

# REC[BIRD]



PSTE N°222

Thibault Copart, Bertrand Glaser, Benoit Leguay, Alexis Mahrouss,  
Hugo Massis, Hugo Vaillaud

Mentor : Elisabeth Rendler

## Table des matières

<b>Remerciements .....</b>	<b>3</b>
<b>Introduction .....</b>	<b>3</b>
<b>Présentation de L'équipe.....</b>	<b>4</b>
<b>Etude de L'existant.....</b>	<b>4</b>
<b>Planning .....</b>	<b>8</b>
<b>Détermination du projet .....</b>	<b>9</b>
<b>Méthode de réalisation .....</b>	<b>9</b>
<b>Architecture générale et schématique .....</b>	<b>10</b>
<b>Scenario des prototypes .....</b>	<b>11</b>
<b>Introduction à AutoScript.....</b>	<b>11</b>
<b>Le principe du code de notre projet.....</b>	<b>12</b>
<b>Définitions des variables et variables globales.....</b>	<b>13</b>
<b>Les mouvements du drone .....</b>	<b>14</b>
Avancer/Reculer .....	14
L'atterrissement .....	14
Les déplacements de cotés.....	15
Difficultés rencontrés sur Autoscript.....	15
<b>Difficultés et solutions apportés .....</b>	<b>16</b>
<b>Valorisation.....</b>	<b>16</b>
<b>Bilan.....</b>	<b>17</b>
<b>Collectif.....</b>	<b>17</b>
<b>Individuel .....</b>	<b>17</b>
<b>Sources .....</b>	<b>19</b>

## Remerciements

Ce projet n'aurait pas pu aboutir sans l'accompagnement qui nous a été fourni, nous souhaitions donc chaleureusement remercier

- L'ECE Paris de nous avoir fourni L'AR Drone 2.0 Elite Edition
- L'équipe Ecebrog et plus particulièrement Elodie Kruczowy
- Quentin Cabanes
- Lukas Lao Beyer, le créateur d'AutoFlight
- Enfin, Madame Elisabeth Rendler pour son accompagnement dévoué tout au long de l'année.

## Introduction

De nos jours, que ce soit pour le loisir, lors de la pratique de sports extrêmes, où dans un cadre professionnel, les gens ressentent de plus en plus le besoin de se filmer. Ce phénomène en est même devenu mode. Au cours de ces dernières années, les ventes d'appareils électroniques permettant la prise de vidéos ont considérablement augmentées. On distingue donc deux types de produits largement répandus pour les prises de vidéos. Tout d'abord, nous avons les caméras embarquées représentées par la fameuse « Go Pro ». Bien que pratiques, ces produits ne permettent pas de prendre des vidéos de soi-même avec une bonne qualité sans l'aide d'autrui, ce qui restreint donc leur utilisation. Ensuite, nous avons les drones civils. Ce type d'appareil permet de prendre des vidéos de qualités en grand angle, mais malheureusement, la grande majorité d'entre eux doivent être manipulés à distance par une personne. Nous sommes donc demandés :

### Comment pourvoir prendre des vidéo de soi - même en action sans l'aide d'autrui ?

Dans le cadre de notre PSTE, nous avons donc décidé de concevoir un drone autonome pouvant suivre automatiquement son utilisateur en mouvement. Pour ce faire, nous avons utilisé un drone ArDrone 2.0 de Parrot ainsi que le logiciel AutoFlight sur lequel nous avons conçu le programme permettant l'automatisation de ce drone à l'origine standard. Ainsi, par ce projet, nous proposons une nouvelle alternative susceptible de séduire dans un marché actuellement en essor. Nous allons donc ici vous présenter notre PSTE [REC]Bird sur lequel nous avons travaillé cette année.

## Présentation de L'équipe

Nous avons été six à collaborer pour l'élaboration de ce projet. Tous en deuxième année du cycle préparatoire, nous nous connaissons de la première année de ce même cycle lors duquel nous nous sommes rencontrés. L'idée de faire équipe pour le PSTE de cette année nous avait déjà traversé l'esprit, et alors qu'on nous rappelait qu'il fallait songer à former les équipes au début de l'année, la nôtre s'est formée naturellement.

Des présentations s'imposent : Tous les six avions plusieurs rôles à jouer au cours de l'année et nous ne manquerons pas d'y revenir plus tard. Nous nous présenterons donc plus brièvement en premier lieu afin de vous laisser le temps d'apprendre à nous connaître.

Thibault Copart est le créatif, maniant photoshop avec brio, c'est lui qui a fait certaines des illustrations de ce rapport. Il a su apporter de bonnes idées et répandre sa motivation au sein du groupe.

Bertrand Glaser est le critique (au bon sens du terme). Dans tout projet, l'apport d'idées est indéniablement ce qui fait avancer le projet. Mais déterminer lesquelles sont bonnes ou mauvaises est un talent au moins aussi important à avoir et c'est par celui-ci que Bertrand s'est le plus démarqué.

Benoît Leguay, le programmateur a apporté son savoir-faire lors de l'élaboration du programme. Il a débuggé des situations qui semblaient ne pas avoir de solution. Il a aussi assisté Hugo Massis dans le rôle de chef de projet en l'aidant à répartir les tâches.

