Application du C++ au domaine des objets connectés

Introduction

Au fil des derniers mois, notre parcours nous a permis de plonger dans l'univers du langage C++. Situé au-dessus du langage C, il est principalement utilisé dans la programmation orientée objet. Au cours de cette période, nous avons exploré les concepts fondamentaux des classes, avec tous les éléments qui les accompagnent : héritage, encapsulation, association et agrégation. Nous avons également découvert des fonctionnalités avancées telles que le polymorphisme, l'abstraction et la généricité.

Le but principal de notre étude a été de concevoir une approche orientée objet répondant aux besoins actuels du domaine des objets connectés et de l'Internet des objets, tout en exploitant pleinement les principales caractéristiques du langage C++ mentionnées précédemment.

C'est ainsi que nous avons porté notre préférence sur le développement d'une plateforme mobile télécommandée, permettant aux pompiers d'accéder à des zones d'incendies difficilement accessibles ou risquées tout en diminuant leur exposition au danger.

En pratique, un pompier situé à distance peut commander le déplacement de la plateforme via une télécommande infrarouge, ainsi que le déplacement (latéral et vertical) du canon situé sur le dessus, qui jouerait le rôle de lance à incendie afin d'éteindre les feux sans s'exposer de façon trop importante.

Notre plateforme n'est qu'un prototype, et ne comprend donc pas ni de réservoir d'eau ni de système de mise sous pression de l'eau, mais se focalise sur le déplacement du « tank », le mouvement du canon, ainsi que le déploiement d'un mode « danger » (son et écran) permettant de prévenir les riverains en cas d'intervention.

Description du char

Nous avons développé un char doté d'une mobilité et d'un contrôle de tourelle. Notre char fonctionne selon deux modes : le mode "happy" et le mode "danger". Le premier mode est destiné à une utilisation lorsque le public n'est pas en danger, tandis que le deuxième mode est conçu pour des situations d'urgence, où il est nécessaire d'alerter le public. Cela est réalisé grâce à l'émission d'une musique (via un buzzer) et l'affichage sur un écran LCD. Dans les deux cas, le tank et la tourelle peuvent être contrôlées par une ou plusieurs télécommandes.

Matériel utilisé

Vous trouverez ci-dessous le matériel utilisé pour la conception de notre tank :

- 4 moteurs DC
- 2 servomoteurs
- Pile de 9 V
- Batterie externe
- Rouleau de papier toilette pour simuler la tourelle
- 1 buzzer
- 1 écran LCD
- 4 roues
- Base de la tourelle en plastique

- Base roulante en plastique
- MEGA 2560
- Arduino Uno
- Potentiomètre
- 1 télécommande

Pour créer notre tank nous avons utilisé deux microcontrôleurs, chacun ayant une fonction spécifique. Tout d'abord, nous avons utilisé un microcontrôleur Arduino Uno avec un shield intégré comprenant un récepteur IR et des drivers de moteur DC, ce qui nous a permis de contrôler le mouvement de la base roulante, des servos moteurs et d'utiliser la télécommande. Puis, le deuxième microcontrôleur, MEGA 2560, s'est chargé des fonctions du mode "danger", qui comprend des lumières LED, de la musique et un écran LCD. En plus des microcontrôleurs, nous avons utilisé 2 servomoteurs, 4 moteurs à courant continu, des câbles, un écran LCD, un haut-parleur, des lumières LED, 4 roues, une base en plastique pour le réservoir, une batterie et une télécommande. La plupart de ces composants ont été récupérés dans différents kits Elegoo que nous avions déjà à la maison.

UML

Tout au long de la conception de notre projet, nous avons considéré deux pilotes distincts : le pilote du char et de la tourelle. Cette distinction pourrait se faire en réel à l'aide de deux télécommandes.

Diagramme cas d'utilisation

Vous trouverez en annexe 1 le diagramme des cas d'utilisation initial et le final dû aux problèmes rencontrés.

Diagramme de séquence

Vous trouverez en annexe 2 et 3 les diagrammes de séquences suivants qui permettent le fonctionnement du char : mouvement base roulante, mouvement de la tourelle et danger mode. Avec Servo 1 le servo qui s'occupe de la rotation horizontale et Servo 2 de la rotation verticale.

