

Cahier des charges

March 9, 2019

1 Présentation du Projet

1.1 Contexte et définition du problème

Le client souhaite faire effectuer des vols autonomes à un drone Bebop 2, avec un retour vidéo en temps réel sur un iPod Touch. Ce dernier pourra être 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.

Cas d'utilisation :

1) Un utilisateur souhaite effectuer une ronde avec un drone pour surveiller sa propriété au moindre effort.
Voici un exemple des étapes qu'il va suivre chronologiquement :

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.
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 vient de 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.
11	L'utilisateur souhaite stopper l'exécution de plan de vol : par exemple, il a repéré quelque chose d'anormal sur la zone de vol et souhaite s'y rendre au plus vite. Il active alors la procédure d'arrêt d'urgence sur sa machine et retire le masque FPV. Le drone stoppe l'exécution du plan de vol et atterrit sur place si les conditions le permettent.

2) En hiver, un utilisateur souhaite faire effectuer au drone le tour de son jardin pour contrôler le niveau d'enneigement de celui-ci sans risquer de glisser.

Voici un exemple des étapes qu'il va suivre chronologiquement :

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 tout autour de son jardin et spécifie une altitude de quelques mètres. Le tracé est fait de manière à faire revenir le drone à son point de départ après avoir fait le tour du jardin.
4	L'utilisateur valide la saisie de son plan de vol, ce dernier est enregistré.
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 vient de 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.

1.2 Objectifs

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 Expression des besoins

2.1 Besoins fonctionnels

1	Saisie du plan de vol	L'utilisateur doit pouvoir saisir un plan de vol sur une carte à travers une interface intuitive. L'utilisateur doit pouvoir spécifier l'altitude du drone à chaque waypoint (point de passage).
2	Traduction du plan de vol au format Mavlink	Le plan de vol saisi graphiquement par l'utilisateur doit être converti au format Mavlink pour être ensuite envoyé au drone.
3	Envoi du plan de vol au drone	La solution doit prendre en charge la récupération du plan de vol réalisé au préalable et son envoi au drone.
4	Choix du plan de vol	L'utilisateur doit pouvoir choisir le plan de vol enregistré sur le drone qu'il souhaite exécuter.
5	Exécution du plan de vol	Une fois le plan de vol choisi, l'utilisateur doit pouvoir lancer l'exécution.
6	Retour vidéo	Tout au long du vol du drone, le retour vidéo de ce dernier doit être envoyé sur un iPod touch en temps réel et avec une latence minimale.
7	Arrêt d'urgence	A tout moment lors du vol, l'utilisateur doit pouvoir déclencher un arrêt d'urgence pour stopper l'exécution du plan de vol et faire atterrir le drone.

2.2 Besoins non fonctionnels

(à compléter)

3 Solutions étudiées

3.1 Comparatif des solutions

	PC + iPod	iPad + iPod	iPod
confort d'utilisation (interface)	bonne	optimale	mauvaise
arrêt d'urgence possible	sur PC & iPod	sur iPod	sur iPod
ergonomie de l'arrêt d'urgence	bonne	mauvaise	mauvaise
minimisation de latence vidéo	bonne	optimale	optimale
répartition des traitements	majorité sur PC	tout sur iPod	tout sur iPod

Les architectures étudiées sont les suivantes, de gauche à droite dans le tableau :

1. une architecture utilisant un PC sous linux pour la saisie du plan de vol et qui commande le drone, ainsi qu'un iPod Touch sur lequel on redirige le flux vidéo.
2. une architecture utilisant un iPad pour la saisie du plan de vol, ainsi qu'un iPod Touch qui commande le drone et récupère le flux vidéo.
3. une architecture utilisant un iPod Touch pour la saisie du plan de vol, qui commande le drone et récupère le flux vidéo.

Comme on peut le constater, la solution utilisant le PC et l'iPod Touch est plus avantageuse. C'est donc cette solution qui sera préférée.

4 Contraintes

4.1 Matériel

Pour réaliser ce projet nous disposons du matériel suivant :

1. un drone Bebop 2;
2. un iPod Touch (préciser version);
3. un accès aux salles machines SAR équipées de machines sous OSX.
4. un accès aux salles informatiques équipées de machines sous Linux.