Nous en parlions, le voilà: Hugo Massis, notre fameux chef de projet. Son talent relationnel qui s'est illustré par l'obtention des coordonnées du DG de Parrot ainsi que de l'aide d'élèves des années supérieures et sa capacité à nous faire travailler en temps en heure ont largement justifié son poste de chef.

Alexis Marhouss est la fouine d'internet, dont les recoins n'ont plus de secret pour lui. Il a glané quantité d'informations liées au projet et a notamment trouvé le logiciel Autoflight. Enfin Hugo Vaillaud est le programmateur... Quoi? Ce rôle est déjà occupé? Que nenni! Deux têtes valent mieux qu'une pour ce travail au centre de notre projet, et sa passion pour l'informatique nous a largement aidé pour la compréhension du langage Python par exemple.

## Etude de L'existant

Les drones sont apparus sur le marché il y a quelques années, et leurs nombres de ventes augmente chaque année. En effet, il existe tout type drones, certains sont télécommandés et d'autres suivent l'utilisateur, ils peuvent être équipés de caméra. De plus la gamme de prix des drones est très variable ce qui permet à une grande majorité de personne pour s'en procurer un. Nous avons donc sélectionné quatre exemple qui illustre tous les différents types de drone possible à trouver en magasin.

Le premier est le **SYMA X5C**, un drone pilotable uniquement par télécommande. Il dispose d'une caméra de 2 Mégapixel intégré, ce qui la rend fixe, avec un fil électrique qui sort du drone ce qui le rend très fragile. Il pèse environ 100

grammes ce qui est très faible, il est par conséquent facilement déstabilisé (bourrasque de vent par exemple) compliqué à manier. Il possède une autonomie d'environ 6 minute pour une heure de charge ce qui est très faible. Enfin, il peut voler jusqu'à 50 mètres de distance ce qui est correct.



Le deuxième drone est le **BEBOP 2** de l'entreprise française « Parrot ».

Contrairement au drone vu précédemment, celui-ci est pilotable depuis un smartphone jusqu'à 300 mètres de distance ou par télécommande jusqu'à 2 kilomètre environ. Son poids est de 500 grammes ce qui le rend assez instable. Il possède environ 25 minutes d'autonomie pour un prix variant entre 550 et 952 si on choisit de le prendre avec la télécommande. Il dispose d'une caméra intégrée de 14 Mégapixel, mais il possède un fisheye afin de filmer en grand angle ainsi stabilisateur d'image afin de la rendre plus nette.



Le drone **LILY** compte parmi les drones haut de gamme avec son prix de 800 euros. De plus, on peut le piloter par smartphone comme le Bebop 2 mais on peut aussi le paramétrier afin qu'il suive ou tourne autour d'une balise portée par l'utilisateur. Cette technologie étant couteuse en énergie, l'autonomie de vol de Lily ne dépasse pas les 20 minutes de vol. Contrairement à ses prédecesseurs, il est waterproof ce qui permet de l'utiliser par temps de pluie. Enfin, il possède une caméra intégrée de bonne qualité mais lors de bourrasques de vent il se peut que l'image ne soit pas très stable.



Pour finir, **L'HEXO+** pèse 1 kilogramme et coûte 1500 euros ce qui fait de lui l'un des drones les plus chers du marché. Il est pilotable par smartphone mais il peut aussi suivre l'utilisateur comme lily mais selon des schémas bien plus précis. Il pourrait facilement être utilisé dans le domaine du cinéma grâce au fait qu'il possède un cardan sur 3 dimensions afin de pouvoir orienter la caméra sur 360 degrés. Ce cardan lui permet de filmer l'utilisateur dans n'importe quelle situation et de stabiliser la caméra quel que soit la position et vitesse de l'appareil. Il a une autonomie de 15 minutes environ mais il possède l'avantage d'avoir une batterie remplaçable. Le prix de la deuxième batterie est de 150 euros.



Pour notre PSTE, l'école nous a d'abord fourni l'AR Drone 1.0 pour finalement nous prêter un modèle plus récent l'AR Drone 2.0. C'est un hélicoptère quadri rotor que l'on peut contrôler par smartphone via un réseau Wifi. Son autonomie est d'environ 12 min et ça vitesse maximal est de 18 Km/h. il possède une caméra directement intégrée ainsi qu'un micro-processeur fonctionnant sur le noyau LINUX donc sur les appareils Androïd et sous IOS. Cette spécificité est la plus importante car elle est nécessaire pour permettre l'aboutissement de notre pste.



## Planning

Nous vous avions présenté le planning prévisionnel lors du premier semestre, il nous a semblé judicieux dans celui-ci de vous présenter le planning réel de ce projet.