Diagramme de classes

Comment faire marcher notre système

Le mode de fonctionnement de notre char est simple. Pour le contrôler, une télécommande basique est requise. Les flèches de la télécommande permettent le déplacement du char, le bouton "ok" permet de l'arrêter, le bouton "1" permet de le démarrer, les boutons 4 et 7 contrôlent le mouvement du premier servomoteur, les boutons 5 et 8 contrôlent le mouvement du deuxième servomoteur. Le bouton 3 permet de passer du mode "happy" au mode "danger" et le bouton 6 permet de passer du mode "danger" au mode "happy". Il est important de noter que les codes des boutons peuvent varier d'une télécommande à une autre. Par conséquent, il est recommandé d'utiliser la fonction « Serial.println » pour vérifier ces codes spécifiques. En affichant les valeurs renvoyées par les boutons sur le moniteur série, vous pourrez déterminer les codes exacts associés à chaque bouton de votre télécommande. Cette étape de vérification vous permettra ensuite d'adapter votre programme en fonction des valeurs obtenues.

Difficultés rencontrées

Comme on peut s'y attendre, le chemin parcouru n'a pas été sans difficultés.

Tout d'abord, nous avons tenté de créer une classe de servo-moteur personnalisée qui permettrait de suivre la position du moteur à tout moment, mais cela s'est avéré difficile. En effet, nous avons rencontré des problèmes de création d'objets qui ont empêché le servo-moteur de fonctionner correctement. Bien qu'il ait réussi à atteindre la position souhaitée, il alternait entre deux valeurs au lieu de rester immobile. Par conséquent, nous avons décidé d'utiliser la classe de servo-moteur proposée par Arduino pour éviter d'autres complications.

En deuxième lieu, bien que notre objectif principal fût d'utiliser le moins de microcontrôleurs possible, nous avons réalisé que nous devions être en mesure de déplacer le char d'assaut en même temps que la tourelle, les fonctionnalités supplémentaires (musique, LED et écran LCD), et que nous devions donc être capable d'exécuter des actions en parallèle. Nous n'avions pas d'autre choix.

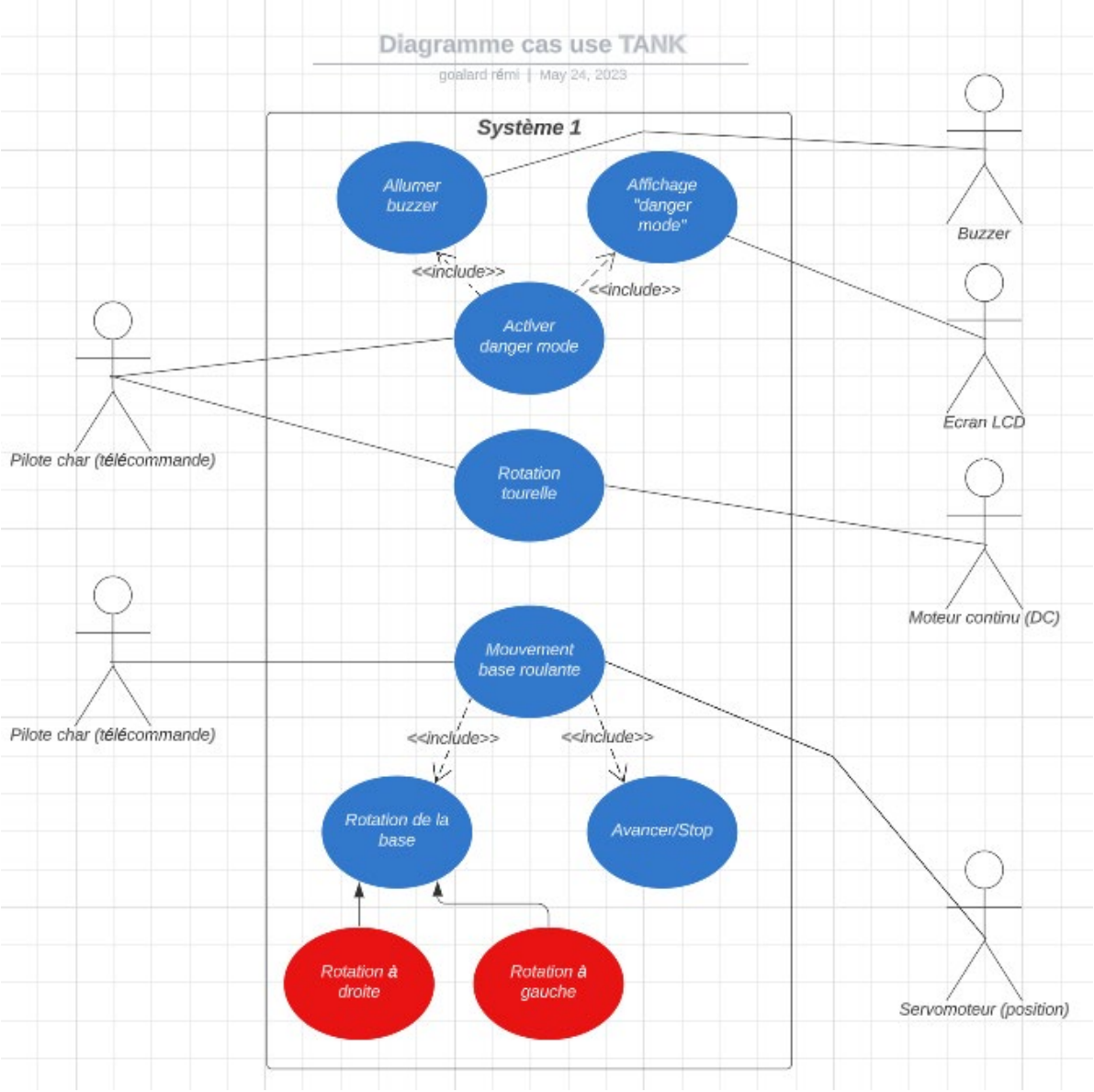
En troisième lieu, nous avons rencontré des difficultés lors de la création de certaines fonctionnalités. En effet, quelques fonctions ont été écrites dans le fichier « .hpp » et non pas dans le fichier « .cpp » correspondant dû à des erreurs de compilation. Probablement ceci est dû à la façon dont elles ont été définies dans les librairies que nous avons utilisées. De même pour certaines variables. Elles ont été créées dans le main puisque dans un fichier « .hpp » créent des erreurs de compilation que nous n'avons pas été capables de résoudre.