Les contraintes techniques se la solution préférée sont :

1. Le PC sous Linux doit avoir deux cartes réseau : une pour pouvoir se connecter au drone en wifi, l'autre pour se connecter au réseau local pour rediriger le flux vidéo.
2. Il faut un réseau local sur lequel sont connectés le PC et l'iPod Touch.
3. Le réseau local doit avoir un accès internet.

4.2 Délais

Date de livraison du produit.

Autres Échéances du projet par exemple le rendu du cahier des charges le . Contrainte de temps : contrainte la plus importante.

(Lister les contraintes par ordre d'importance)

4.3 Autres contraintes

Concernant le plan de vol :

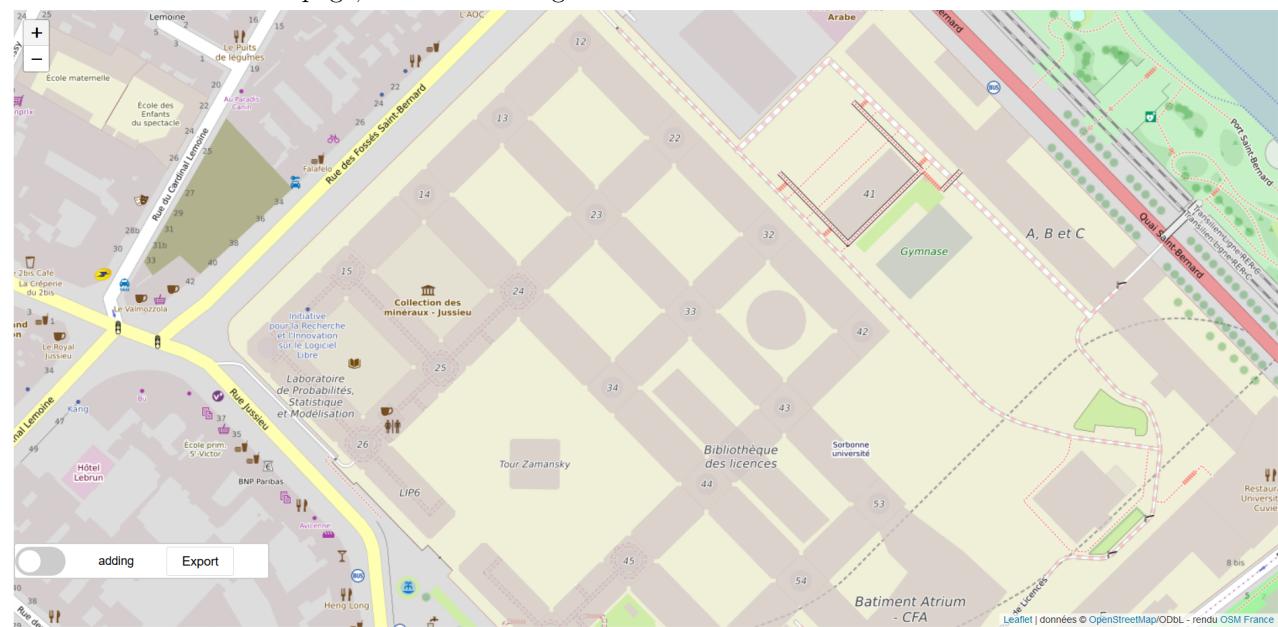
1. L'application de saisie du plan de vol doit intégrer une carte interactive permettant de tracer ce dernier.
2. On devra pouvoir spécifier l'altitude à laquelle le drone doit se trouver aux différents points du parcours.

Concernant le retour vidéo :

1. La qualité du retour vidéo est directement liée à la distance avec le serveur central (le PC) ainsi qu'à l'encombrement de l'espace dans lequel le drone vol. Un terrain avec des obstacles comme des bâtiments ou des arbres réduit fortement la portée du signal vidéo.
2. Le retour vidéo du drone doit être envoyé à un iPod Touch en temps réel.
3. Sa qualité doit être convenable (HD).
4. La latence du retour vidéo sur l'iPod doit être minimale (de l'ordre de la seconde), et la vidéo doit être fluide.