## Détermination du projet

Notre idée de projet a évolué au cours du premier semestre. A l'origine, nous voulions concevoir un drone innovant, permettant de suivre automatiquement un utilisateur tout en le filmant de manière stable à l'aide d'un cardan motorisé, inclinant la caméra en fonction des orientations du drone. L'idée était d'associer ces deux caractéristiques sur notre appareil afin d'apporter une réelle innovation dans le domaine. Malheureusement, nous nous sommes rendu compte que projet serait trop coûteux, et que du fait de sa complexité, nous n'avions aucune garantie de réussite. Nous nous sommes donc penchés sur l'idée de concevoir un programme permettant au drone de suivre son utilisateur par analyse visuelle, automatisant ainsi le vol de l'appareil. Ce programme a été testé sur l'AirDrone 2.0 de Parrot via un logiciel de contrôle AutoFlight. C'est donc via ce projet PSTE que nous avons conçu ce programme. Contrairement à l'idée de départ, le drone autonome n'est pas une innovation en soi. Néanmoins, notre programme (avec quelques modifications) peut être fonctionnelle sur beaucoup d'autres appareil et offre donc la possibilité de faire passer un drone de standard à autonome.

## Méthode de réalisation

Au cours du premier semestre, mon équipe et moi même avions clairement défini notre projet ainsi que le planning et la méthode de réalisation. Sachant que nous allions recevoir l'AirDrone 2.0 de Parrot, nous avons rapidement entamé des recherches sur cet appareil afin de trouver un moyen d'automatiser son pilotage. Ces recherches représentent une part très importante du projet. Elles nous permettent de réutiliser ce qui a déjà été fait par d'autres. Ainsi, l'acquisition de ces éléments favorise l'avancement et la qualité du projet. Suite à ces recherches nous avons découvert le logiciel AutoFlight, développé par Parrot, permettant de contrôler et de programmer le pilotage d'un AirDrone 2.0 via un ordinateur. Nous avons donc décidé d'utiliser ce logiciel pour notre projet car il offre une liaison simple entre l'ordinateur et le drone. En plus du logiciel, Parrot a aussi développé une librairie de fonctions, codées dans le langage Python, représentant des instructions de vol pour le drone. Grâce à cette interface et cette librairie, nous avons donc pu développer un code permettant à l'AirDrone 2.0 de suivre un utilisateur automatiquement par reconnaissance visuelle de QR CODE. Le principe est simple, le drone détecte un QR CODE dans le dos de l'utilisateur, analyse sa dimension et se positionne fonction de coordonnées du QR CODE sur l'écran... La réalisation du code fut la partie la plus intéressante du projet. Nous avons tous du fournir un travail important sur l'analyse des coordonnées du QR CODE détecté par le drone, ainsi que sur les fonctions donnant les instructions de vol à l'appareil. Une fois le programme terminé, nous avons réalisé tous les tests permettant de s'assurer de la fonctionnalité du drone. Ces tests nous ont permis de détecter et de résoudre les différentes erreurs du programme. Une fois les derniers tests concluant nous avons considéré notre projet comme achevé.

