

Hivexplore



Essaim de drones explorateurs

Présenté par Misha K.-R., Nathanaël B.-D., Rose B., Samer M., Simon G. et
Yasmine M.

19 avril 2021

Évolution des robots explorateurs

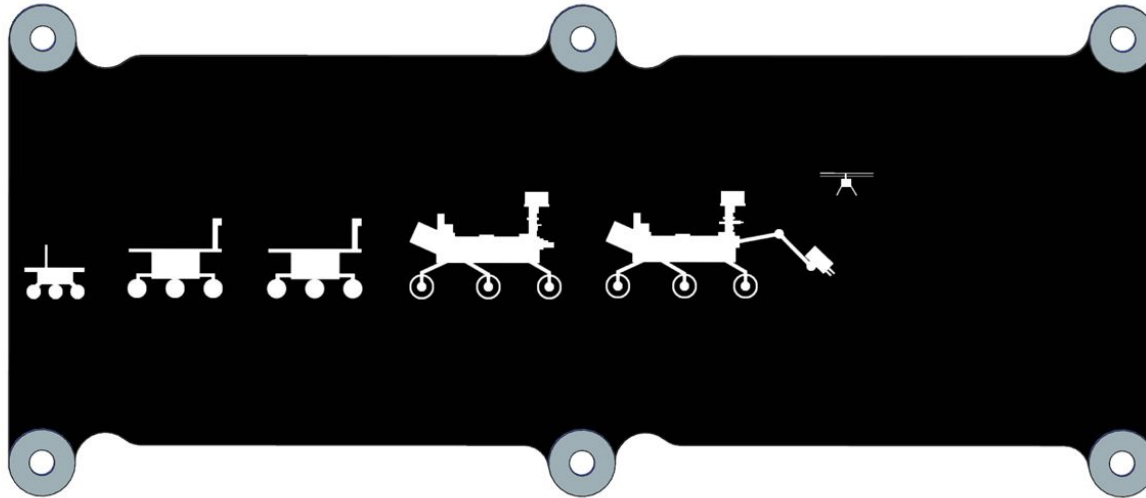


Figure 1. Rover Evolution Plate Artwork [plaque d'évolution des robots]. NASA JPL-Caltech, 2020. Image du domaine public.

Présentatrice: Yasmine M.

Robot *Perseverance*

- Grand et lourd
- Très outillé
- Non axé sur la mobilité
- Assez lent pour la couverture de grands terrains

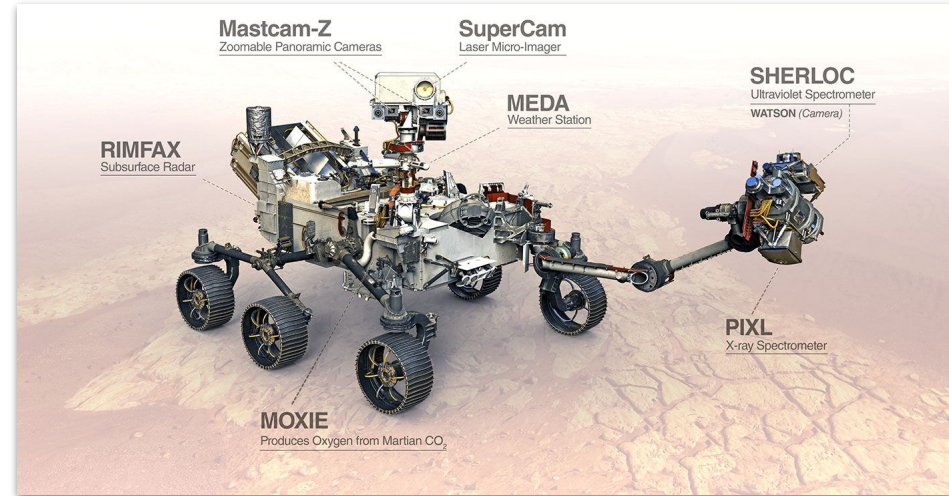


Figure 2. Perseverance. NASA, 2020. Image du domaine public.

Présentatrice: Yasmine M.



Essaim de drones explorateurs

- Drones rudimentaires
- Exploration et cartographie des lieux
- Contrôle par une station au sol
- Interface utilisateur
 - Envoi de commandes (exploration, retour à la base)
 - Visualisation de la carte en temps réel
 - Surveillance des drones



Figure 3. Drone. Jeshoots, 2017. Image du domaine public

Présentatrice: Yasmine M.



Plan de la présentation

1. Description du système

- 1.1 Architecture logicielle générale
- 1.2 Fonctionnement du client Web
- 1.3 Fonctionnement du serveur
- 1.4 Fonctionnement des drones

2. Démonstrations

- 2.1 Démonstration avec ARGoS
- 2.2 Démonstration avec les drones

3. Algorithmes

- 3.1 Évitement d'obstacles
- 3.2 Évitement de drones
- 3.3 Exploration
- 3.4 Retour à la base

4. Tests et résultats

- 4.1 Tests et résultats
- 4.2 Points saillants de notre solution

5. Gestion de projet

- 5.1 Réunions hebdomadaires
- 5.2 Gestion du développement logiciel
- 5.3 Coordination hors des réunions
- 5.4 Rôles de l'équipe

Présentatrice: Yasmine M.