5 Cinématique des écrans :

1. Lors de l'ouverture de la page, la carte est vierge:



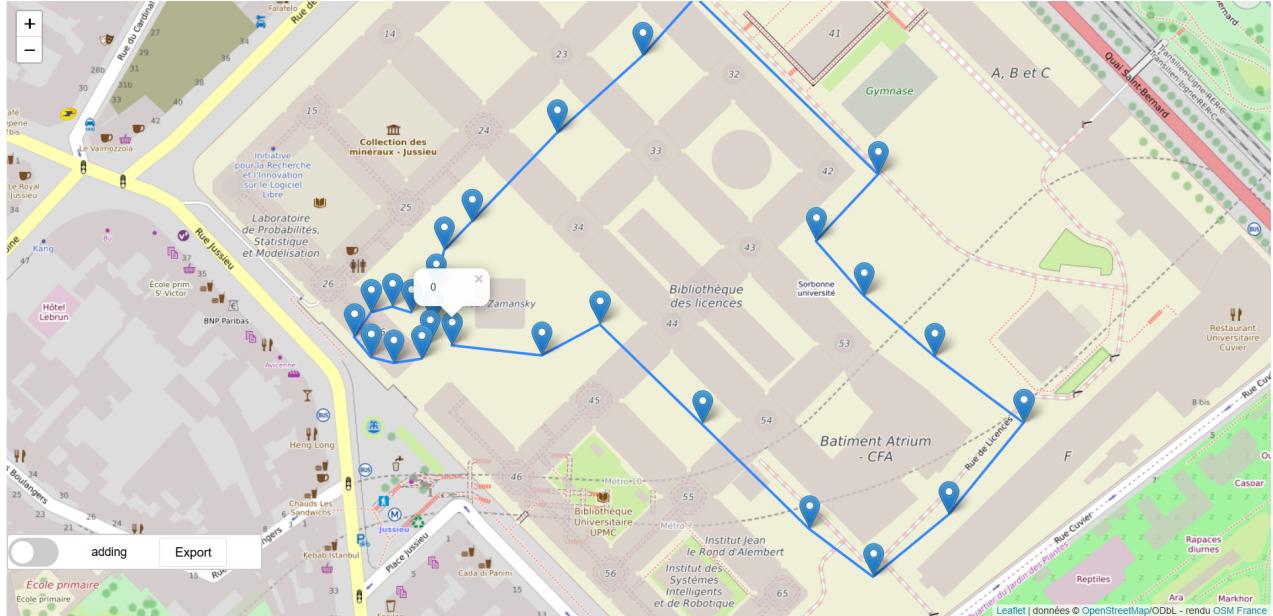
La carte est interactive et on peut la déplacer avec la souris. Les boutons situés en haut à gauche permettent de zoomer et dézoomer sur la carte.

Le cadre situé en bas à gauche contient un switch (interrupteur) et un bouton : le switch permet de changer le mode d'interaction, et le bouton permet de finaliser le plan de vol en l'exportant dans un fichier. Il existe deux modes d'interaction : le mode d'ajout et le mode de suppression. Le mode dans lequel on se situe est indiqué à coté du switch ("adding" / "removing").

