DomotIcApp

Diane Dewez, Iman Akabi,
Enora Lucas, Meven Moser, Corentin Chatellier

Encadrants : François Pasteau, Daniel Guillard

27 mai 2016

1 Introduction

Les nouvelles technologies permettent des progrès dans de nombreux domaines. DomotIcApp, l'application sur laquelle nous travaillons, permet d'améliorer grandement la vie des personnes en fauteuil roulant. Réalisé en lien avec Ergovie, une entreprise rennaise vendant et réparant des fauteuils roulants et plus récemment développant des solutions de contrôle de domotique, le but de ce projet est de proposer une application simple permettant de contrôler l'environnement d'une personne en fauteuil roulant (exemples : allumer la lumière, ouvrir une porte, fermer les stores. . .).

La tablette sur laquelle l'application sera installée sera fixée directement sur le fauteuil de l'utilisateur. Ce fauteuil est notamment équipé de plusieurs capteurs, capables de donner des informations quant à l'état du fauteuil (vitesse, position du joystick de direction, etc.).

Par ailleurs, ce projet s'inscrit dans la continuité d'un projet réalisé auparavant par un autre groupe d'étudiants de l'INSA de Rennes. Ce projet portait sur le même sujet, certaines composantes de l'application sont donc déjà complètement ou partiellement implémentées.

Nous détaillerons dans un premier temps le cahier des charges qui nous a été fourni. Les spécifications qui s'y trouvent nous ont fait choisir certaines solutions techniques plutôt que d'autres. Nous avons découpé ce projet en plusieurs phases de travail qui nous ont mené jusqu'à l'application finale que nous présenterons à la fin de ce rapport.

2 Cahier des charges

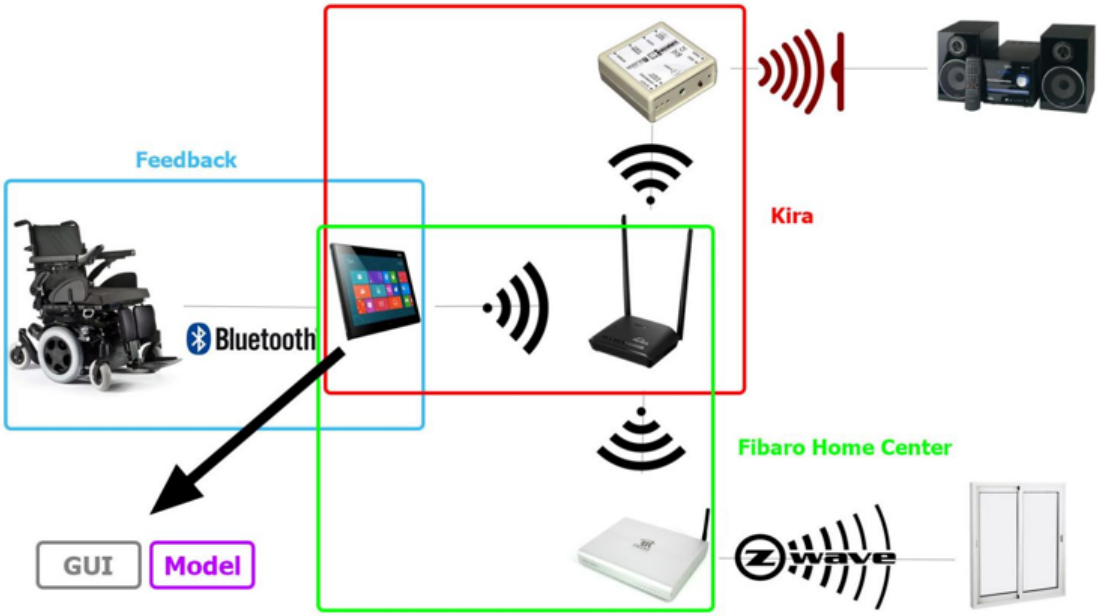


FIGURE 1 – Schéma général de l'application

2.1 Communication entre la tablette et la Kira128

L'application doit pouvoir communiquer avec la Kira128 pour effectuer des actions sur les différents composants infrarouges présents dans la maison (télévision, chaine hifi...). Les codes IR utilisés pour contrôler ces appareils doivent être enregistrés dans la Kira128 au préalable.