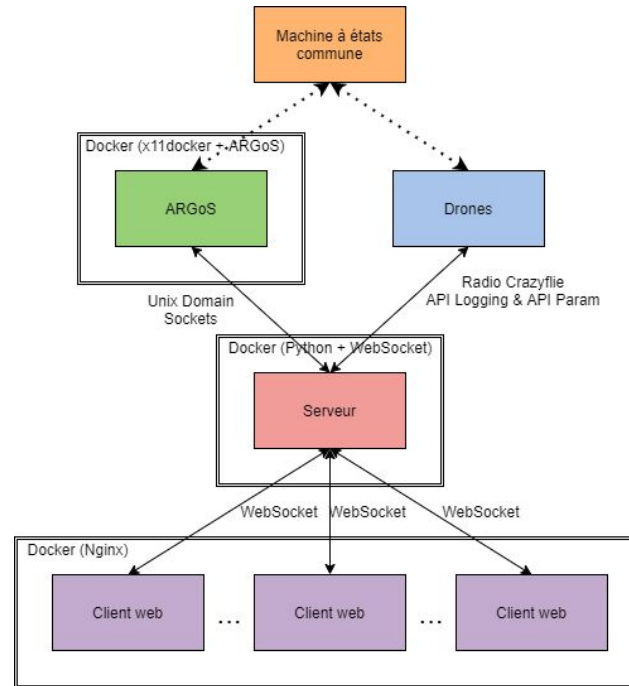


1. Description du système



1.1 Architecture logicielle générale

- Client
 - Interface pour les utilisateurs
- Serveur
 - Intermédiaire entre les clients Web et les drones
 - Abstraction entre la simulation et les *Crazyflies*
- ARGoS
 - Simulation des drones
- Drones
 - *Crazyflies* de Bitcraze



Présentateur: Simon G.

Figure 4. Architecture logicielle générale



1.2 Fonctionnement du client Web

- Contrôle de la mission
 - Départ de la mission
 - Atterrissage d'urgence
 - Retour à la base
 - Fin de la mission
- Nombre de drones
- État de la mission

