Drone programmer Documentation :

0  Prérequis :

Afin de travailler sur cette application il est nécessaire d’être équipé comme suit :

* Un ordinateur avec MacOS
* xCode
* Un drone Parrot bepob 2
* Cloner le repository github
* Vérifier que [ces champs](https://developer.parrot.com/docs/SDK3/#ios) soient remplis correctement

1 But de l’application :

Le but de l’application droneProgrammer est de pouvoir piloter un drone bepop2 de la marque Parrot.

Il y’a plusieurs moyens de piloter le drone :

1. Un pilotage *live* où le drone réagit au divers bouton pressé. Ce mode sera appelé « Free flight » dans le reste de cette documentation ainsi que dans le code de l’application.
2. Un pilotage programmer. Dans ce mode il est possible de programmer le plan de vol du drone. Dans ce mode, il faut entrer une suite de commande, éventuellement des obstacles. Il est ensuite possible de simuler le vol afin de vérifier que le drone ne touchera aucun obstacle avant de le lancer.

2  Fonctionnement globale de l’application :

1. Photo storyboard
2. Le chemine de l’application est le suivant : On commence par une vue initial (qui n’est pas codée littéralement, apparait que dans le storyboard) . C’est un menu où il y’a trois options.
   1. « Free flight »
   2. « Gestionnaire » Depuis ce gestionnaire on a une liste de plan de vol sauvegardé. Lorsqu’on choisit choisit un plan de vol on arrive directement au mode créateur avec ce plan de vol
   3. « Plan de Vol » qui correspond à un mode création de plan de vol. Il possible d’ajouter les commandes suivantes :
      * Décoller
      * Atterrir
      * Tourner à droite
      * Tourer à Gauche
      * Avancer
      * Reculer
      * Monter
      * Descendre

Il également possible d’ajouter des obstacles qui sont représentée en coordonnées x,y et z, de même que des objectifs à placer. Une fois que le plan de vol est créé il est possible de le simuler. Si la simulation réussie (Tous les objectifs touchés et aucun obstacles percutés, il est alors possible de lancer

3 Les technologies utilisées :

* Swift -> Langage de programmation de l’application
* Json -> Encodage des plans de vols
* SceneKit -> Librairie 3D pour la simulation
* Objective-C -> Langage de programmation pour les *samples* du drone.

Il est conseillé d’avoir de bonnes connaissances dans les trois premières technologies, la dernière n’est que peu utilisée. Cependant, pour aller plus loin, il est possible d’utiliser directement l’API du drone, qui est écrite en objective-C et qui permet d’aller plus loin que les *samples*.

4  Liens divers :

* [Documentation librairire bebop 2](https://developer.parrot.com/docs/reference/bebop_2/index.html#bebop-2-reference)
* [Utilisation des samples](https://developer.parrot.com/docs/SDK3/#use-samples)
* [Json documentation](https://www.w3schools.com/js/js_json_intro.asp)
* [Swift documentation](https://swift.org/documentation/)
* [SceneKit documentation](https://developer.apple.com/documentation/scenekit)

5 Contacts :

Professeur : Didier Buchs, [didier.buchs@unige.ch](mailto:didier.buchs@unige.ch)

Assistants : Damien Morard, [damien.morard@unige.ch](mailto:damien.morard@unige.ch)

Dimitri Raccordon, [dimitri.raccordon@unige.ch](mailto:dimitri.raccordon@unige.ch)

Étudiants : Mathias Tonini, [mathias.tonini@etu.unige.ch](mailto:mathias.tonini@etu.unige.ch), [mathias.tonini@gmail.com](mailto:mathias.tonini@gmail.com)

Deniz Sungurtekin, [deniz.sungurtekin@etu.unige.ch](mailto:deniz.sungurtekin@etu.unige.ch), [denizsungurtekin@gmail.com](mailto:denizsungurtekin@gmail.com)