2.2 Communication entre la tablette et la box Fibaro Home Center

Un autre type de box doit également pouvoir être utilisé : la Fibaro Home Center, une box capable de contrôler des périphériques via un protocole appelé Z-Wave. Il s'agira la plupart du temps de portes automatisées, de volets, de lampes, etc. L'application doit pouvoir communiquer avec la box Fibaro pour effectuer des actions sur les équipements domotiques de la maison compatibles avec le protocole Z-Wave (ouvrir une porte, fermer les volets...) ainsi que recevoir les informations nécessaires sur l'état des composants du réseau, c'est-à-dire savoir si une lampe est allumée par exemple. Pour la box Fibaro, l'application ne peut pas récupérer les périphériques enregistrés ; ils doivent être connus au préalable par la personne effectuant la configuration de l'application.

2.3 Communication entre la tablette et le fauteuil

L'objectif de cette fonctionnalité est de centraliser les données envoyées par les capteurs installés sur le fauteuil. On pourra ainsi réaliser un tableau de bord permettant de visualiser les informations données par les capteurs installés sur le fauteuil (exemple : vitesse).

2.4 Interface homme-machine (IHM)

L'interface graphique servira de télécommande à l'utilisateur qui pourra ainsi contrôler les équipements de sa maison. Cette interface ainsi qu'une première représentation du modèle de l'application ont déjà été réalisées intégralement en `C#`, lors du projet antérieur à celui-ci. Cependant, les parties concernant le contrôle de la domotique et le lien avec le fauteuil n'avaient pas été implémentées. L'interface actuelle doit donc être rendue fonctionnelle et faire le lien entre l'utilisateur et ces deux parties. De plus, l'application doit pouvoir être portée le plus simplement et rapidement possible sur des systèmes autres que Windows. Le modèle de l'application doit donc être traduit dans un langage permettant le développement sur une plus large gamme de systèmes. Ceci permettra également de désolidariser la partie purement graphique de l'application de sa partie modèle.

L'interface graphique devra être simple d'utilisation. Son ergonomie doit être adaptée aux problématiques du handicap lourd :

- Un minimum de boutons permettant de choisir une pièce de la maison, et l'équipement avec lequel on veut interagir.
- Des couleurs contrastées avec des illustrations simples permettant de bien distinguer les différentes icônes.
- Changement de couleur de l'icône pour indiquer que le clic a été pris en compte (feedback visuel).
- Un bouton spécialement conçu pour changer de page (pas de "défilement par glissé" qui pourrait perturber l'utilisateur).

L'application devra également proposer un système de navigation alternatif, dans le cas où un appui sur un élément précis de l'interface est difficile voire impossible. Ce système de navigation fonctionnera de la manière suivante :
Premièrement, les différents groupes de boutons seront mis en surbrillance successivement. Lorsque le groupe contenant le bouton que l'utilisateur souhaite actionner est mis en surbrillance, l'utilisateur le signalera via un appui sur l'écran de la tablette ou bien via un contacteur relié à la tablette. Deuxièmement, chaque bouton de la zone sélectionnée sera mis en surbrillance, de manière successive. Lorsque le bon bouton est mis en surbrillance, l'utilisateur le signale de la même manière qu'avant. L'action est alors effectuée.

L'application aura deux modes. Un premier mode "Utilisateur" sera dédié à une utilisation courante par la personne en fauteuil. Par défaut, l'application ne contiendra aucune icône. Un second mode "Administrateur" sera utilisé par une personne habilitée pour configurer les différentes fonctionnalités exploitables par l'utilisateur.

L'administrateur doit pouvoir configurer plusieurs aspects de l'application :

- Ajouter des icônes et leur associer une action de manière simple.
- Organiser les icônes selon les pièces de la maison ou selon différentes thématiques (la taille des icônes, leur couleur...).
- Choisir le mode de navigation et gérer la vitesse de défilement pour le second mode.

3 Choix techniques

Les choix techniques qui nous étaient proposés étaient très restreints, dû au fait que ce projet était dans la continuité de celui commencé l'année dernière (voir introduction). Toute l'IHM commencée en `C#` devait être continuée en `C#`. Ce langage est adapté pour les tablettes Windows, support principal de l'application. Le matériel nous était fourni par l'entreprise, nous n'avions donc pas le choix non plus pour celui-ci.

