

UNIVERSITÉ PIERRE-ET-MARIE-CURIE

LICENCE INFORMATIQUE 3ÈME ANNÉE

PROJET 3I013

---

## Rapport de Projet

---

*Auteurs:*

Nicolas CASTANET  
Maël FRANCESCHETTI  
Daoud KADOCH  
Fabien MANSON

*Enseignant:*

Fabrice KORDON

3 Avril 2019



# Table des Matières

# **1 Présentation du Projet**

## **1.1 Contexte**

Un client nous souhaite effectuer un vol autonome avec un drone Bebop 2 tout en visualisant sur un iPod Touch le retour vidéo de la caméra embarquée sur le drone. L'iPod sera placé dans un masque de vue à la première personne. La solution devra permettre à l'utilisateur de saisir un plan de vol sur une carte interactive et de le faire exécuter par le drone tout en ayant un retour vidéo sur l'iPod.

## **1.2 Solutions étudiées**

- 1.
- 2.

## 1.3 Use Case

### 1.3.1 Création d'un plan de vol

Dans le premier cas d'utilisation, l'utilisateur souhaite créer un nouveau plan de vol afin de le faire suivre à son drone dans le futur, voici les étapes qu'il va suivre:

15cm—c—X—	
1	L'utilisateur lance l'application et arrive sur la page d'accueil.
2	L'utilisateur lance la fonctionnalité de saisie du plan de vol sur une machine connectée au réseau local.
3	L'utilisateur saisit le plan de vol sur la carte en spécifiant les points de passage du drone ainsi que les altitudes que le drone doit adopter au cours du vol.
4	L'utilisateur valide la saisie de son plan de vol, ce dernier est enregistré.

### 1.3.2 Exécution d'un plan de vol

Dans le deuxième cas d'utilisaton, l'utilisateur possède déjà un ou plusieurs plan de vol enregistrés sur sa machine, par exemple le tour de son énorme jardin qu'il souhaite surveiller sans avoir à ce déplacer. Il fais désormais excécuter au drone l'un de ces plan de vol et observe le retour vidéo sur son Ipod. Voici les étapes suivient :

15cm—c—X—	
1	L'utilisateur lance l'application et arrive sur la page d'accueil.
5	L'utilisateur allume le drone et y connecte sa machine en wifi.
6	L'utilisateur lance la fonctionnalité d'exécution du plan de vol.
7	L'utilisateur démarre l'iPod touch, le connecte au réseau local et lance l'application de réception vidéo. La réception vidéo en temps réel sur l'iPod commence.
8	L'utilisateur sélectionne parmi les plans de vols présents sur le drone celui qu'il souhaite réaliser.
9	L'utilisateur place l'iPod dans le masque FPV, le met sur sa tête, puis lance l'exécution du plan de vol. Le drone décolle.
10	Le drone effectue le plan de vol choisi, l'utilisateur voit en temps réel ce que le drone filme et peut constater l'état d'enneigement de son jardin.
11	L'utilisateur attend la fin de l'exécution du plan de vol pour retirer le masque et aller récupérer le drone une fois qu'il aura atterri à son point de départ, comme prévu.

## 1.4 Rappel du cahier des charges

Nous rappelons ici les principaux points du cahier des charges dans cette section. Les besoins fonctionnels et le diagramme des fonctionnalités se trouvent dans le cahier des charges.

### 1.4.1 Périmètre

#### À qui s'adresse le produit ?

Ce projet est destiné à un public désirant effectuer une ronde d'une durée de 20 à 25 minutes maximum avec son drone.

La portée maximale du drone est de 200 mètres pour une altitude de 100 mètres tout au plus.

#### Les Limites

Le drone ne sera pas capable d'éviter les obstacles durant sa ronde, c'est à l'utilisateur d'établir un trajet cohérent avec son environnement.

De plus, dans le cas où le trajet proposé est trop long par rapport à l'autonomie du drone, un simple message d'avertissement sera affiché pour prévenir l'utilisateur, l'application ne sera pas en mesure d'empêcher cette ronde.

