
Sommaire :

Cas d'utilisation “Jouer le scénario”	2
Manipuler la radiocommande ou l’application mobile.....	3
Logiciels utilisés :	4
Protocole d’échange entre le serveur HTTP et la radiocommande :.....	5
Tableau des valeurs envoyées au serveur HTTP :.....	7
La radiocommande :	8
L’application mobile :	10
Interfaces de l’application mobile :	12
Les différentes difficultés rencontrées :	16
Etat d’avancement sur le projet :	18
Les apports du projet :	19
Diagramme de GANTT :	22

Cas d'utilisation "Jouer le scénario"

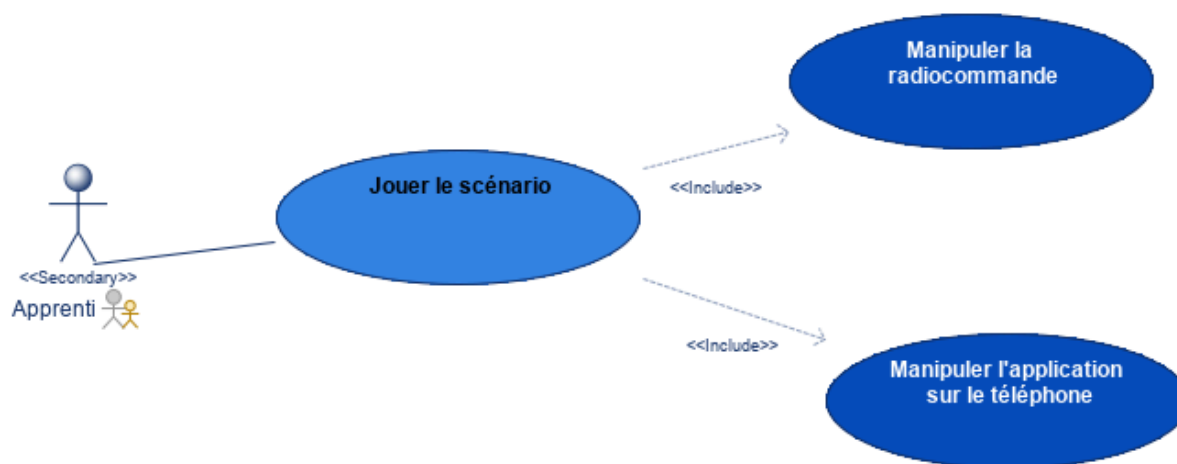


Figure 1 : Diagramme du cas d'utilisation "Jouer le scénario"

Manipuler la radiocommande ou l'application mobile

Précondition	La connexion entre la radiocommande et l'ordinateur est établie. La connexion entre le smartphone et l'ordinateur est établie.
Scénario normal	L'apprenti manipule la radiocommande ou l'application mobile en utilisant le gaz, le roulis, le tangage et le lacet. Cela génère des commandes qui sont envoyées de la radiocommande en connexion USB qui ensuite sont convertis par l'ordinateur.
Post-conditions	Le drone se déplace dans la simulation en recevant les trames de la radiocommande ou de l'application Android.
Post-condition échec	Le drone ne se déplace pas dans la simulation

Logiciels utilisés :



QT Creator

J'ai utilisé le logiciel QT Creator afin de pouvoir développer mon application mobile en QML et pour développer un programme pour recevoir les commandes de la radiocommande. Grâce à ce programme, je vais pouvoir envoyer les commandes au serveur HTTP afin de piloter le drone dans la simulation.



GitHub

J'ai utilisé GitHub afin de faire plusieurs commits pour sauvegarder et importer/exporter mes codes que j'ai effectués. Voici le lien du projet drone virtuel : https://github.com/avillatte/Projet_Drone_2025.git.