Améliorations possibles

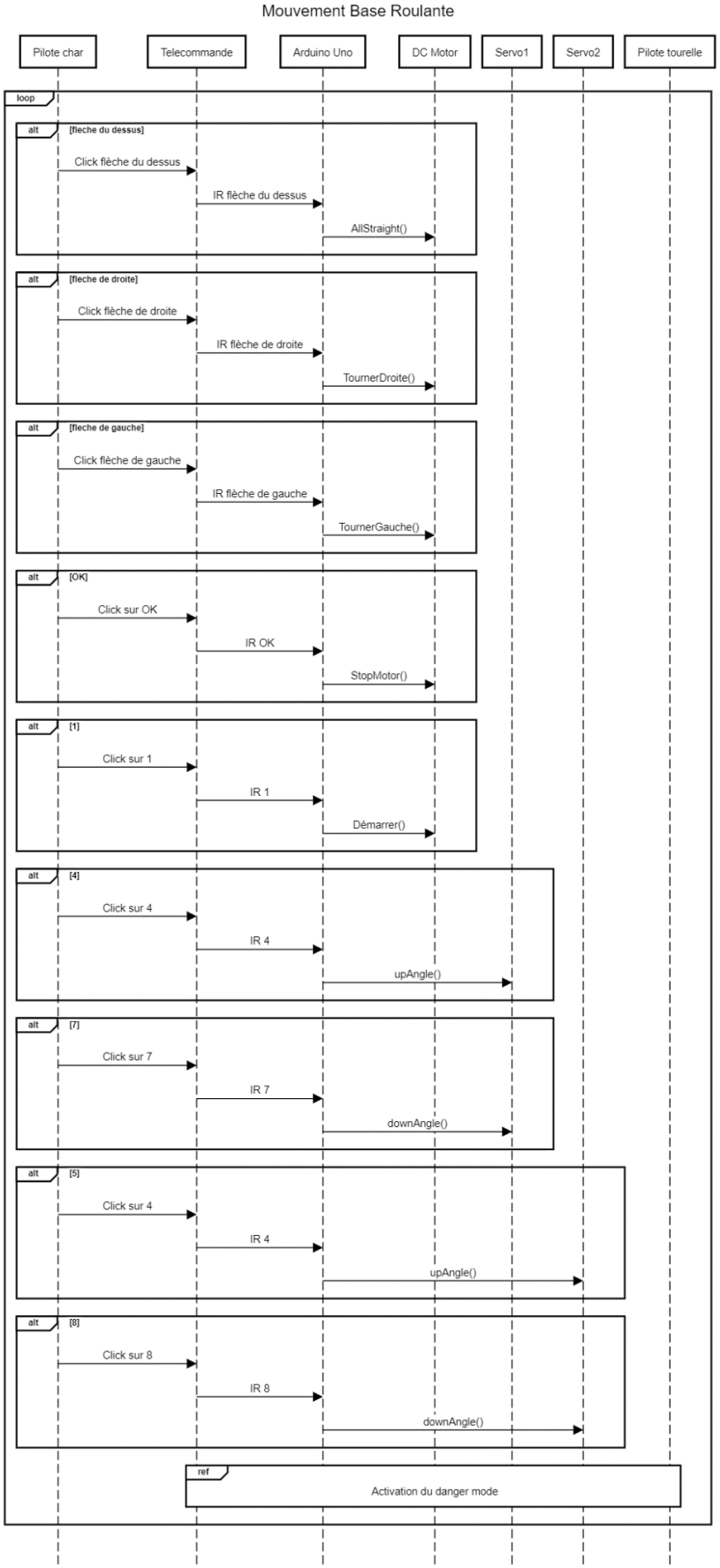
Il y a plusieurs améliorations possibles. Tout d'abord, il serait intéressant d'ajouter plusieurs chansons pour la partie musique, afin de pouvoir différencier plusieurs échelles de danger. Puis, un aspect intéressant serait de remplacer la télécommande IR par une application mobile ou un site internet qui permettrait de contrôler d'une part la base roulante et de l'autre la tourelle par WIFI. Nous avons essayé dans un premier temps de réaliser ceci. Or, quatre séances de projet n'ont pas été suffisantes. Une autre fonctionnalité que nous n'avons pas ajouté par manque de temps ce sont des leds afin d'alerter visuellement une situation de danger. Enfin, il pourrait être intéressant d'explorer d'autres fonctionnalités à ajouter au char, comme un système de communication sans fil pour permettre à plusieurs chars de communiquer entre eux et de travailler ensemble dans leur quête dans monde meilleur.