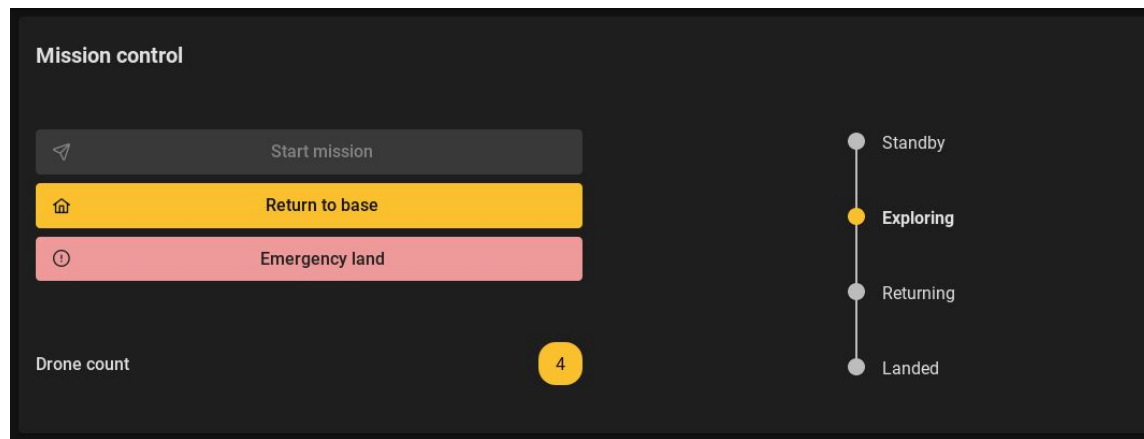


Figure 5. Interface du contrôle de la mission

1.2 Fonctionnement du client Web (suite)

- Visualisation des données des drones
 - Vitesse
 - Batterie
 - État
 - DEL

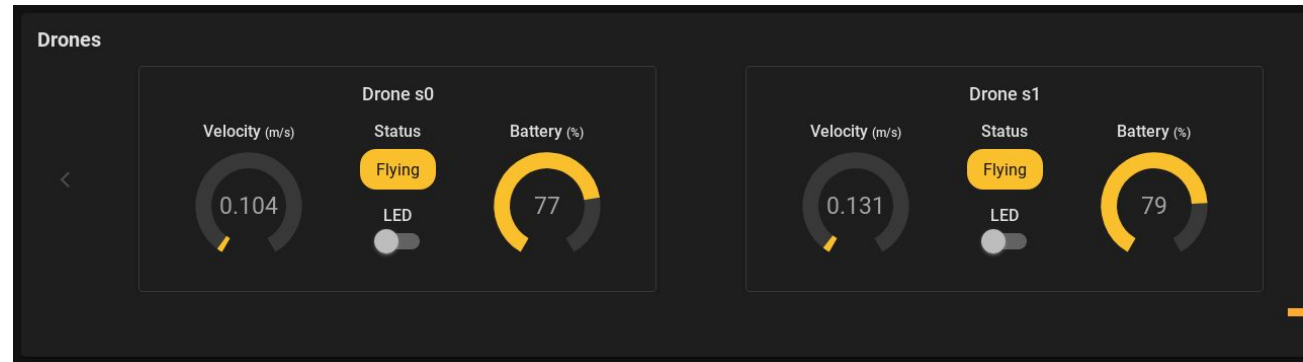


Figure 6. Interface de visualisation des données des drones

1.2 Fonctionnement du client Web (suite)

- Visualisation de la carte générée par les drones
 - Carte 3D (*three.js*)
 - Points de la pièce
 - Positions des drones

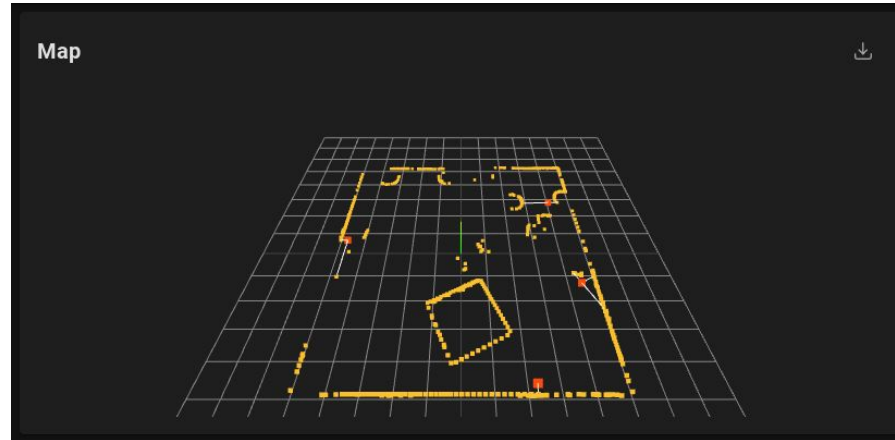
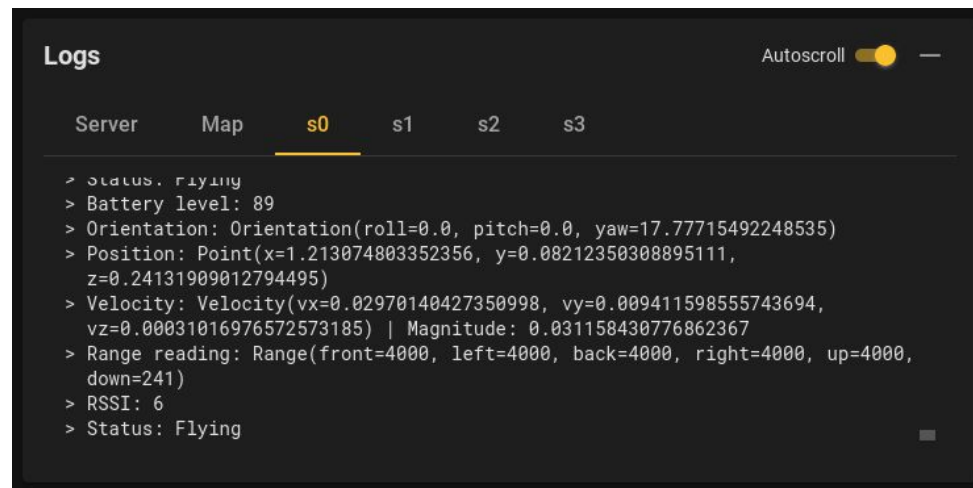


Figure 7. Carte générée par les drones

Présentateur: Simon G.