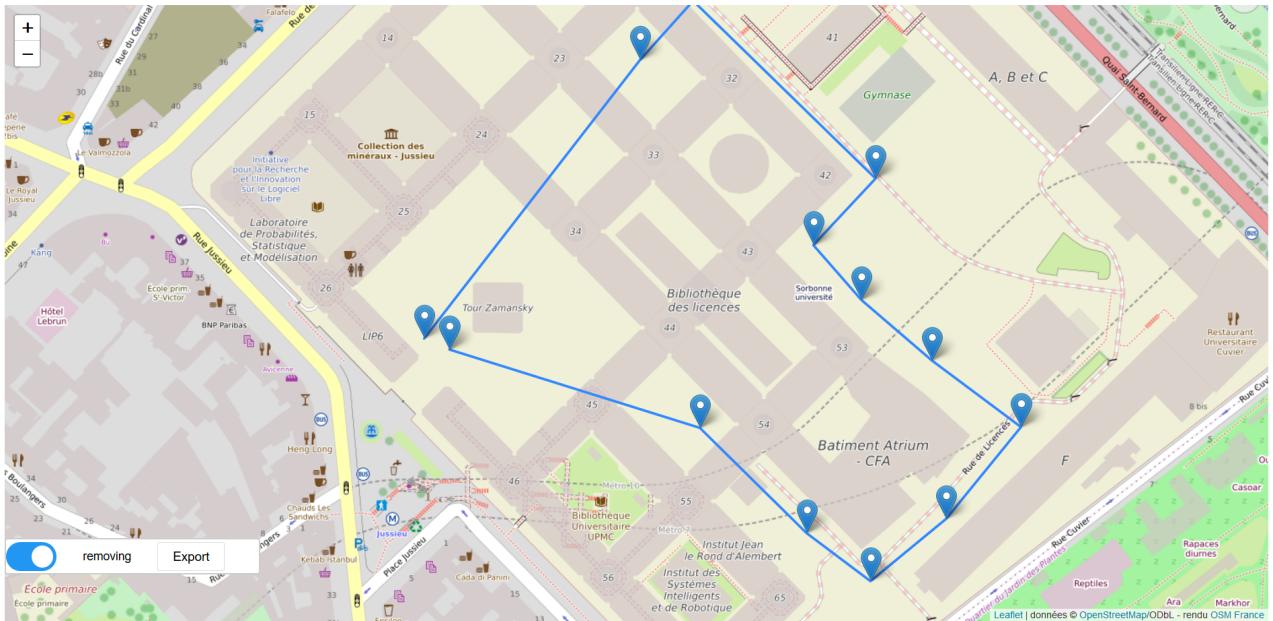
En cliquant sur la carte, un marqueur est placé à l'endroit du clic. Chaque marqueur représente un point de passage du drone dans son itinéraire de vol. Les marqueurs peuvent être déplacés à la souris (drag and drop). Lors du survol de la souris sur un marqueur, une bulle indique son numéro d'ordre (0 pour le point de départ).

- En mode d'ajout, en cliquant sur un marqueur, on peut spécifier dans une boite de dialogue l'altitude à laquelle doit se situer le drone à ce point lors du vol.
- En mode de suppression, en cliquant sur un marqueur, on supprime ce dernier. Les marqueurs sont alors automatiquement remis dans le bon ordre et le trajet est retracé.

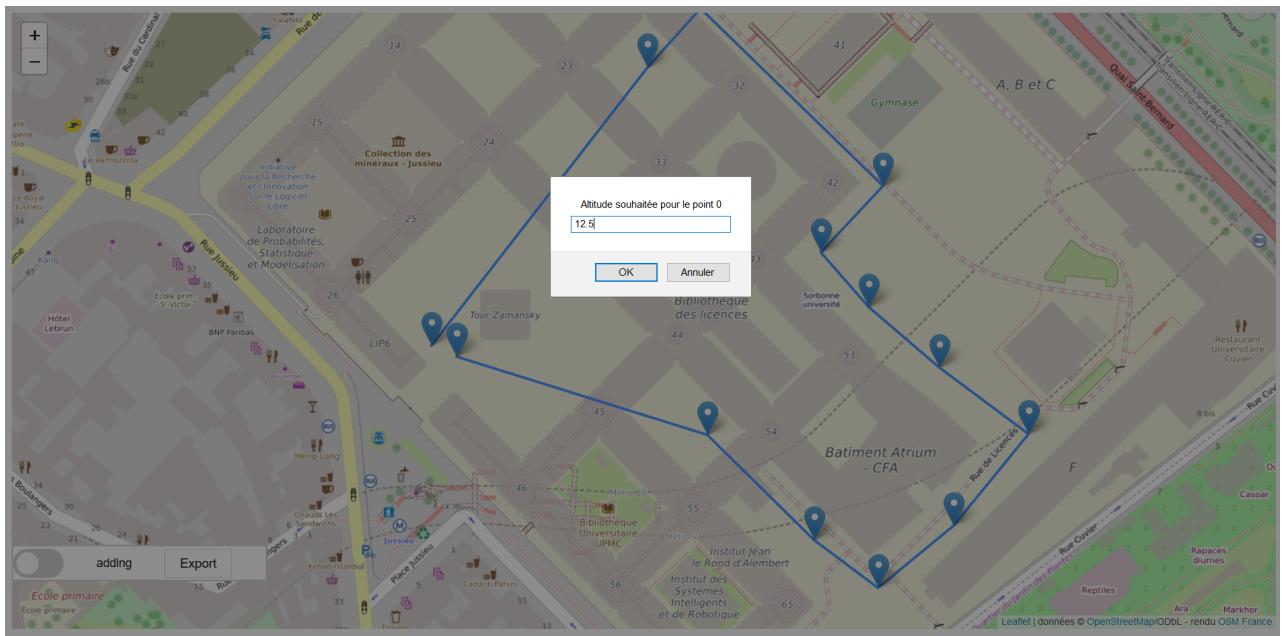
2. On place des marqueurs en cliquant aux endroits voulus sur la carte. Au survol d'un marqueur, la bulle comportant son numéro d'ordre apparaît :