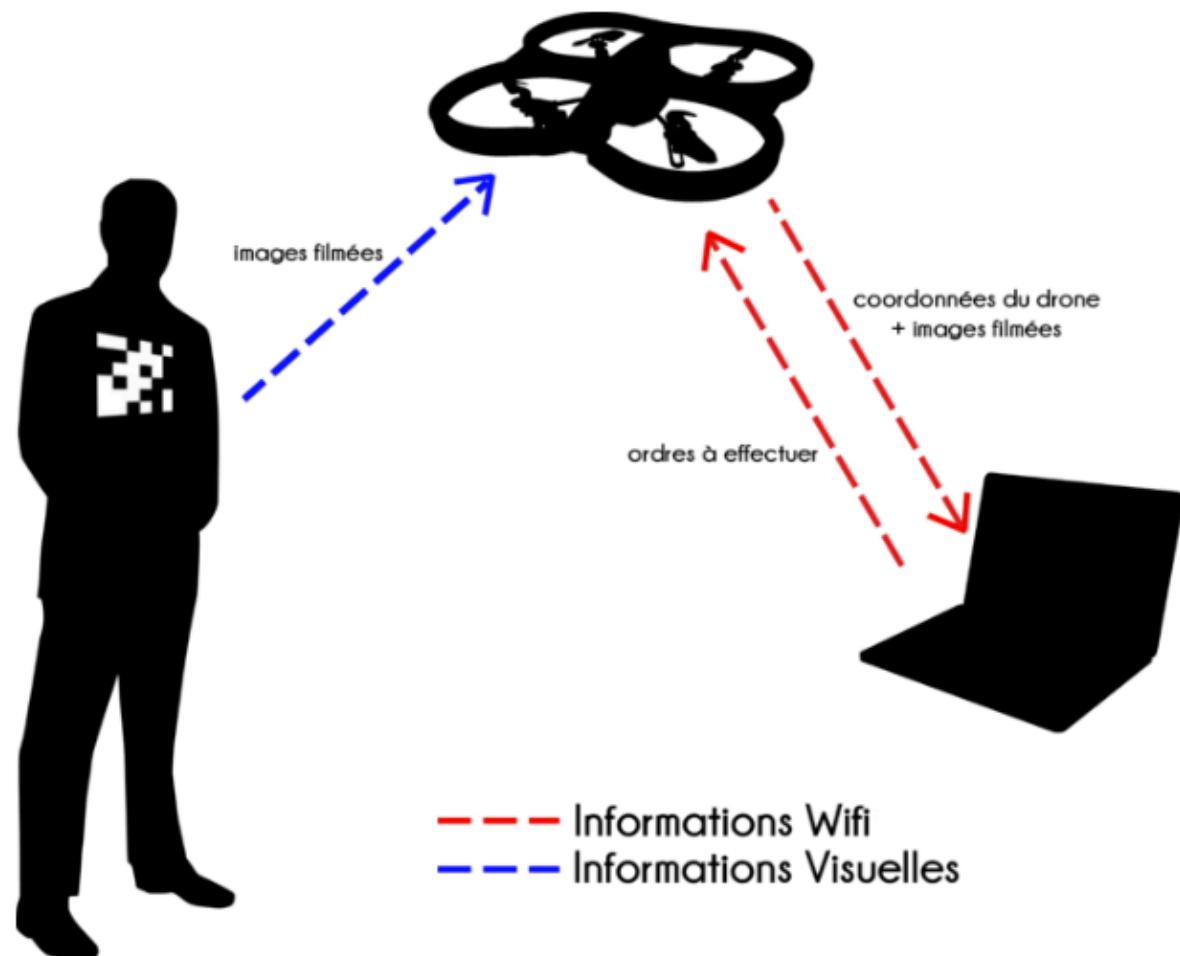
## Architecture générale et schématique

Notre projet est essentiellement constitué d'un programme sur ordinateur mais ce dernier permet au drone d'interagir automatiquement avec son entourage. Nous allons donc voir l'architecture détaillée de notre système au travers du schéma ci-dessous :

Comme indiqué, le drone échange diverses informations et ordres avec notre programme sur l'ordinateur, mais récupère aussi l'image filmée par la caméra frontale.

Notre programme commence par faire décoller le drone en lui envoyant un ordre. Une fois le drone stabilisé le programme a pour but de repérer un code QR à la caméra, d'analyser sa position et de déplacer le drone en conséquence.

Pour ce faire, le programme récupère la vidéo filmée par le drone et isole une image toute les 30ms. L'image est analysée au sein du programme sur l'ordinateur puis ce dernier envoie un ordre au drone. Le drone se charge de faire réagir ses moteurs en conséquence. Cette



opération est reproduite tant que le drone ne repère pas à l'écran le code QR correspondant à l'atterrissement. L'avantage de cette architecture est que l'analyse visuelle se fait sur l'ordinateur et permet donc d'économiser la batterie du drone et d'augmenter sa vitesse de réaction.

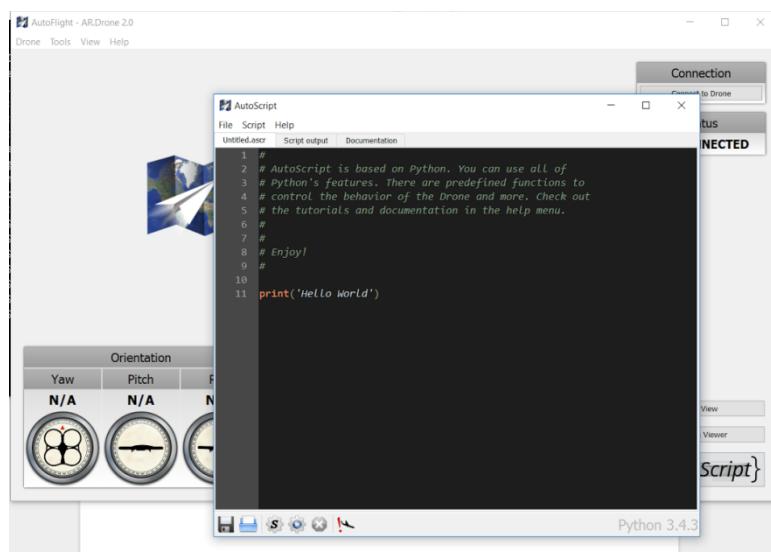
## Scenario des prototypes

Afin d'éviter au mieux que le drone fasse ce que bon lui semble et donc risquer de l'endommager, il a fallu faire fonctionner notre imagination afin d'éviter toute mauvaises surprises. C'est pourquoi nous établissons maintenant un scenario des prototypes, qui rassemblera les cas gérés par notre code.

Avant tout nous allons vous expliquer le fonctionnement de notre code pour vous situer les possibles problèmes rencontrées.

### Introduction à AutoScript.

Nous utilisons donc la console d'AutoFlight : AutoScript pour communiquer avec le drone, le langage est le python.



*La boîte de dialogue d'AutoFlight*

Lorsqu'un code est exécuté il agit sur le drone et nous avons la possibilité de voir une sortie console dans l'onglet à côté. Aussi si nous souhaitons juste consulter les sorties consoles sans déplacer le drone il est possible de simuler le code. Cette simulation nous montrera par ailleurs les déplacements que le drone aurait opérés.

