDeService/APP_RobotiqueDeService

Gantt du semestre :