1.2 Fonctionnement du client Web (suite)

- Lecture des journaux
 - Regroupements logiques
 - Journaux utiles au débogage



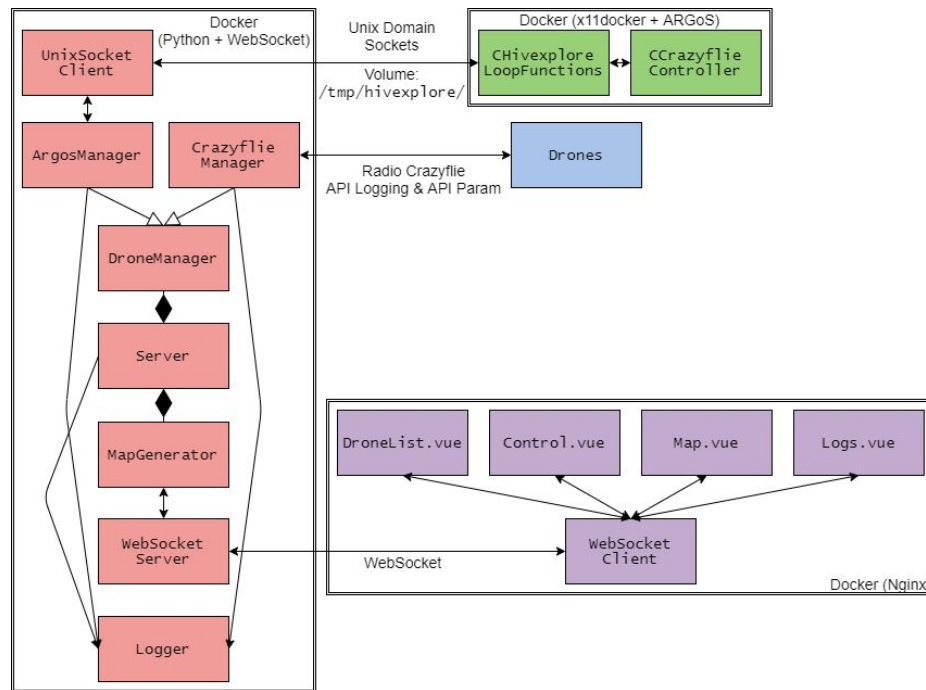
```
Logs Autoscroll ☒
```

Server	Map	s0	s1	s2	s3
<pre>> status: flying > Battery level: 89 > Orientation: Orientation(roll=0.0, pitch=0.0, yaw=17.77715492248535) > Position: Point(x=1.213074803352356, y=0.08212350308895111, z=0.24131909012794495) > Velocity: Velocity(vx=0.02970140427350998, vy=0.009411598555743694, vz=0.00031016976572573185) Magnitude: 0.031158430776862367 > Range reading: Range(front=4000, left=4000, back=4000, right=4000, up=4000, down=241) > RSSI: 6 > Status: Flying</pre>					

Figure 8. Visualisation des journaux

1.3 Fonctionnement du serveur

- *Asyncio*
- Générateur de carte
- Abstraction des drones et de la simulation
- *WebSocket*
- *Sockets Unix*
- *Logger*



Présentateur: Simon G.

Figure 9. Diagramme de l'architecture logicielle de la station au sol



1.4 Fonctionnement des drones

- *Crazyflie 2.1* de *Bitcraze*
 - Communication: *Crazyradio PA*
 - Capteurs: *Multi-ranger* et *Flow deck*
- Utilisation de l'API de *Bitcraze* pour programmer les drones
- Utilisation d'ARGoS pour simuler le comportement des drones et les lieux à explorer



Figure 10. Crazyflie. Bjorn Mauritz, 2017. CC BY 3.0

1.4 Fonctionnement des drones (suite)

Tableau 1. Comparaison entre les drones *Crazyflies* et la simulation *ARGoS*

Caractéristiques	<i>Crazyfly</i>	<i>ARGoS</i>
Réception de données (fréquence de 1 Hz)	Cadriciel d'acquisition de données	<i>Sockets Unix</i>
Envoi de commandes	Cadriciel de paramétrisation	<i>Sockets Unix</i>
Logique de contrôle	Logique basée sur les vitesses	Logique basée sur les positions
Langage de programmation	C	C++

Présentatrice: Rose B.



1.4 Fonctionnement des drones (suite)

- Cadriciel d'acquisition de données
 - Niveau de la batterie
 - Orientation
 - Position
 - Vitesse
 - Distances mesurées par les capteurs
 - RSSI
 - État des drones
- Cadriciel de paramétrisation
 - État de la mission
 - Contrôle de la DEL
 - Distance entre la position initiale et la station au sol

Présentatrice: Rose B.



2.1 Démonstration avec *ARGoS*

(voir le partage d'écran de Misha K.-R.)

Présentateur: Misha K.-R.



2.2 Démonstration avec les drones

(voir la caméra vidéo de Samer M.)

Présentateur: Samer M.



3.1 Évitement d'obstacles

- Correction de la vitesse considérant les quatre capteurs de côté
- Logique appliquée en arrière-plan lorsque le drone est en mouvement
- Correction proportionnelle à la proximité de l'obstacle
- Robustesse en situations complexes (ex : passage étroit, changements dans l'environnement, etc.)