Cette fenêtre contient donc le terminal, les sorties consoles et enfin une documentation de l'ensemble des fonctions pré-codé afin de simplifier le dialogue avec le drone. Au commencement nous avons donc testé ces fonctions comme par exemple :

The image shows two side-by-side screenshots of the AutoScript IDE. The left screenshot displays a script named 'Untitled.ascr' with the following Python code:

```

1 # 
2 # AutoScript is based on Python. You can use all of
3 # Python's features. There are predefined functions to
4 # control the behavior of the Drone and more. Check out
5 # the tutorials and documentation in the help menu.
6 #
7 #
8 # Enjoy!
9 #
10
11 import time
12 basicctl.takeoff()
13 time.sleep(10)
14 print("fini sleep")
15 basicctl.land()

```

The right screenshot shows the 'Script output' tab with the following log messages:

```

-----[SCRIPT SIMULATION STARTED]-----
[SIMULATION] Taking off
fini sleep
[SIMULATION] Landing
-----[SCRIPT SIMULATION STOPPED]-----

```

Both windows are running on Python 3.4.3.

Terminal

Sortie en

Nous avons donc pu bien comprendre comment utiliser AutoScript. Il nous restait à coder un code faisant suivre un utilisateur. En fouillant dans la documentation nous avons trouvé un sous-programme de détections d'April-tag (sorte de QR code). Nous avons donc utilisé ces codes pour réaliser notre projet.

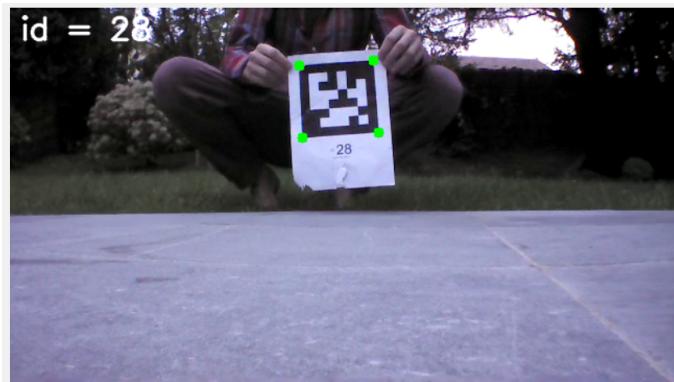


AprilTag de la famille 36h11 ayant l'id numéro 12

Figure 1: AprilTag 36h11 ID = 12

## Le principe du code de notre projet

Nous sommes donc partis d'un code qui détecte un April-Tag l'affiche sur le rendu vidéo du drone d'AutoFlight. Lorsqu'un April-Tag est détecté notre programme va stocker dans une liste, un nouveau tuple contenant lui-même une liste. Pour être plus clair, dès que le programme détecte un April-Tag, il va créer un tuple qu'il va insérer dans une liste appelé « détections ». Ce tuple contient, l'id du tag détecté, un booléen indiquant si le tag détecté est conforme, et enfin une liste possédant les coordonnées en x et en y des 4 sommets du Tag.



Nous voyons donc ici en vert que le programme a détecté les 4 coins du tag, aussi il a relevé l'indice du tag, le numéro 28 en haut à gauche

```
[([ 08, True, [(136.5301513671875, 87.84490203857422), (126.95362091064453, 100.2144546508789), (153.34365844726562, 98.9261474609375), (152.86793518066406, 87.03298950195312))])]
```

On voit ci-dessus l'affichage d'un tuple de détection, ici nous avons donc détecté un April Tag correcte (« True »), ayant l'id 08, aux coordonnées indiquées ((136,87),(126,100)...etc.). A noter aussi que l'axe des y est orienté vers le bas.

## Définitions des variables et variables globales

Il est nécessaire au départ du programme de définir des variables pour la simplification du code. Par exemple nous déclarons la hauteur max et la largeur max, correspondant au point en bas à droite sur l'écran. Aussi nous définissons les variables des coordonnées des 4 sommets du Tag à un instant t.

```
topLeftx=detections[0][2][3][0]
topLefty=detections[0][2][3][1]
topRightx=detections[0][2][2][0]
topRighty=detections[0][2][2][1]
downRightx=detections[0][2][1][0]
downRighty=detections[0][2][1][1]
downLeftx=detections[0][2][0][0]
downLefty=detections[0][2][0][1]
```

Ici nous avons la definitions des coordonnées des sommets de l'April tag

Pour la réalisation du projet il était aussi nécessaire de calculer le périmètre du carré nous avons donc des variables pour la hauteur, la largeur et évidemment le périmètre.

## Les mouvements du drone

### Avancer/Reculer

Pour avancer et reculer nous utilisons le périmètre de l'April tag à l'instant t que nous comparons à un variable globale appelé « périnéal ». Cette variable est le périmètre que le tag a à l'écran quand il est à la distance voulue.

```

53
54
55     if peri < periIdeal*90/100:
56         basicctl.move_rel(-0.2, 0, 0, 0)
57         print("avance")
58         time.sleep(0.1)
59
60     if peri > periIdeal*110/100:
61         basicctl.move_rel(0.2, 0, 0, 0)
62         time.sleep(0.1)
63         print("recule")
64

```

Nous avons donc ici les 2 conditions de déplacements, pour chacune d'entre on compare le périmètre actuel avec le périmètre souhaité si le périmètre est inférieur alors il faudra avancer pour augmenter celui et atteindre le periideal, et inversement. Aussi, afin d'éviter que le drone ne bouge alors qu'il est pratiquement dans la zone voulue, nous mettons un pourcentage de ce periideal.