### 1.4.2 Ressources

**Nous rappelons ici le matériel à notre disposition pour la réalisation du projet :**

1. Un drone Bebop 2.
2. Un iPod Touch de 6ème génération sous iOS 12.
3. Un accès aux salles machines SAR équipées de machines sous OSX.
4. Un accès aux salles informatiques équipées de machines sous Linux.

L'équipe est composée de quatre étudiants en troisième année de licence d'informatique.

Nos connaissances en programmation sous Linux sont bonnes mais le langage Objective-C et l'univers iOS nous sont pour le moment encore inconnus.

### 1.4.3 Objectifs négociés

Après avoir examiné la demande du client et retenue une solution matérielle, les objectifs suivants ont été formulés dans le cahier des charges :

1. Permettre à l'utilisateur de saisir un plan de vol étape par étape sur une carte interactive et de spécifier les altitudes du drone à chaque point de passage.
2. Permettre à l'utilisateur de sauvegarder son plan de vol pour le réutiliser plus tard.
3. Permettre à l'utilisateur de lancer l'exécution du plan de vol réalisé au préalable.
4. Rediriger le flux vidéo du drone vers l'iPod Touch.
5. Minimiser les latences vidéo (de l'ordre de la seconde).
6. Permettre d'arrêter le vol en cours en cas d'urgence.

## 2 Architecture Logicielle

### 2.1 Interface utilisateur

L'interface utilisateur est une interface GTK permettant d'interagir avec le SDK Parrot afin d'enregistrer des trajets sur le drone et de les exécuter.

Elle permet également d'interagir avec le Serveur local afin de saisir interactivement le plan de vol à l'aide de l'application javascript et de créer les fichiers Mavlink.

Voici l'architecture d'interaction de l'interface utilisateur:

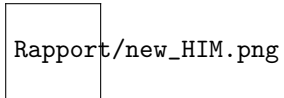


Figure 1: Interface utilisateur

## 2.2 Controle du drone

Le controle du drone est axé autour de la génération du fichiers mavlink à l'aide de l'application JS sur le serveur local et du SDK Parrot qui permet d'envoyer des commandes au drone et de connaître son état.


Après avoir saisi interactivement un trajet à l'aide de l'application javascript, le fichier Mavlink généré est envoyé au SDK Parrot.

Le SDK permet ensuite le controle du drone à savoir:

1. Connexion au drone.
2. Exécution d'un plan de vol.
3. Arrêt d'urgence.

Le décollage et l'arrêt d'urgence du drone sont des commandes envoyés directement par le serveurs, ses commandes seront envoyés au serveur par l'IPod afin de permettre un décollage et un arrêt ergonomique.

Il permet également la réception de l'état du drone et du flux vidéo qui sera par la suite traité avec FFMPEG et envoyé en RTP à l'Ipod.



Rapport/new\_controle\_drone.png


Figure 2: Controle du drone

## 2.3 Application IOS

L'application IOS permet d'afficher le flux vidéo du drone et d'envoyer des requêtes au serveur en effectuant des gestes simples avec l'Ipod afin de démarrage du drone et d'effectuer l'arrêt d'urgence.

Dans un premier temps, l'application scanne le réseau afin de se connecter au serveur local. Après la connexion une requête de demande de description du flux vidéo au format SDP est automatiquement envoyée au serveur.

Le descripteur de flux est alors retourné par le serveur à l'application IOS, en parallèle, elle reçoit également le flux vidéo grâce au protocole RTP, le SDK VLC permet ensuite l'affichage vidéo.



Rapport/new\_archi\_logicielle.png

Figure 3: Application IOS



## 2.4 Architecture logicielle globale

Rapport/Architecture\_logicielle\_v2.jpg

Figure 4: Architecture logicielle

### **3 Problèmes rencontrés**

#### **3.1 Problèmes matériels**

#### **3.2 Solutions**

## 4 Organisation du travail

### 4.1 Diagramme de Gantt

### 4.2 Répartition des tâches

## 5 Déploiement

## 6 Comparatif aux objectifs fixés

## 7 Améliorations possibles

## 8 Conclusion