Dossier CIEL2 2025s

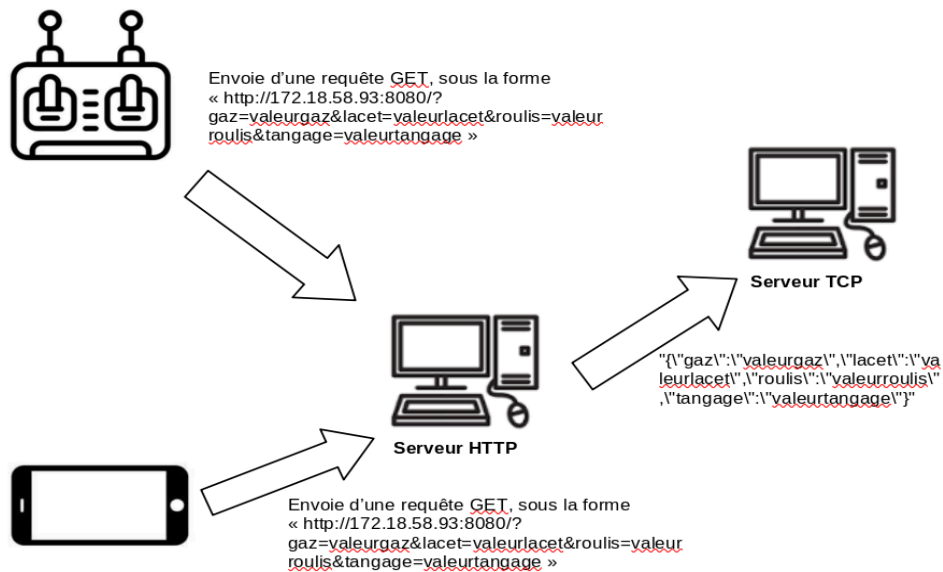
Simulation de pilotage de drone



Protocole d'échange entre le serveur HTTP et la radiocommande :

Nous avons utilisé le protocole de communication HTTP en méthode GET car il présentait plusieurs avantages pour nous. En premier lieu, le fait de pouvoir rentrer directement les valeurs que nous voulions tester directement dans l'adresse URL, ce qui rend la requête plus facile à lire et à comprendre. Ensuite, la méthode GET permet d'envoyer 2048 caractères de données, ce qui est largement suffisant notamment lors de l'envoi de la requête par rapport aux valeurs et aux paramètres que nous envoyons au serveur.

Figure 2 : Synoptique de communication de la simulation de drone



Cette image illustre un système de transmission de commandes à distance, où une télécommande ou un smartphone envoie une requête HTTP de type GET à un serveur HTTP, contenant des paramètres correspondant aux axes de contrôle d'un appareil (gaz, lacet, roulis, tangage), sous la forme :

`http://172.18.58.93:8080/?gaz=...&lacet=...&roulis=...&tangage=...`

Le serveur HTTP reçoit cette requête, extrait les valeurs, puis les reformate en un objet JSON tel que :

`{"gaz":"valeurgaz","lacet":"valeurlacet","roulis":"valeurroulis","tangage":"valeurtangage"}`.

Ce message JSON est ensuite transmis à un serveur TCP, qui pourra relayer ces instructions à l'appareil commandé, comme un drone ou un robot. Cette architecture permet de séparer l'interface de commande (utilisateur) du traitement réseau, facilitant l'adaptabilité du système à différents types de contrôleurs tout en assurant une communication claire et structurée vers la machine cible.

Dossier CIEL2 2025s

Simulation de pilotage de drone

Tableau des valeurs envoyées au serveur HTTP :

Axes	Valeurs application mobile	Valeurs radiocommande
Gaz	gaz:"0 à 100"	gaz:"-54 à 82"
Lacet	lacet:"-100 à 100"	lacet:"-56 à 67 "
Tangage	tangage:"-105 à 105"	tangage:"-97 à 64"
Roulis	roulis:"-100 à 100"	roulis:"-60 à 63"

En revanche ces valeurs étant trop grandes pour Unreal Engine, nous devons diviser les valeurs pour rendre la simulation du vol de drone plus réaliste.

Pour la radiocommande et l'application mobile, nous divisons les valeurs de gaz par 50, et les valeurs de lacet, tangage, roulis par 300.

La radiocommande :