### L'atterrissement

Concernant l'atterrissement, il suffit de montrer à la camera du drone l'April tag ayant l'id numéro 29. En le reconnaissant, le programme va afficher en console qu'il l'a vu puis il mettre la condition de sortie de la boucle « while » (boucle qui gère les déplacements tant qu'on ne lui demande pas d'arrêter) à « true ».

```

24
25     while cond !=1:
26         basicctl.move_rel(0, 0, 0, 0)
27         detections = imgproc.tag_detections()
28         img = imgproc.latest_frame()
29         i = 0
30         for (id, good, points) in detections:
31             cv2.putText(img, 'id = ' + str(id), (10, 30 +
32                 for (x, y) in points:
33                     cv2.circle(img, (int(x), int(y)), 5, (0, 2
34                     i+=1
35
36             if detections[0][0]==29:
37                 print("executez l'ordre 66")
38                 cond=1
39

```

## Les déplacements de cotés

Nous avons aussi prévu des mouvements de translations horizontaux et verticaux. Pour cela, nous avons divisé l'écran en 9. Lorsque les quatre sommets de l'April Tag sont dans les deux tiers droit de l'écran, le drone se déplace vers la droite. (Pour le moment) lorsqu'il est dans les deux tiers supérieur, il se déplace vers le haut et ainsi de suite. Ensuite, On fait mesurer à l'ordinateur la distance qui sépare le sommet en haut à gauche et celui en bas à gauche, et pareil pour les sommets de droite. On compare ces deux distances. Si une des deux est plus grande que l'autre d'un facteur 1,1, alors le drone se déplacera horizontalement en conséquence

```

if (((downLeftx-topLeftx)*150/100) > (downRightx-topRightx)):
    #print("translation droite")

if (((downRightx-topRightx)*150/100) > (downLeftx-topLeftx)):
    #print("translation gauche")

if ((downLeftx > x2t) and (downRightx > x2t) and (topRightx > x2t) and (topLeftx > x2t)):
    print("translation droite")
    basicctl.move_rel(0, 0.1, 0, 0)

if ((downLeftx < x1t) and (downRightx < x1t) and (topRightx < x1t) and (topLeftx < x1t)):
    print("translation gauche")
    basicctl.move_rel(0, -0.1, 0, 0)

```

## Difficultés rencontrés sur Autoscript

Nous avons rencontré plusieurs problèmes pendant ce PSTE. Nous écarterons ici ceux liés au drone 1.0 car c'est celui dont nous disposons actuellement qui nous intéresse. Le logiciel que nous utilisons pour faire en sorte que le drone nous suive est en bêta et conçu par des amateurs. Il comporte donc des défauts qui nous ont ralenti pendant nos phases de test. Tout d'abord, impossible de sauvegarder notre code, que nous étions contraints de copier, sauvegarder sur un fichier .txt et recopier à chaque fois que nous relancions le logiciel. Ajoutez à cela le fait que le logiciel se ferme de manière totalement aléatoire de temps à autre (sans compter les freeze) et l'expérience des tests devient vite pénible, notre travail partant en fumée régulièrement. Nous avons eu un peu de mal à nous familiariser avec le langage de programmation Python qui comprend des règles très différentes des langages que nous connaissons. De plus, la compilation d'Auto flight est assez exigeante: En effet, l'utilisation de la touche de tabulation pour indenter le code rend sa lecture impossible, il fallait nécessairement appuyer 4 fois sur espace pour chaque indentation. Les erreurs de code (aussi insignifiantes soient-elles) nous étaient signalées par d'autres erreurs inventées par le compilateur, nous ralentissant beaucoup. Ensuite, comprendre le mode de fonctionnement des fonctions d'AutoFlight n'a pas été particulièrement simple. Si se familiariser avec le langage n'était déjà pas évident l'appliquer directement à des concepts tels que les tuples l'était encore moins. Enfin, nous avons connu plusieurs difficultés liées à la détection des April Tag. Nous avons mis du temps à comprendre qu'il faut préciser à quelle famille appartient notre April Tag. Par conséquent, notre programme détectait beaucoup de choses (chaises, fenêtres, murs) mais pas nos April Tag. Nous avons eu par la suite eu l'idée de blinder la détection pour que le drone réagisse uniquement à un April Tag précis.