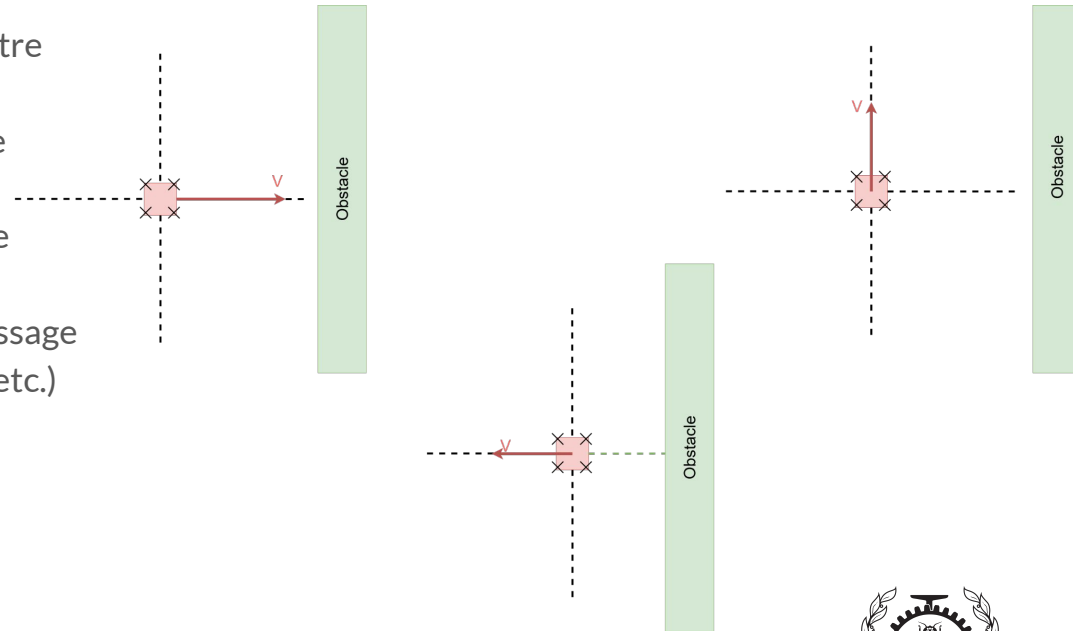


Figure 11. Algorithme d'évitement d'obstacles

Présentateur: Samer M.



3.2 Évitement de drones

- Algorithme nécessitant la position relative à un point de référence commun
- Calcul du vecteur de correction dans la direction opposée aux autres drones
- Correction considérant tous les drones à proximité
- Logique appliquée en arrière-plan lorsque le drone est en mouvement
- Logique rarement nécessaire en mission grâce à l'efficacité de l'exploration

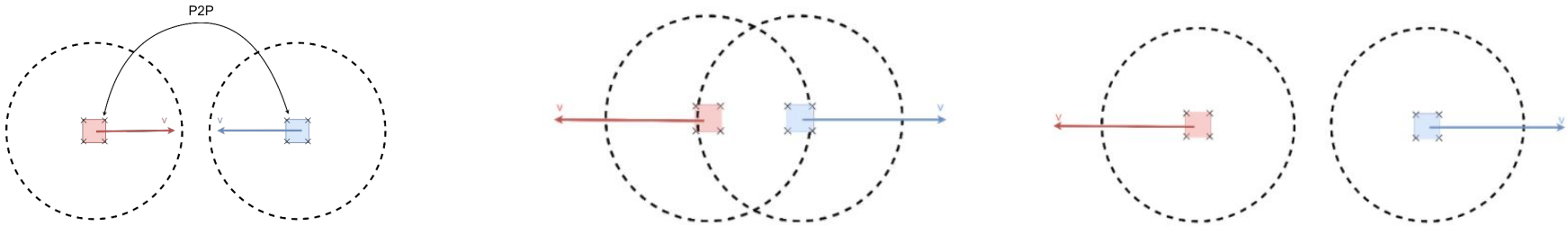


Figure 12. Algorithme d'évitement de drones

Présentateur: Samer M.

3.3 Exploration

- Avancement
- Alternance de la direction de rotation
- Réorientation dans la direction opposée au centre de masse
 - Obtention des positions des autres drones par communication pair-à-pair
 - Calcul du centre de masse