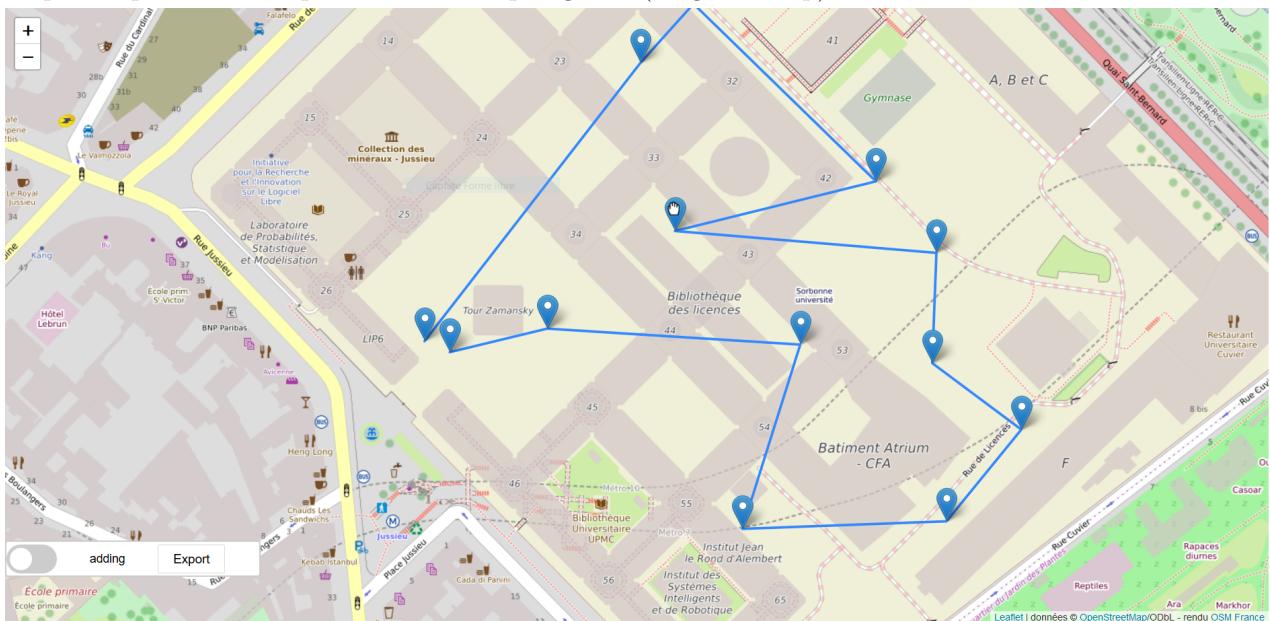
3. On souhaite retirer des marqueurs, on active le mode de suppression en cliquant sur le switch et on clique ensuite sur les marqueurs à supprimer. L'ordre des marqueurs et le tracé du trajet d'adaptent :



4. On clique sur un marqueur pour spécifier l'altitude voulue à ce point :



5. On peut déplacer les marqueurs avec un cliquer-glisser (drag and drop) sur ces derniers :



6. Une fois le plan de vol terminé, on clique sur le bouton pour obtenir le fichier correspondant au trajet défini :