## Difficultés et solutions apportés

Au cours de cette année, nous avons dû faire face à de nombreuses difficultés et d'événements inattendus qui nous ont ralenti dans l'avancement du projet. La recherche de solutions vis à vis de ces problèmes fut un enjeu majeur. En effet, on ne pouvait pas se permettre une accumulation de retard vis à vis du programme qui avait été fixé. Nous avons donc toujours essayé d'être le plus réactifs face aux situations inattendues. Au cours du second semestre, nous avons rencontré un problème au moment de la livraison du drone. En effet, au lieu de recevoir un AirDrone 2.0 nous avons reçu l'AirDrone 1.0. De ce fait, l'utilisation d'Autoflight ainsi que des fonctions de la librairie, permettant de donner les instructions de vol au drone, était impossible. Nous avons donc perdu un temps précieux à travailler sur la liaison entre le drone et l'ordinateur, ainsi que sur les fonctions de déplacements relatifs à l'AirDrone 1.0. Forte heureusement notre mentor madame Rendler à accepter de nous changer notre drone. Nous avons donc pu recommencer à travailler sur AutoFlight. Le second problème majeur est survenu au moment de la réalisation du programme sur AutoFlight. Ne maîtrisant pas le langage de programmation imposé par le logiciel nous avons eu du mal à implémenter notre algorithme. De plus, nous avons pris détection de nombreuses erreurs de logiques dans le code au moment de la réalisation des différents tests. Nous avons donc consacré un temps important à leurs résolutions afin d'obtenir un résultat satisfaisant.

## Valorisation

Comme nous l'avons vu dans le cahier des charges, il existe déjà sur le marché des drones capables d'effectuer un déplacement automatique en suivant un utilisateur. Cependant leur coût est extrêmement élevé pour la plupart et cette fonctionnalité reste donc très inaccessible.

Notre PSTE cherche donc à y remédier en proposant une solution gratuite, simple d'installation et compatible avec tous les Parrot AR Drone 2.0, un des drones les plus rependus à ce jour et ayant un prix relativement abordable.

Contrairement aux autres drones sur le marché, le système que nous avons créé n'utilise aucun module GPS ou autre mode de géolocalisation pour suivre l'utilisateur. Ces dispositifs étant coûteux nous avons préféré passer par une analyse visuelle, l'AR Drone 2.0 de Parrot possédant déjà une caméra, aucune modification n'a donc été apportée pour filmer. Une analyse visuelle permet bien plus de précision et de liberté de mouvement. De plus l'analyse de l'image se fait sur l'ordinateur, ce qui permet un traitement plus rapide et moins coûteux en batterie : le drone ne perd pas en autonomie et est plus réactif.

Pour que son drone puisse le suivre, l'utilisateur n'aura qu'à imprimer un QR Code, télécharger le logiciel AutoFlight gratuitement et y ajouter notre programme. Les QR codes sont disponibles sur internet gratuitement tout comme le logiciel Autoflight, ce qui rend la mise en place du système très rapide et accessible à tous.

## Bilan

### Collectif

A l'approche de notre soutenance nous avons enfin atteint les objectifs que nous avions fixés, mais pas sans le moindre aléa. Ce projet nous a beaucoup intéressé dès le début car l'idée était la nôtre et c'est pourquoi nous étions si motivés à réussir. Bien que les membres de l'équipe se connaissaient déjà avant nous avons dû prendre un certain temps avant de trouver une organisation productive. En effet comme nous ne sommes pas dans le même TD, les diverses réunions devaient pouvoir satisfaire l'emploi du temps de tout le monde. De plus le PSTE est un projet bien plus conséquent que tous ceux imposés dans l'année, l'organisation devait donc être un des maîtres mots. Et comme un projet de cette ampleur n'est pas sans embûche nous avons dû trouver des solutions à de nombreux problèmes. Tout d'abord la définition du sujet et notre problématique : trop ambitieuse pour notre niveau scolaire. Mais même une fois les objectifs clarifiés nous avons dû faire face à de nombreux contretemps. Comme expliqué précédemment nous avons reçu notre commande très tardivement, et même une fois reçue le modèle n'était pas le bon. Bien que décourageante, cette épreuve nous a appris à rester motivés afin de recommencer nos recherches sur le modèle reçu. Cette réorientation a ensuite pris fin avec un échange de drone et nous avons donc pu continuer notre travail sur le modèle initialement prévu. Notre travail d'équipe et notre motivation ont finalement permis de surmonter cette perte de temps et d'atteindre nos objectifs : un drone qui suit l'utilisateur sans besoin de le contrôler. Somme toute, le projet choisi par notre équipe cette année nous a beaucoup apporté, autant d'un point de vue technique avec l'apprentissage d'un nouveau langage de code et la prise en main du logiciel AutoFlight, que d'un point de vue humain avec le travail en équipe et la motivation. Nous sommes tous d'accord pour dire que RecBird est le projet le plus enrichissant que nous ayons fait à l'ECE à ce jour.

### Individuel

#### Hugo Massis

Ce fut pour moi une aventure extrêmement intéressante que de mener à bien ce projet en tant que chef de projet, et je pense avoir appris beaucoup. Tout d'abord, j'ai eu la chance de travailler sur un projet qui m'intéressait avec des personnes que j'apprécie.

Ce PSTE nécessitait une réelle autonomie ainsi qu'un grand esprit d'adaptation et je pense que mon équipe et moi-même ont su relever ce défi.

Je pense également que ce projet m'a démontré l'importance d'une organisation solide.