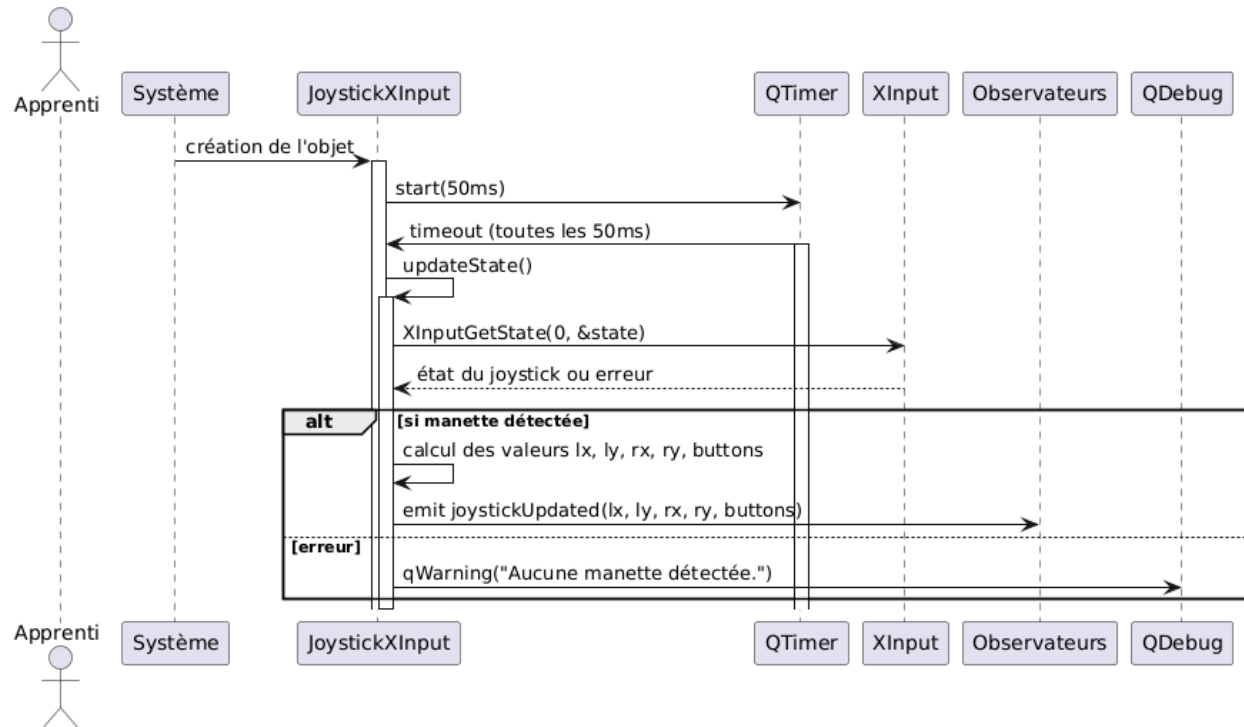


Figure 3 : Diagramme de séquence pour la radiocommande

Dossier CIEL2 2025s Simulation de pilotage de drone

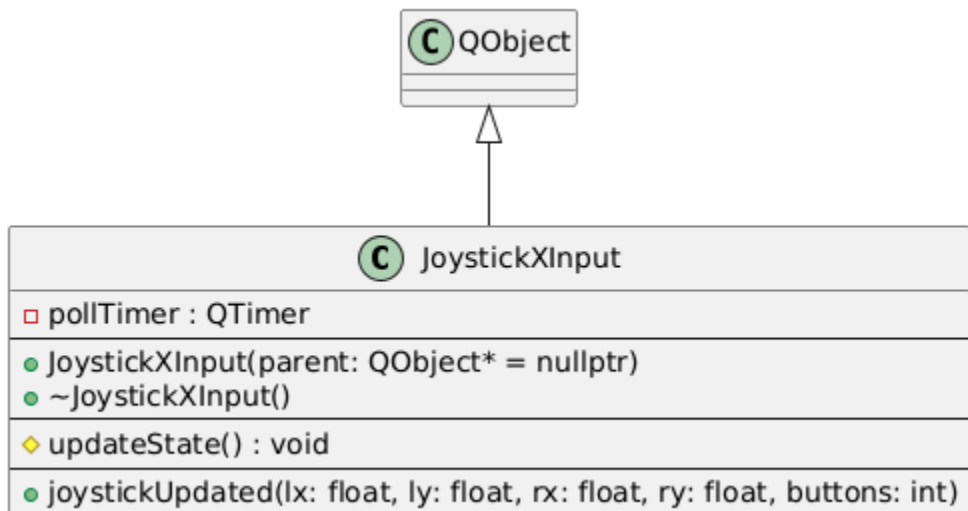


Figure 4 : Diagramme de classes pour la radiocommande

Dossier CIEL2 2025s Simulation de pilotage de drone

L'application mobile :