Annexes

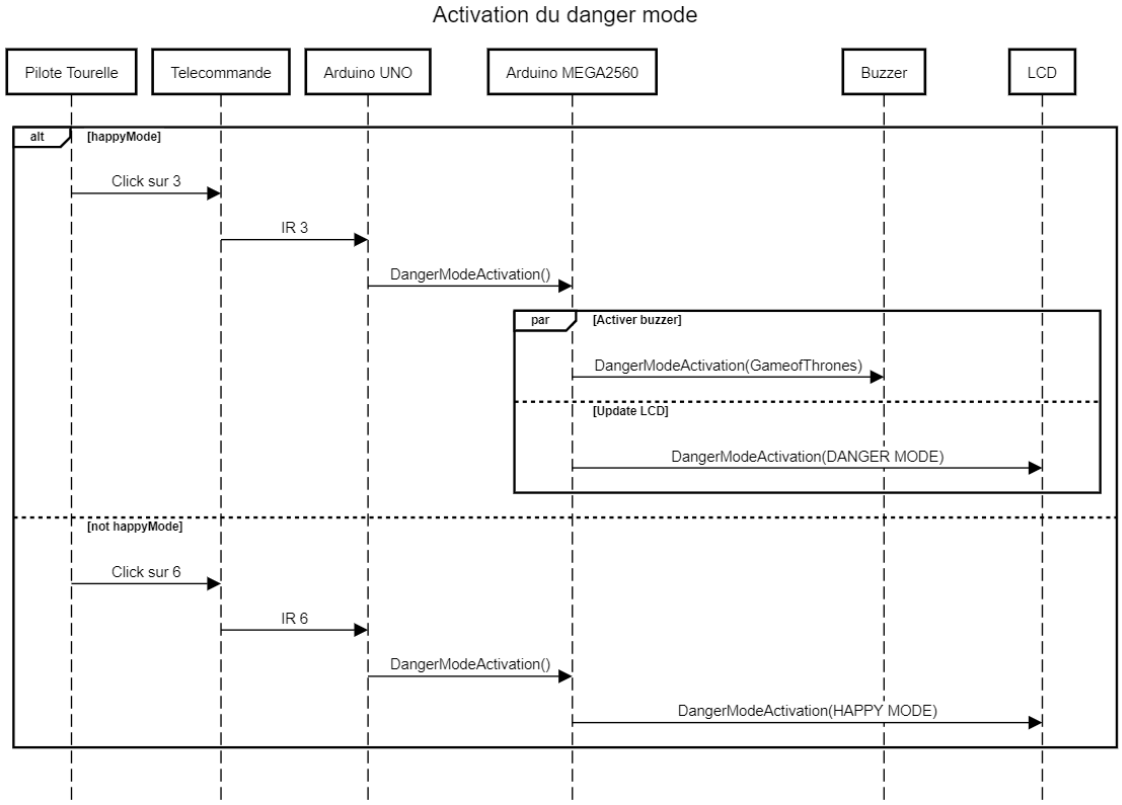
1. Diagramme des cas d'utilisation



2. Diagramme de séquence de la base roulante



3. Diagramme de séquence d'activation du danger mode



4. Diagramme de classes