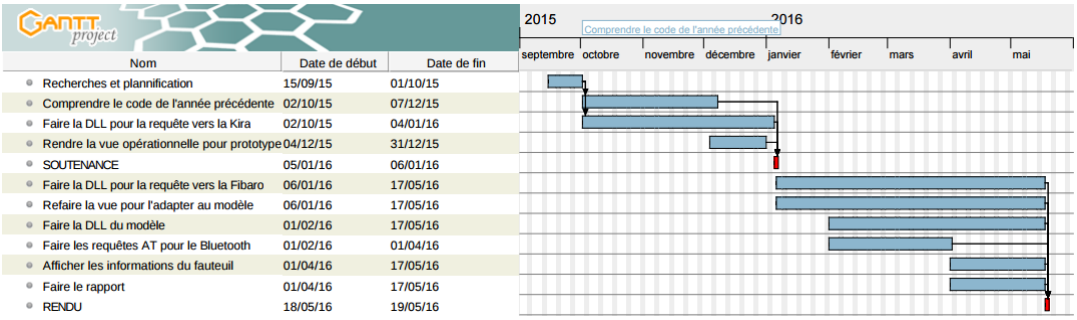
Le premier choix technique auquel nous avons dû réfléchir est celui concernant le langage dans lequel serait traduit le modèle de l'application, puisque celui-ci devait être reprogrammé pour des soucis de compatibilité vers d'autres systèmes d'exploitation (voir cahier des charges, partie IHM). L'application doit pouvoir dans le futur être utilisée sur d'autres plates-formes telles que Android. Il fallait donc un langage compatible avec toutes ces plates-formes, mais aussi en adéquation avec nos connaissances techniques de 3ème année. Trois langages répondaient à ces exigences : le Java, le C et le `C#`. Java avait l'avantage d'être plus simple pour nous car nous l'utilisions depuis la 1ère année, contrairement au C que nous venions d'apprendre. Cependant, pour le challenge et l'expérience que nous apporterait l'apprentissage d'un nouveau langage et aussi pour s'assurer des bonnes performances de notre application tout en conservant une programmation orientée objet, nous avons choisi d'utiliser le `C++` pour traduire le modèle.

Ensuite nous devions déterminer si nous allions utiliser les protocoles UDP ou HTTP pour la Kira. Comme la Fibaro ne fonctionnait elle qu'en HTTP, nous avons choisi d'utiliser le protocole HTTP pour simplifier notre code. Enfin il nous fallait choisir la bibliothèque dans laquelle nous allions effectuer les requêtes HTTP en `C++`. Après quelques recherches, il s'était avéré que HappyHttp était la plus intéressante. Simple à utiliser, elle est aussi complètement embarquée et portable sur n'importe quel système d'exploitation, répondant parfaitement à nos attentes.

Notre constat est que ces choix ont été plutôt bons puisque nous les avons suivis jusqu'à la fin du projet. Le `C++` n'a pas été très dur à apprendre car nous avions déjà de bonnes notions de C et de programmation orientée objet. De plus il s'utilise facilement à partir du `C#` : comme HappyHttp est directement compilée dans le `C++` et portable nous avons pu le faire à l'aide de DLLs. Le fait que ces différentes parties s'intègrent bien entre elles nous a permis de reprendre et faire évoluer le projet de l'année dernière sans trop de soucis.

4 Plannification

Voici de diagramme de Gantt réel du projet, qui montre le temps que nous avons consacré sur chaque tâche et leur ordonnancement.



Nom	Date de début	Date de fin
Recherches et plannification	15/09/15	01/10/15
Comprendre le code de l'année précédente	02/10/15	07/12/15
Faire la DLL pour la requête vers la Kira	02/10/15	04/01/16
Rendre la vue opérationnelle pour prototype	04/12/15	31/12/15
SOUTENANCE	05/01/16	06/01/16
Faire la DLL pour la requête vers la Fibaro	06/01/16	17/05/16
Refaire la vue pour l'adapter au modèle	06/01/16	17/05/16
Faire la DLL du modèle	01/02/16	17/05/16
Faire les requêtes AT pour le Bluetooth	01/02/16	01/04/16
Afficher les informations du fauteuil	01/04/16	17/05/16
Faire le rapport	01/04/16	17/05/16
RENDU	18/05/16	19/05/16

FIGURE 2 – Diagramme de Gantt du projet, généré avec GanttProject

Certains éléments ont pris plus de temps que prévu. La réappropriation du code du projet de l'année dernière était plus dure que nous le pensions. Sans rapport technique à notre disposition, nous ne savions pas par où commencer puisque le projet ne fonctionnait pas. Comprendre le code que nous avons repris a donc pris beaucoup de temps.