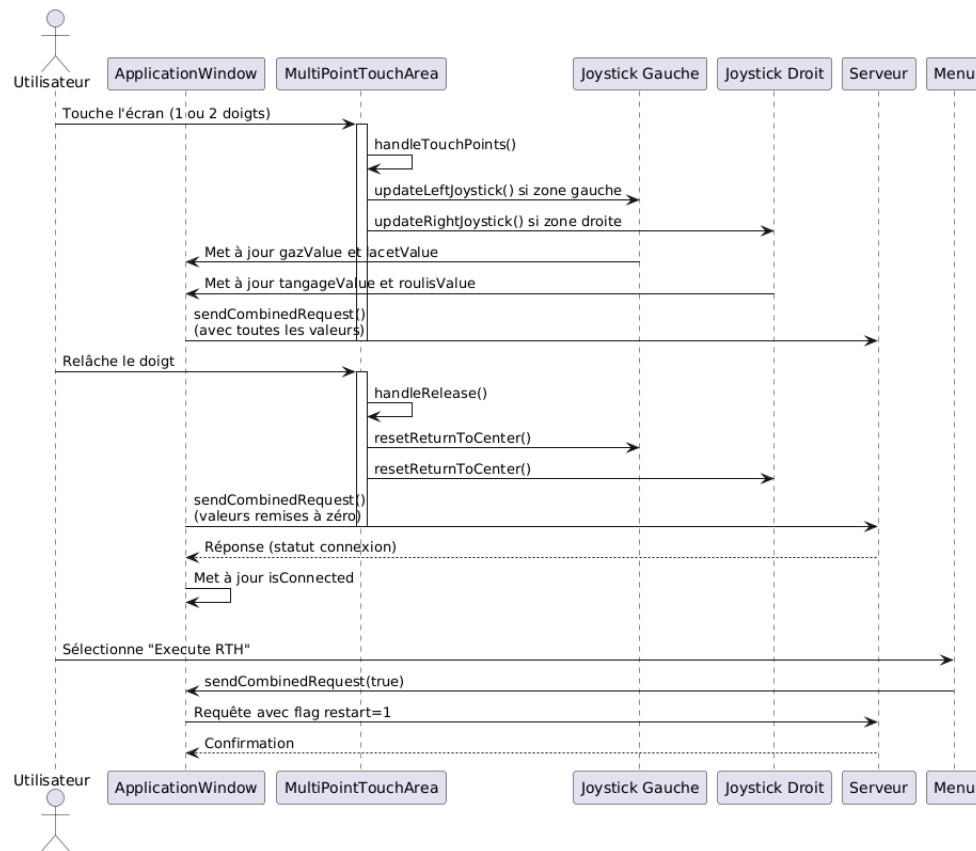


Figure 5 : Diagramme de séquence pour l'application mobile

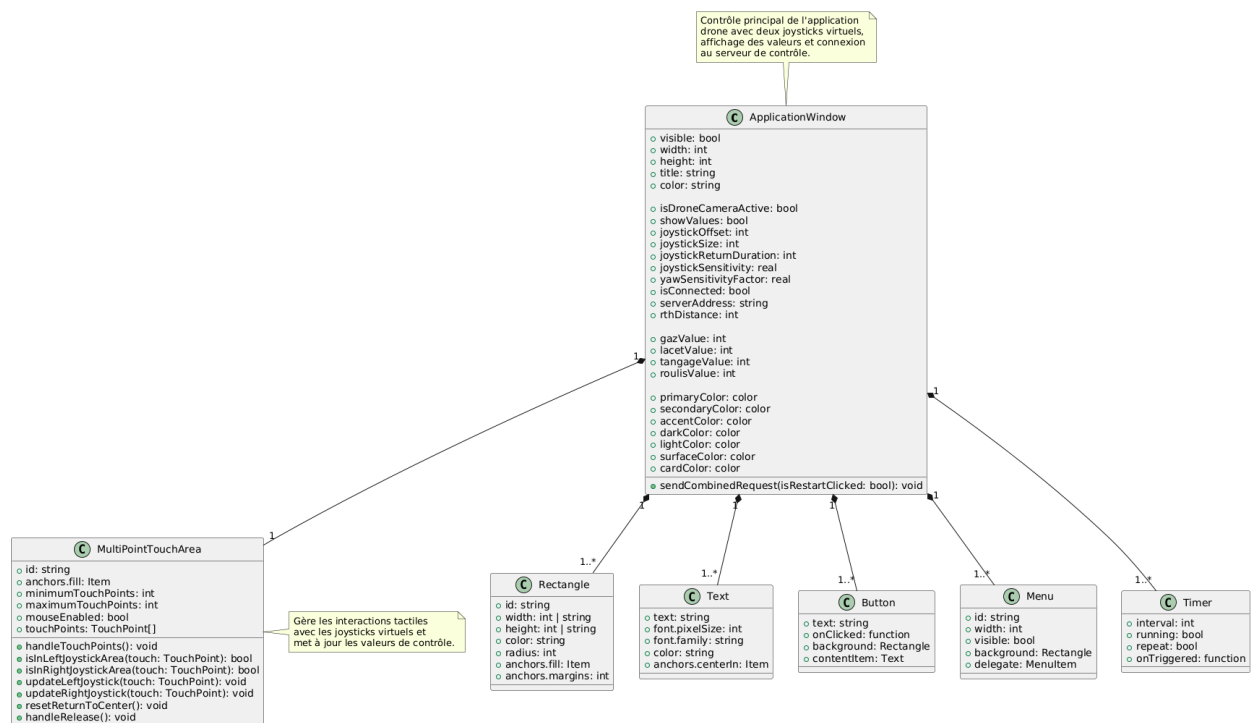


Figure 6 : Diagramme de classes pour l'application mobile

Interfaces de l'application mobile :