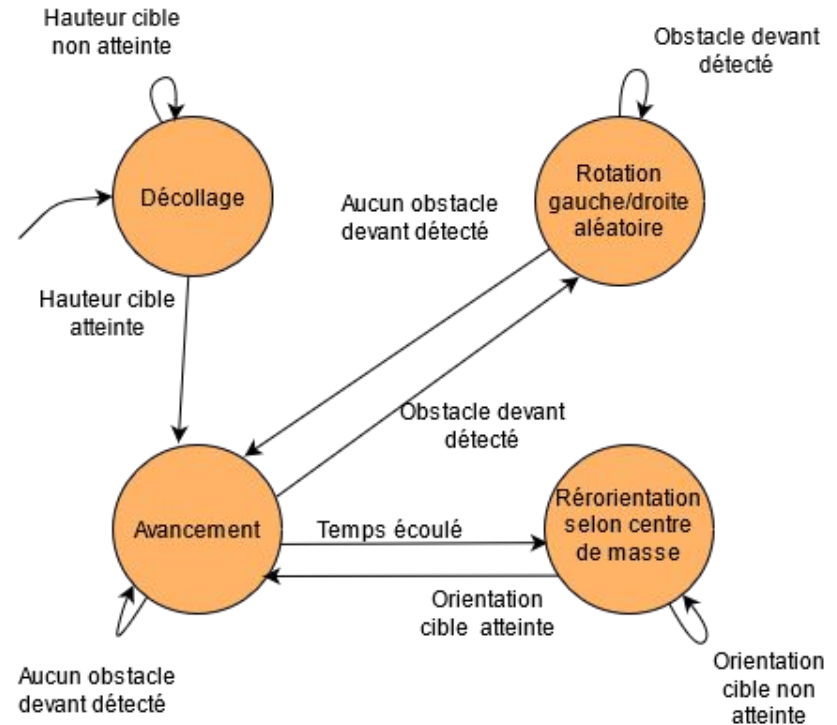
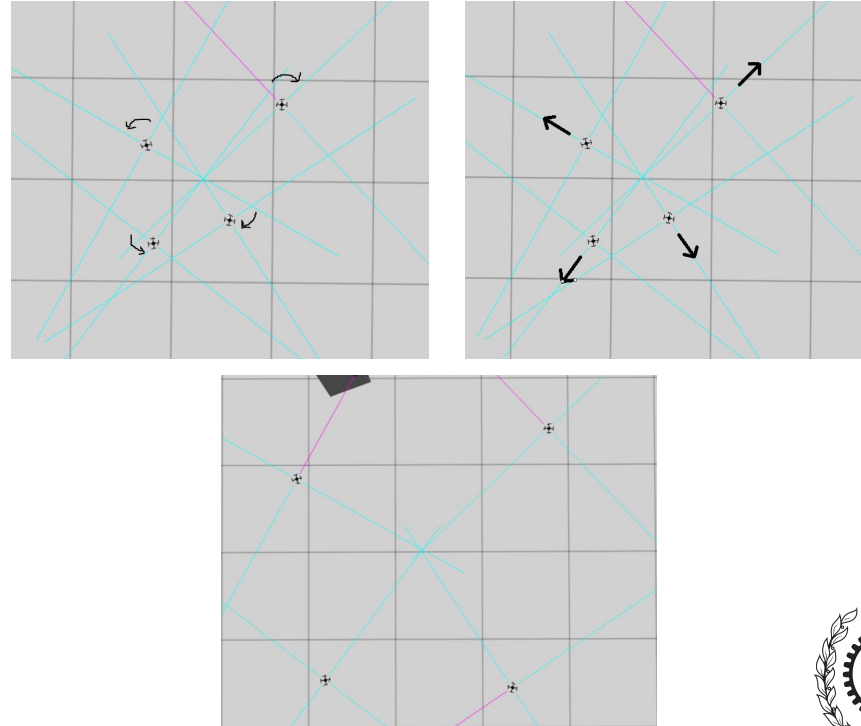


Figure 13. Machine à états d'exploration

3.3 Exploration (suite)

Réorientation dans la direction opposée au centre de masse

- Permet une exploration plus efficace de tous les lieux
- Favorise l'évitement des autres drones



Présentatrice: Yasmine M.

Figure 14. Étapes de la réorientation dans la direction opposée au centre de masse

3.4 Retour à la base

- Réorientation vers la base
- Rotation et avancement pour contourner un obstacle
 - Temps limité pour contourner dans un certain sens
 - Doublage du temps maximum et changement de direction
- Vérification d'une voie libre
 - À droite si rotation à gauche
 - À gauche si rotation à droite