Ensuite, la génération du code de la DLL pour effectuer une requête HTTP vers la Fibaro a aussi pris beaucoup plus de temps que pour la Kira. Ceci s'explique par le fait que la Fiabro possède un système d'authentification contrairement à la Kira, rendant plus complexe la requête. De plus, quelques erreurs de programmation en `C++` ont retardé la production de la DLL permettant les requêtes HTTP.

De manière générale, les objectifs principaux ont été atteints. D'autres options sur l'application comme l'option de défilement, n'ont pas été réalisées, car il ne s'agit pas d'options nécessaires à l'utilisation de l'application, et l'implémentation de ces options aurait été trop coûteuses en terme de temps.

5 Présentation de l'application

Actuellement, l'application est à un stade avancé. Nous avons implémenté la plupart des fonctionnalités nécessaires.

5.1 Contrôle

FIGURE 3 – Page d'accueil

La page principale de l'application permet d'accéder aux différentes pièces de la maison de l'utilisateur. En cliquant dessus, il accède aux équipements de la pièce en question et peut ensuite déclencher une action associée à cet équipement.

5.2 Feedback

En cliquant sur un bouton, l'utilisateur peut avoir accès à une nouvelle page présentant différentes informations sur le fauteuil. Ces informations sont affichées de manière simple. Il est possible d'améliorer cette page en affichant la vitesse sous forme de compteur, la position du joystick en temps réel sur une cible, de manière plus visuelle qu'une simple valeur, etc.

5.3 Configuration

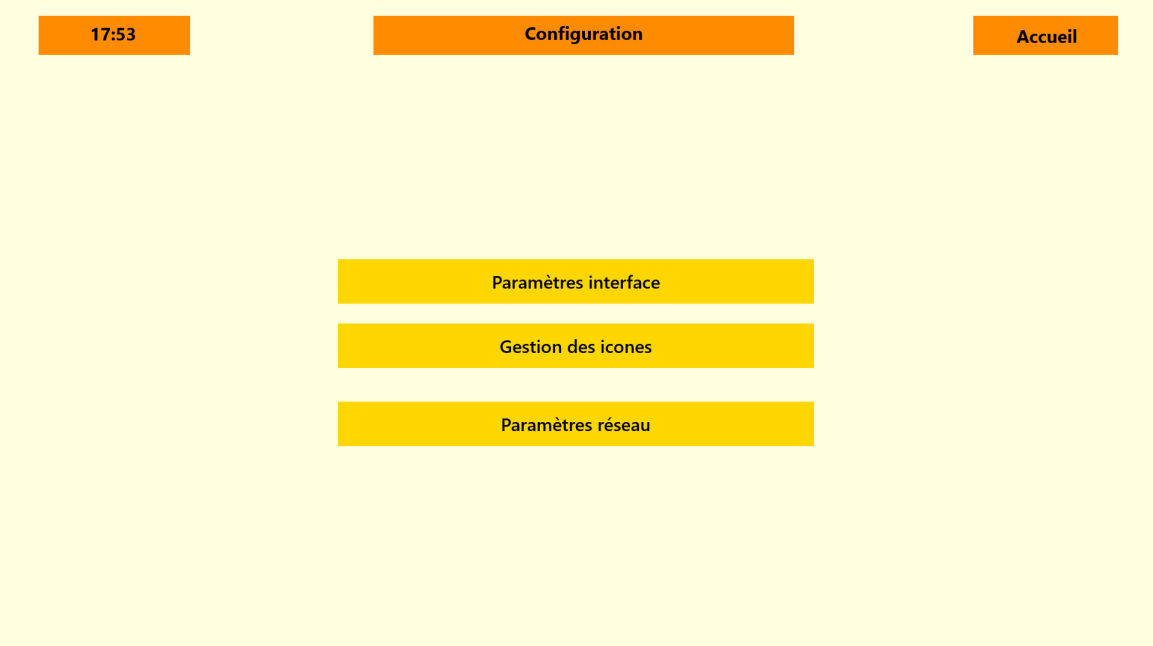


FIGURE 4 – Page de configuration

Pour ce qui est de la partie accessible par un administrateur, il est possible de modifier plusieurs paramètres :