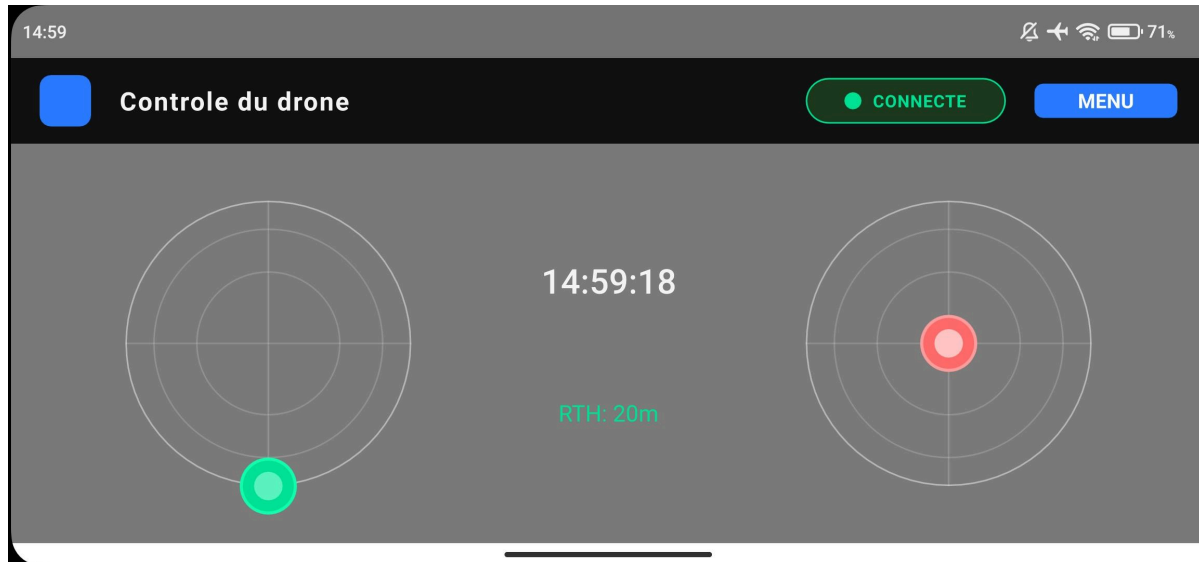


Figure 8 : Interface de l'application mobile avec les joysticks et différents boutons

Voici l'interface de l'application mobile afin de pouvoir diriger le drone dans la simulation et d'effectuer un pilotage sans avoir besoin d'avoir la radiocommande à proximité. Nous pouvons voir les deux joysticks de pilotage, un à gauche pour les *gaz* et le *lacet* et à droite pour *roulis* et le *tangage*.

