

## **Procédures de tests fonctionnels - CDR**

**R.F.1 Chaque drone physique doit individuellement répondre à la commande "Identifier" disponible dans l'interface utilisateur. Lors de cette commande, au moins une des DELs du drone doit clignoter ou changer de couleur.**

1. Envoie du micrologiciel Crazyfly sur les robots aériens;
2. Lancement des conteneurs Docker client (UI) et server;
3. Appuyer sur 'Identifier' correspondant aux robots à identifier;
4. Vérification du comportement attendu en regardant les DELs du robots à identifier.

**R.F.2 L'essaim de drones doit répondre aux commandes suivantes, disponibles sur l'interface utilisateur : "Lancer la mission" : décollage et début de la mission et "Terminer la mission" : atterrissage immédiat.**

### **Robots aériens:**

1. Envoie du micrologiciel Crazyfly sur les robots aériens;
2. Lancement des conteneurs Docker client (UI) et server;
3. Appuyer sur 'Lancer Mission';
4. Vérification que les robots sont en mesure de s'envoler;
5. Appuyer sur 'Terminer Mission';
6. Vérification que les robots sont en mesure d'atterrir.

### **Simulation:**

1. Lancement des conteneurs Docker client (UI), server et simulation;
2. Appuyer sur 'Start' dans la simulation;
3. Appuyer sur 'Lancer Mission';
4. Vérification que les robots sont en mesure de s'envoler et vérification de l'état;
5. Appuyer sur 'Terminer Mission';
6. Vérification que les robots sont en mesure d'atterrir et vérification de l'état.

**R.F.4 Après le décollage, les drones doivent explorer l'environnement de façon autonome. L'algorithme est à la discrétion du contractant. Cela peut être une séquence de mouvements aléatoires (random walk).**

et

**R.F.5 Les drones doivent éviter les obstacles détectés par leurs capteurs (objets statiques ou autres drones sans distinction).**

**Robots aériens:**

1. Même procédure que celle de R.F.2.
2. Analyser le comportement physique des robots dans la volière par rapport au mur et en ajoutant certains obstacles à la volière. Les robots devraient être en mesure de naviguer à travers la volière sans toucher aux murs, aux obstacles et à l'autre robot aérien.

**Simulation:**

1. Même procédure que celle de R.F.2.
2. Analyser le comportement des robots de simulation en vérifiant que ceux-ci ne touchent à aucun obstacle et autres robots.

**R.F.10 L'interface utilisateur pour l'opérateur doit être disponible comme service Web et visualisable sur plusieurs types d'appareils (PC, tablette, téléphone) via réseau. Au moins deux appareils doivent pouvoir être connectés en même temps pour la visualisation des données lors d'une mission. Le contrôle peut être restreint (ou non) à un seul appareil.**

1. Tester l'affichage de l'interface utilisateur sur différentes plateformes en utilisant la fonction développeur de Chrome. Il est possible de changer le format de l'affichage pour les tablettes et téléphone
2. Pour tester plus d'un client, nous testons avec l'ouverture d'un autre client dans une deuxième instance d'un fureteur Web comme Chrome, et vérifions que les actions de premier client s'appliquent sur le deuxième sans interférence.

Le client de la station au sol sera chargé de contrôler les robots aériens et la simulation. Un client sur un poste différent ne peut que visualiser l'exploration.

**R.C.3** L'environnement virtuel pour les tests dans ARGoS doit pouvoir être généré aléatoirement (peut être fait simplement dans le fichier de configuration du simulateur). L'environnement simulé doit avoir un minimum de 3 murs (sans compter les 4 murs extérieurs).

1. Les cartes générées aléatoirement se créent automatiquement à l'ouverture de la plateforme ARGoS grâce au *tag distribute* dans le code XML. Un tag est utilisé pour un obstacle.