Je suis aujourd’hui fier du travail que nous avons mené au cours de cette année, et ceux, malgré les différents aléas que nous avons rencontrés.

Alexis Mahrouss

Ce projet m'a permis de me responsabiliser car il faut travailler régulièrement afin de ne pas prendre retard sur le planning prévisionnel. Des créneaux horaires sont libérés afin que nous puissions travailler le PSTE mais personnes ne nous notes absent si nous venons pas à l'ECE pour travailler dessus. Il faut donc se prendre en main afin de ne pas toujours repousser les réunions avec notre groupe en disant « on verra la semaine prochaine ». De plus, notre projet contient une partie de codage en langage python, ce qui m'a permis d'apprendre les bases d'un nouveau langage que nous n'avons pas étudier pendant mes deux premières années à l'ECE. Ce fut une réelle difficulté car apprendre un nouveau langage nécessite du temps et de la pratique mais au final, cela reste un point positif. Enfin, comme ce projet est à réaliser en équipe, pour bien avancer il faut être à l'écoute des autres et respecter les idées des chacun, ce qui peut paraître difficile au premier abord mais cela vient avec le temps et la succession de projets. Pour finir, ce projet m'a beaucoup apporté tout au long de l'année et j'en tire une note positive.

Hugo Vaillaud

J'ai beaucoup apprécié ce projet que j'ai pu faire avec des personnes avec qui je m'entends bien et qui travaillent avec sérieux. plus orienté sur la programmation que mon ancien projet, j'ai bien évidemment pris plus de plaisir à contribuer à la réussite de ce projet. Le thème est dans l'ère du temps, et pouvoir se dire qu'on a réussi à faire en sorte qu'un drone nous suive uniquement par reconnaissance d'image est quelque chose de motivant. Je n'ai pas ressenti qu'un de nous se la coulait donc pendant que les autres travaillaient, nous faisions nos recherches tous ensemble. Une telle cohésion de groupe ne peut qu'être agréable.

Le seul point noir a été le problème de changement de modèle de drone qui a finalement été rectifié.

Merci à mon équipe donc, et à Mme Rendler qui nous a encadré cette année.

Thibault Copart

RecBird a été à mes yeux le projet le plus intéressant depuis mon arrivée à l'ECE. Riche en péripéties j'ai su rester motivé afin de toujours aller de l'avant même quand nos objectifs semblaient impossibles. De plus le sujet me tenait vraiment à cœur, j'ai donc beaucoup apprécié m'investir. Enfin, étant le projet le plus important de l'année, le PSTE nécessite de la rigueur et m'a donc beaucoup apporté en terme d'organisation et de travail en équipe.

Bertrand Glaser

D'un point de vu personnel, ce projet a été très enrichissant pour moi et de bien des façons. Tout d'abord, travailler durant une année sur le développement d'un produit technique relativement complexe m'a donné une idée de ce qu'on me demandera plus tard lors de

mon futur parcours professionnel. Ce projet a sollicité mes capacités de réflexion et de raisonnement, mais aussi de rigueur et d'organisation. En plus, mes recherches ont permis d'améliorer mes connaissances dans certaines matières. Par exemple, programmer les déplacements du drone sur AutoFlight m'a permis de découvrir le python, langage que je n'avait jamais vu auparavant. J'ai pris beaucoup de plaisir à faire ce projet. C'était l'occasion pour moi l'occasion de développer un produit qui me plaisait dans une bonne ambiance de groupe.

### Benoit Leguay

Ce projet [REC] Bird a été pour moi une grande source d'expérience sur plusieurs domaines. D'une part, la réalisation du programme de déplacement dynamique du drone fut une tâche complexe surtout du à l'apprentissage d'un nouveau langage de programmation le python. Mais aussi d'une application d'un code bien différent de ce que l'on peut avoir à l'ECE. En effet, voir les fruits de notre programme s'opérer sur le plan de vol d'un drone était très amusant et surtout motivant. Aussi, le fait d'utiliser des nouvelles librairies m'a beaucoup appris sur le monde de la programmation et des logiciels libres.

D'autre part, au cours du 2<sup>ème</sup> semestre, nous avons été confrontés à des problèmes de délais, du à la livraison en retard et du mauvais drone, j'ai pu donc comprendre que ce genre de chose était très fréquente dans les projets quel qu'ils soient.

Enfin, j'ai trouvé qu'il y avait une très bonne répartition des tâches au sein de notre groupe, et cela fut très appréciable. Chacun savait ce qu'il devait faire et apporter sa pierre sans gêner le travail de l'autre. De ce fait il n'y avait pas de temps mort au cours de nos réunions. Je pense avoir gagné en team-management ce qui me paraît essentiel pour ma future vie professionnelle.

## Sources

Le SDK de L'ArDrone : <http://developer.parrot.com/ar-drone.html>

Documentation du logiciel AutoFlight :  
<http://electronics.kitchen/docs/autoflight/gettingstarted.html>

Site Python : <https://www.python.org/>

OpenClassRoom <https://openclassrooms.com/courses/apprenez-a-programmer-en-python>