Dossier CIEL2 2025s

Simulation de pilotage de drone

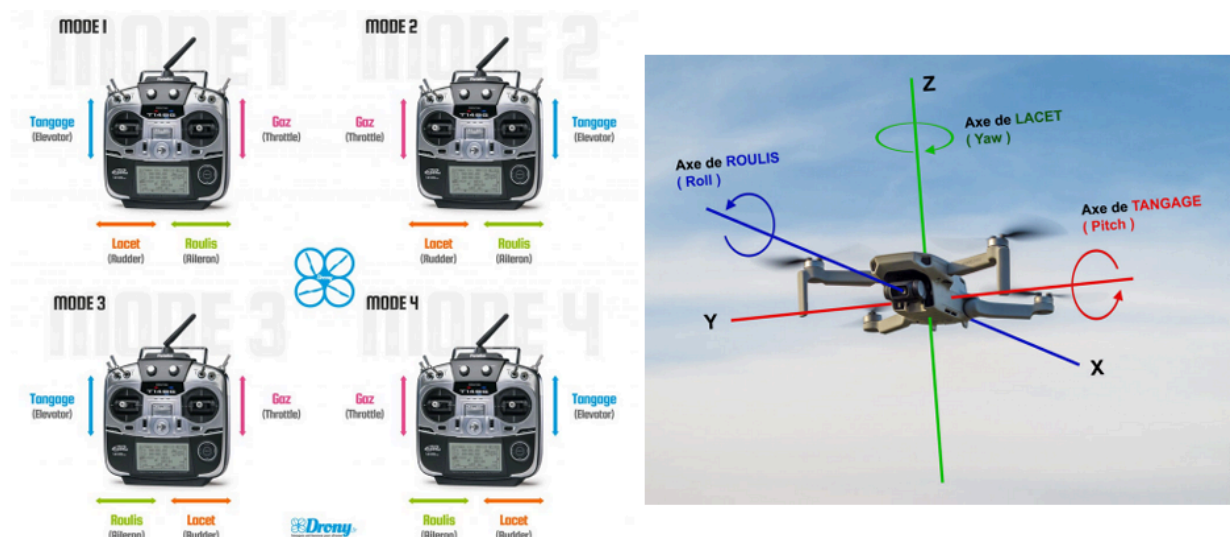


Figure 7 : Les différents axes d'un drone

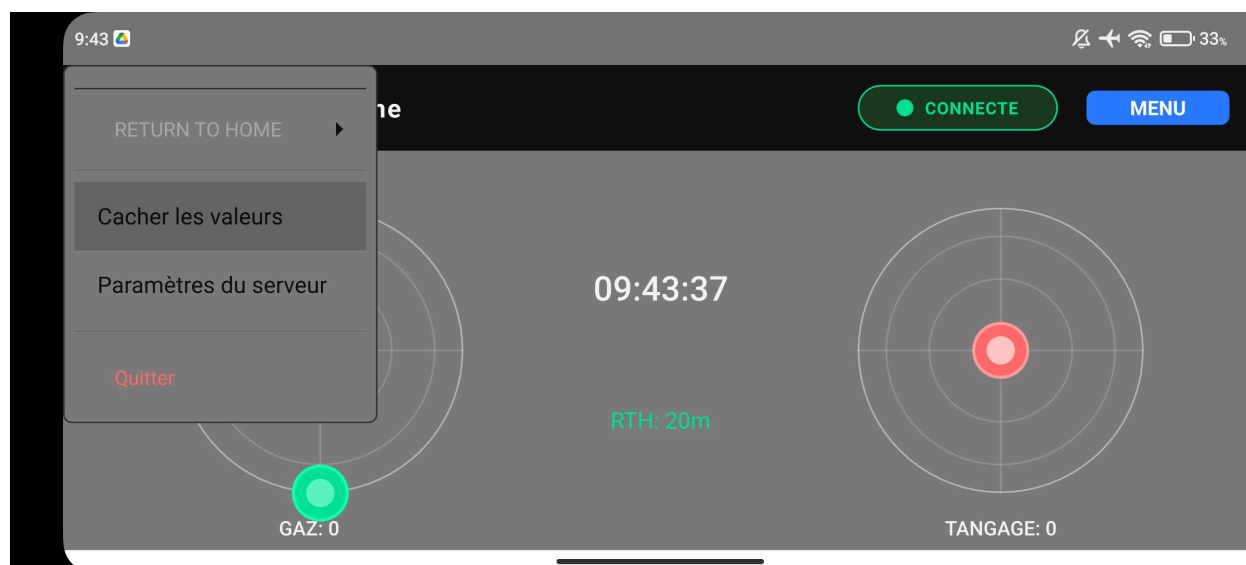


Figure 8 : Interface de l'application mobile avec le menu

Dossier CIEL2 2025s

Simulation de pilotage de drone

Voici le menu afin d'accéder aux paramètres de l'application. Nous avons plusieurs possibilités dont cacher les valeurs des différents axes, de pouvoir changer l'adresse du serveur que je vais vous montrer par la suite et l'option RTH que j'expliquerai également par la suite.



Figure 9 : Interface de l'application mobile avec le changement d'adresse

Voici la fenêtre pour changer l'adresse du serveur. Cela va notamment permettre de pouvoir se connecter à un nouveau serveur au lieu de lancer et téléverser le programme sur le téléphone. L'avantage est que nous n'avons pas besoin d'avoir un ordinateur à proximité pour se connecter à nouveau sur un serveur.

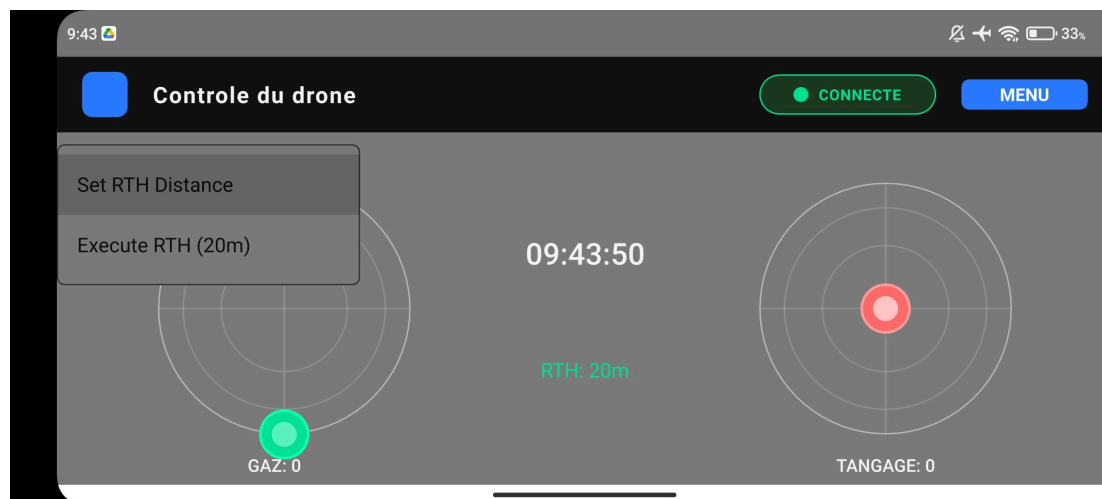


Figure 10 : Interface de l'application mobile avec le Return To Home

Le RTH (Return To Home), ou retour à la maison, est une fonction des drones qui permet de le faire revenir automatiquement à son point de départ. Il existe plusieurs modes de RTH :