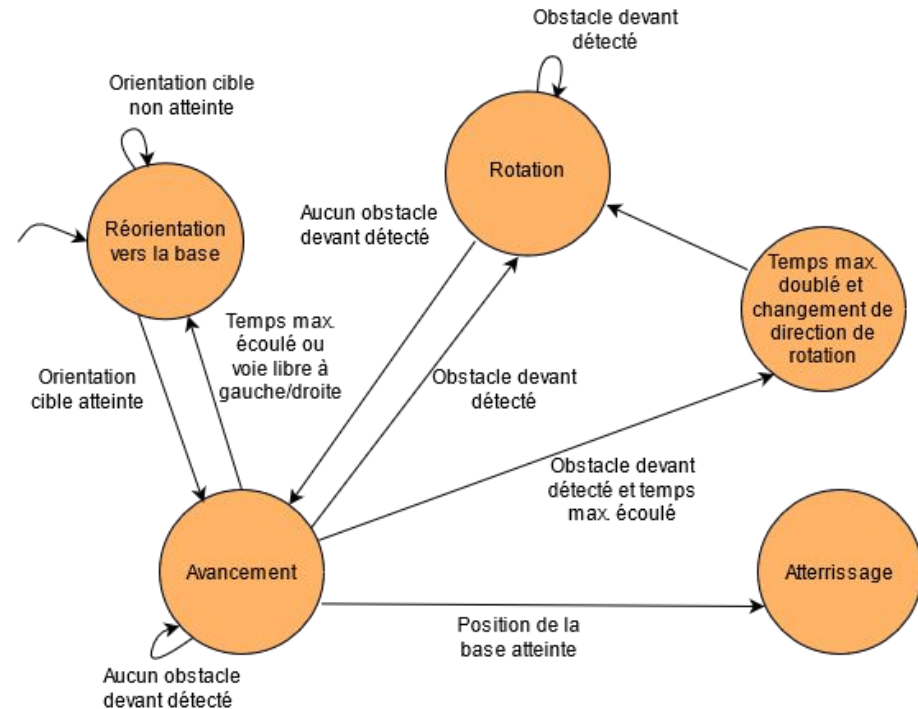


Figure 15. Machine à états du retour à la base

4.1 Tests et résultats

- Développement : ARGoS ➡ Code embarqué ➡ Revue de code ➡ Validation finale dans la volière
- Tests
 - Liste de contrôle
 - Environnements aléatoires en simulation
 - Création de scénarios complexes dans la volière
- Contrôle de la qualité
 - 100% du code a été soumis à un processus de révision
 - Aucun bogue connu ne demeure
- Résultat : solution applicable à tous les cas testés
- 100% des requis acceptés ont été respectés... et plus encore

Présentateur: Misha K.-R.



4.2 Points saillants de notre solution

- Interface utilisateur raffinée
 - Interface réactive pour différentes tailles d'écran
 - Support pour cellulaire et tablette
 - Thème jaune et noir inspiré des abeilles
- Requis optionnels pour la carte générée
 - Carte 3D
 - Positions des drones en temps réel
 - Lignes de visualisation pour les capteurs des drones

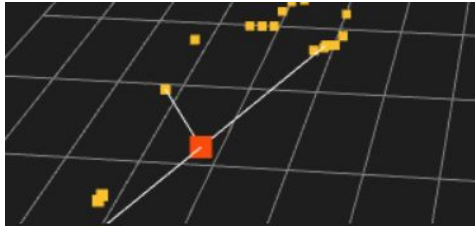


Figure 17. Détection de points

Présentateur: Misha K.-R.

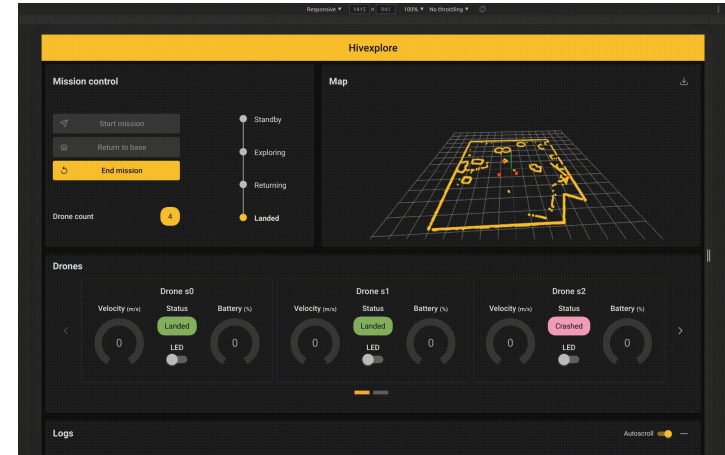


Figure 16. Interface Web réactive



4.2 Points saillants de notre solution (suite)

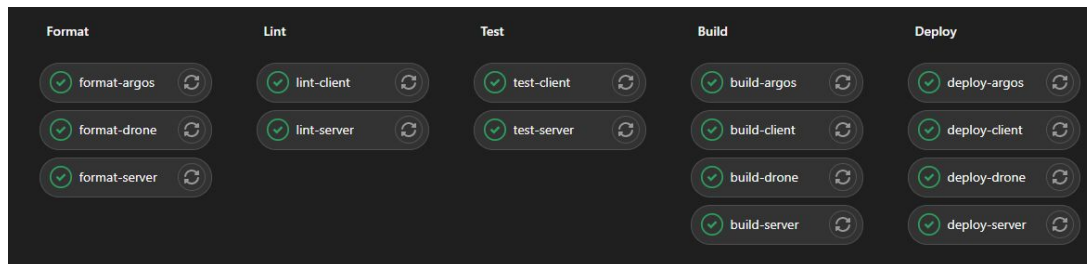
- Expérience utilisateur conviviale
 - Démarrage à l'aide d'une seule commande (*Docker Compose*)
 - Messages de diagnostic
 - Documentation facile à lire : *README.md*
- Algorithmes puissants pour l'exploration et le retour à la base
- Intégration continue
 - Conteneurisation des quatre sous-projets: client, serveur, *Crazyflie* et *ARGoS*
 - Pipeline *GitLab CI/CD* avec les conteneurs *Docker*

```
> ./start.sh --help
Usage: start.sh [-h] {drone|argos|build}

Hivexplore - mapping rooms with drone swarms!

Command:
  drone    use the Crazyradio to connect to Crazyflies
  argos    use ARGoS to simulate the drones
  build    rebuild all Docker images
```

Figure 18. Script de démarrage à une commande



Présentateur: Misha K.-R.