- Modification du thème en cas de déficience visuelle,
- Modification de la taille des icônes,
- Modification des paramètres réseau,
- Ajout de pièces et d'équipements contrôlables,
- Personnalisation des icônes représentant les pièces et les équipements.

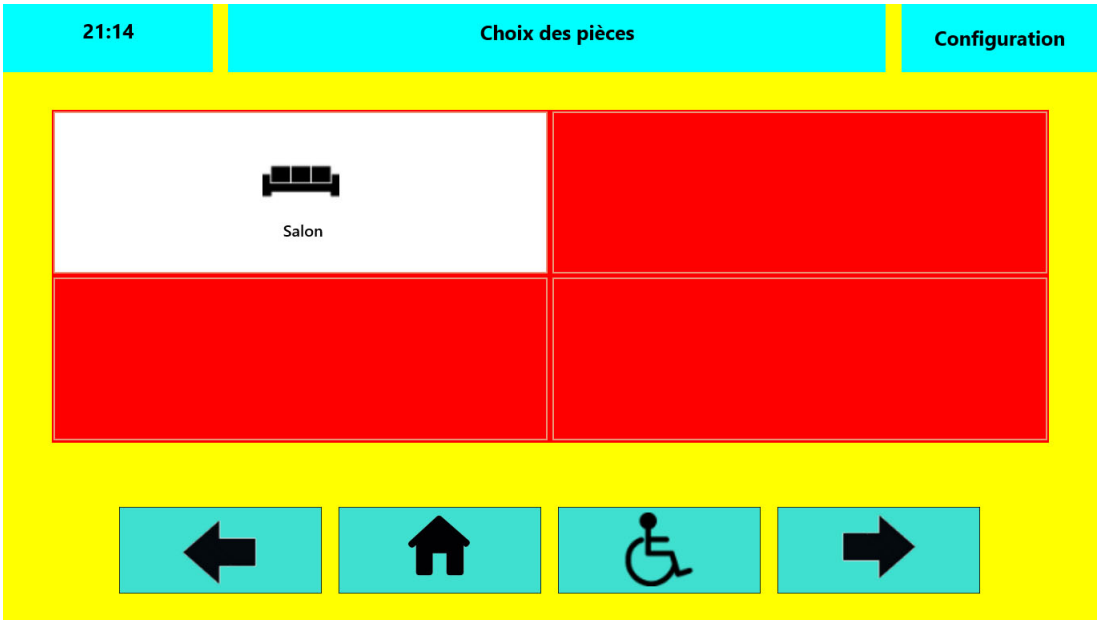


FIGURE 5 – Page d'acceuil pour le 'Thème 2' et la taille 'Grande'

6 Conclusion

Ce projet s'est révélé très enrichissant dans la mesure où il a consisté en une approche concrète du métier d'ingénieur. En effet, la prise d'initiative, le respect des délais et le travail en équipe seront des aspects essentiels de notre futur métier. De plus, il nous a permis d'appliquer nos connaissances à un domaine pratique, qui se révèle aujourd'hui d'intérêt général : le domaine du handicap.

Les problèmes auxquels nous avons fait face sont principalement des problèmes techniques (importation d'une DLL dans un projet `C#`, problèmes de programmation en `C++`..) dûs à notre manque d'expérience. Ce projet nous aura donc permis d'améliorer nos compétences en développement. Nous avons aussi pu réaliser la difficulté de travailler à plusieurs sur un projet : mettre en commun le code produit, savoir utiliser les points forts de chacun, programmer plus tard les tâches qui ont besoin que d'autres soient finies...

Finalement, principalement grâce à une bonne cohésion d'équipe nous avons réussi à atteindre l'objectif que nous nous étions fixé : produire une application utilisable même s'il reste une fonctionnalité non implémentée (système de navigation alternatif).