1. **RTH manuel (Smart RTH)** : ce mode est déclenché par le pilote, généralement en appuyant sur un bouton de la radiocommande ou dans l'application. Le drone revient alors automatiquement au point de départ enregistré au décollage.
2. **RTH automatique en cas de perte de signal (Failsafe RTH)** : si le drone perd la connexion avec la radiocommande pendant un certain temps, il enclenche automatiquement le retour au point de départ. Si la connexion est rétablie, le pilote peut reprendre le contrôle.

-
3. **RTH en cas de batterie faible (Low Battery RTH)** : lorsque le drone détecte que sa batterie est trop faible pour continuer le vol en toute sécurité, il déclenche le retour à la maison de lui-même. Cela évite qu'il tombe en panne de batterie en plein vol.

Dans tous les cas, le drone commence par monter à une altitude prédéfinie pour éviter les obstacles, puis revient en ligne droite vers le point de départ, et enfin atterrit automatiquement. Le bon fonctionnement de cette fonction dépend du GPS, donc il est important que le drone ait bien enregistré sa position de départ au moment du décollage.

Les différentes difficultés rencontrées :

Lors de mon projet, j'ai rencontré plusieurs difficultés, notamment en apprenant à utiliser le langage QML, qui possède une syntaxe spécifique et nécessite de comprendre le lien entre l'interface graphique et la logique métier, souvent en C++. La prise en main de QML a été un vrai défi pour créer des éléments interactifs comme un bouton "Return to Home" et le connecter à une commande réelle envoyée au drone. Par ailleurs, comprendre le fonctionnement d'une application mobile m'a demandé de m'adapter à des contraintes spécifiques, comme la gestion des permissions, la communication en temps réel avec le drone (pour recevoir sa position ou déclencher le RTH), et l'affichage dynamique des données, tout en tenant compte des limites d'un environnement tactile et mobile. Ces aspects techniques ont compliqué l'intégration fluide de la fonction RTH dans l'application, car ils nécessitent une bonne coordination entre l'interface utilisateur, les données reçues du drone et la logique de commande.

L'une des difficultés majeures du projet a été de trouver un protocole de communication efficace entre l'application mobile, la radiocommande et le casque virtuel, car ces trois éléments utilisent des technologies différentes et doivent échanger des données en temps réel, avec un minimum de latence, notamment pour garantir une bonne réactivité dans l'expérience de réalité virtuelle. Il fallait que ce protocole soit à la fois léger, compatible avec les différents systèmes (Android, PC, casque), et capable de gérer des échanges bidirectionnels. En parallèle, j'ai dû comprendre comment la radiocommande communique avec le PC, ce qui impliquait d'explorer les types de connexions disponibles (USB, série, Bluetooth), d'analyser le format des trames envoyées, souvent sans documentation, et d'identifier si un protocole standard comme MAVLink était utilisé ou non. Une fois cette communication comprise, j'ai développé un programme capable de détecter les actions effectuées sur la radiocommande (comme un appui sur un bouton) et de transformer ces actions en requêtes réseau envoyées au serveur. Cela demandait de gérer à la fois la lecture des entrées, leur traitement, et l'émission rapide et fiable des messages, tout en assurant une bonne synchronisation avec l'application et le casque virtuel.

Etat d'avancement sur le projet :

Ce qui a été fait :

L'envoi des commandes depuis la radiocommande vers un serveur HTTP fonctionne. Un programme a été mis en place pour détecter les actions effectuées sur la radiocommande et les convertir en requêtes HTTP. Cela permet déjà de commander certaines actions à distance, comme un retour au point de départ (RTH). L'application mobile a aussi été partiellement développée : l'interface graphique existe et certaines bases sont fonctionnelles, même si la communication avec le serveur n'est pas encore finalisée.

Ce qu'il reste à faire :

Il faut maintenant permettre à l'application mobile d'envoyer des commandes elle aussi au serveur HTTP, comme cela a été fait avec la radiocommande. Ensuite, il reste à améliorer les performances de l'envoi des commandes depuis la radiocommande, notamment en réduisant la latence, en évitant les pertes de données ou les envois répétés non désirés. Enfin, une synchronisation complète entre l'application mobile, la radiocommande et le serveur reste à finaliser, pour garantir que toutes les commandes (comme le RTH) fonctionnent de manière fluide, quelle que soit leur origine.

Les objectifs sont donc de fiabiliser la fonction RTH, d'avoir une application mobile pleinement opérationnelle, et de rendre la communication entre tous les composants stable et performante.