Figure 19. Pipeline GitLab



5.1 Réunions hebdomadaires

- Réunions planifiées
 - Lundi après-midi : réunion complète
 - Jeudi matin
- Déroulement des réunions
 - Ordre du jour sur **Google Docs**
 - *Stand-up*
 - Rétrospective de *sprint*
 - Ce qui s'est bien passé
 - Ce qui s'est mal passé
 - Ce qui est à continuer
 - Ce qui est à arrêter
 - Établissement des horaires de tous les membres : qu'est-ce qui bloque, comment s'organiser avec les drones?
 - Gestion et planification du sprint sur **GitLab**
 - Pauses

Présentateur: Nathanaël B.-D.



5.2 Gestion du développement logiciel

- Utilisation de logiciel **GitLab** pour la gestion des tâches
- Tâches créées et assignées pour chaque sprint à chaque semaine
 - Utilisation des *milestones*
 - Utilisation des *boards*
 - Suivi des heures et des estimés
- *Merge requests*
 - Revue par les autres membres de l'équipe
 - Fusion à la branche principale après deux approbations
- Rapport d'avancement
 - Ce qui a été fait
 - Ce qui est en retard
 - Ce qu'on prévoit faire pour la semaine prochaine

Présentateur: Nathanaël B.-D.



5.3 Coordination hors des réunions

- Utilisation de l'application **Discord** en tout temps pour communiquer
- Canaux de communication
 - #annonces : annonces importantes d'avis général
 - #questions : questions des membres sur les tâches
 - #avancement : avancements reliés aux tâches personnelles
 - #réunions : coordination des réunions
 - Et plusieurs autres

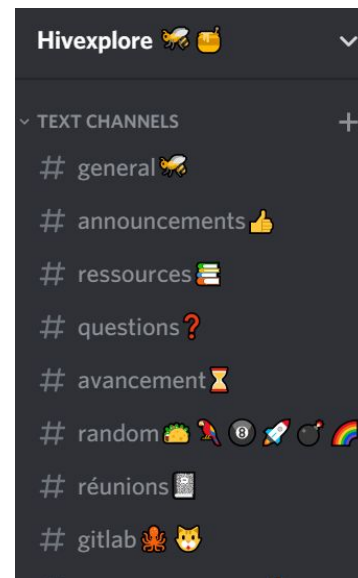


Figure 20. Divers canaux de communication

5.4 Rôles de l'équipe

Misha Krieger-Raynauld Animateur	Nathanaël Beaudoin-Dion Coordonnateur	Rose Barmani Maître du temps
Samer Massaad Secrétaire	Yasmine Moumou Avocat du diable	Simon Gauvin Porte-parole

Figure 21. Rôles formels des membres de l'équipe

Présentateur: Nathanaël B.-D.



Conclusion

- Description du système
- Démonstration
- Algorithmes
- Tests et résultats
- Gestion de projet

Présentatrice: Rose B.



Réflexions

- Si nous devions refaire le projet
 - Tenir nos sprints du jeudi au jeudi pour mieux accommoder nos horaires (travail la fin de semaine)
 - Faire le CI plus tôt pour en profiter
- Ce qui nous a aidé
 - Séparer les deux drones pour permettre à deux personnes de les avoir en début de projet
 - Planifier de façon prévoyante et détaillée les tâches pour éviter les blocages et les imprévus
 - Programmer en paires
 - Communiquer de façon ouverte et fréquente
 - Avoir la capacité à s'adapter à l'horaire de chacun

Présentatrice: Rose B.



Travaux futurs

- Améliorer calcul du niveau de batterie
- Implémenter les capteurs de distance du haut et du bas dans ARGoS
- Utiliser le nouvel actionneur de vitesse pour contrôler les drones dans ARGoS comme les vrais drones
- Implémenter DOOR-SLAM (*Distributed, Online, and Outlier Resilient Simultaneous Localization And Mapping*)

Présentatrice: Rose B.



Références

NASA/JPL-Caltech. (2020). Rover Evolution Plate Artwork. Tiré de <https://mars.nasa.gov/resources/25651/rover-evolution-plate-artwork/>

NASA. (2020). The Perseverance rover carries seven instruments to conduct its science and exploration technology investigations. Tiré de https://mars.nasa.gov/system/resources/detail_files/25045_Perseverance_Mars_Rover_Instrument_Labels-web.jpg

Jeshoots. (2017). Drone silhouette with camera flying in the sunset light. Tiré de <https://jeshoots.com/drone-silhouette-with-camera-flying-in-the-sunset-light/>

Bjorn Mauritz. (2017). Crazyflie2.0-585px.JPG. Tiré de <https://github.com/bitcraze/bitcraze-website/blob/master/src/images/Crazyflie2.0/Crazyflie2.0-585px.JPG>