Les apports du projet :



J'ai appris à installer et configurer une application Android, ce qui m'a permis de mieux comprendre les spécificités du développement mobile, notamment la gestion des permissions, la compatibilité avec différents appareils, et les contraintes liées à la consommation des ressources. Cette étape m'a aussi sensibilisé à l'importance des tests sur plusieurs modèles pour assurer la stabilité de l'application.

Dossier CIEL2 2025s Simulation de pilotage de drone



Le travail en équipe a été très formateur. Il m'a appris à mieux communiquer, à partager mes idées et difficultés, mais aussi à écouter et intégrer les retours des autres membres. Cette collaboration a permis une meilleure répartition des tâches selon les compétences de chacun, ce qui a rendu le projet plus fluide et efficace. J'ai compris combien la coordination et la gestion des dépendances sont cruciales pour avancer ensemble sans blocage.



La formation BIMD, spécifique aux drones, a été un apport majeur. Elle m'a donné des connaissances solides sur les principes du pilotage, la sécurité, la réglementation et les aspects techniques liés aux drones. Cette formation m'a permis de mieux comprendre les fonctions essentielles comme le Return to Home, la gestion des communications entre radiocommande et drone, et les protocoles utilisés, ce qui a grandement facilité le développement du projet.

Dossier CIEL2 2025s

Simulation de pilotage de drone



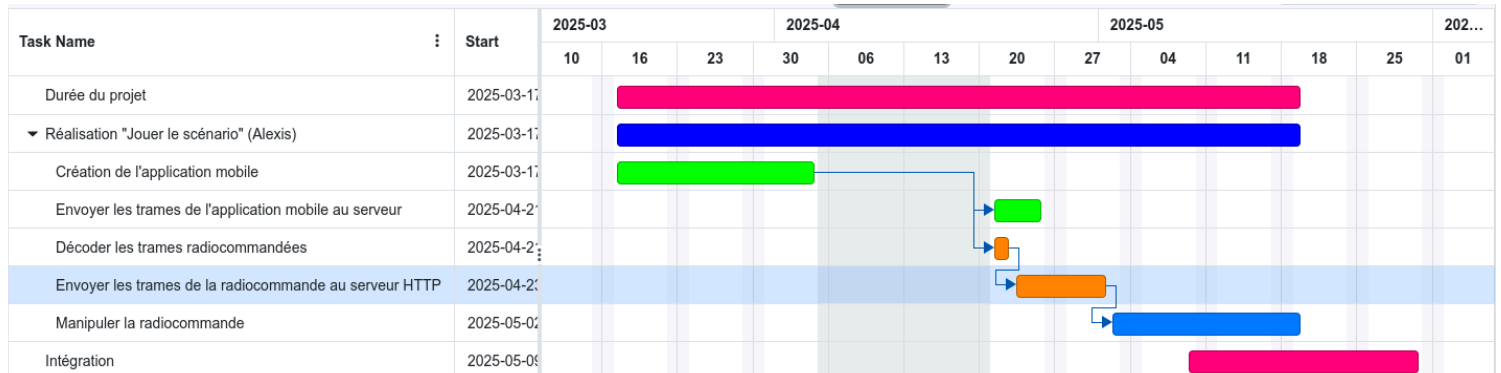
L'organisation a aussi joué un rôle clé. Structurer le projet avec un planning clair, définir des objectifs précis, et anticiper les obstacles nous a permis de respecter les délais et de mieux gérer les imprévus. Cette rigueur dans la gestion du projet a contribué à améliorer la qualité et la cohérence du travail.



Enfin, travailler avec un casque virtuel et le système Meta a été une expérience très enrichissante. Cela m'a permis de découvrir comment intégrer la réalité virtuelle dans un système de contrôle de drone, en tenant compte des contraintes liées à la latence, à la synchronisation des données et à l'interactivité. Manipuler ce matériel m'a donné une meilleure compréhension des défis techniques pour offrir une expérience immersive et fluide aux utilisateurs.

Dossier CIEL2 2025s Simulation de pilotage de drone

Diagramme de GANTT :



Le diagramme de Gantt est un outil de planification qui permet de visualiser l'enchaînement et la durée des différentes tâches d'un projet sur une ligne de temps. Chaque tâche est représentée par une barre horizontale dont la position et la longueur indiquent respectivement sa date de début, sa durée et sa date de fin. Ce diagramme facilite le suivi de l'avancement, la gestion des priorités, l'identification des chevauchements ou des dépendances entre les tâches, et aide à assurer une meilleure coordination entre les membres du projet. Nous avons pris du retard sur la manipulation de la radiocommande car les valeurs étaient trop élevées et il y avait surtout 2 axes identiques donc il fallait les différencier. Sur l'intégration, nous n'avons pas mis beaucoup de temps pour intégrer les différentes parties.