

P53. Contrôle 3D d'essaims de drones

Année 2016

Encadrants : Thierry DUVAL (LUSSI) et Didier GUEROT (ITI)

Partenaires : François Legras, Deev Interaction

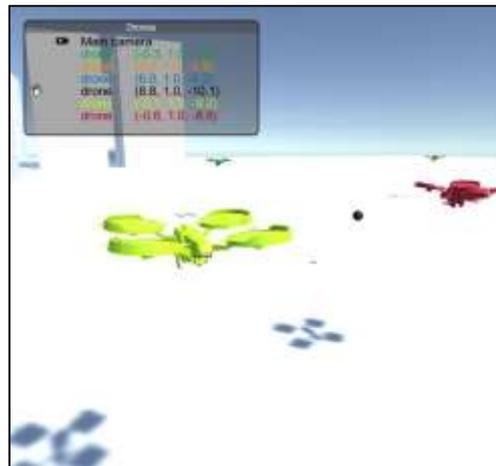
Mots clés : Interaction 3D, Visualisation 3D, Essaims de drones, Systèmes robotiques (ROS), drones volants, supervision, Unity 3D, C#, Octomap, SLAM.

Résumé : Ce projet consiste à réaliser une simulation de contrôle d'essaims de drones tournant sur ROS depuis une interface développée sous Unity offrant des possibilités de visualisation et d'interactions 3D.

1. Présentation et contexte du projet

Deev Interaction est une entreprise opérant dans le secteur de la défense et de la sécurité civile. Depuis quelques années, elle étudie l'apport de la visualisation 3D de situations tactiques et de l'interaction 3D pour la simulation de commande d'essaims de drones de surveillance.

C'est dans ce cadre que nous a été soumis le projet de conception d'une simulation permettant de faire évoluer des drones dans un environnement 3D complexe. Ces drones devront restituer l'environnement observé, permettant ainsi leur supervision et leur contrôle depuis une seule et même plateforme en 3D, qui offre de meilleures possibilités d'interaction qu'une vue de dessus en 2D.



Environnement Unity3D

2. Méthodologie développée pour aboutir

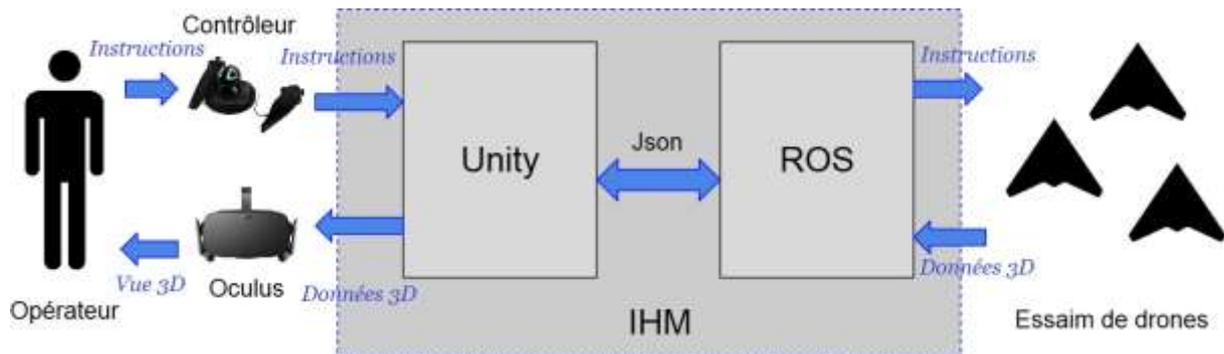
Pour mener à bien ce projet, nous avons dégagé 5 lots principaux s'inscrivant dans un « cycle en V » allant de l'analyse des besoins jusqu'à l'intégration. Afin de garantir une meilleure performance lors de la réalisation technique, nous avons couplé ce cycle en V à une méthode SCRUM consistant à effectuer des sprints hebdomadaires sur les différentes tâches. Les sprints permettent l'implémentation des algorithmes sur ROS, Unity 3D, le protocole d'interfaçage des choix d'ingénierie, mais également la réalisation de tests unitaires et d'intégration qui garantissent la validation ou la reconduction des tâches. Cette structure nous permet ainsi de redéfinir certains choix d'ingénierie en cas de tests non concluants, et donc de garantir la livraison du produit final dans les délais.

La supervision globale du projet se fait par le responsable de projet en collaboration avec l'équipe de suivi horaire et des risques.

3. Développement des différentes tâches et principaux résultats

1) Couplage Unity-ROS

Le couplage entre Unity et ROS est fait à travers un canal virtuel utilisant le format de données JSON. Ce canal permet de transporter les données de contrôle de l'opérateur vers les drones et les informations mise à jour de position et de reconstitution d'environnement dans le sens inverse.



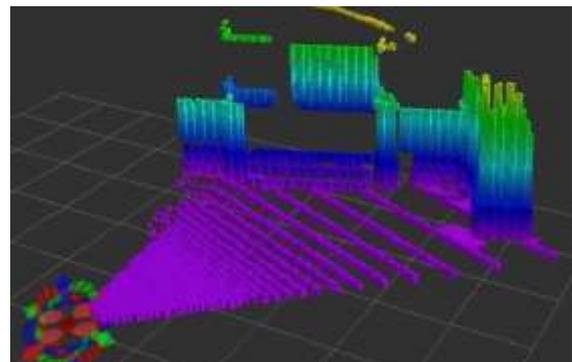
Vue globale du projet – Interfaçage Unity Ros

2) Contrôle des drones, interaction 3D

Deux modes de navigation sont disponibles : le mode manuel et le mode guidé. Dans le mode manuel, l'utilisateur contrôle le drone en temps réel, et le suit en vue subjective. Dans le mode guidé, l'utilisateur assigne au drone une destination ou une trajectoire qu'il va suivre de manière autonome. En cas de rencontre d'obstacle, les drones enverront un message à l'utilisateur qui décidera du comportement à adopter. Ce comportement pourra être défini automatiquement par la suite.

3) Visualisation 3D de l'environnement

Les drones disposent de capteurs laser qui permettent la détection d'obstacles. Les informations de profondeur de champ sont utilisées pour générer des "octrees". Cette structure de données permet de stocker et d'exploiter facilement des informations sur un environnement 3D.



4. Conclusions et perspectives

Reconstruction 3D à partir des données laser

Le prototype développé permet de superviser et de surveiller une zone géographique à l'aide d'un essaim de drones, et d'acquérir une vue en 3D de l'environnement dans lequel ils évoluent.

Un challenge futur peut être d'améliorer le rendu de l'environnement coté utilisateur en équipant les drones de caméras. Les images prises par la caméra permettront de générer des textures et donc de fournir à l'utilisateur plus d'informations sur l'environnement. Les drones pourront également être rendus plus autonomes en développant leur comportement, en particulier lorsqu'ils font face à des obstacles.

Bibliographie

- [1] Rouvrais S., Ormrod J., Landrac G., Mallet J., Gilliot J-M., Thepaut A., and Tremenbert P. "A mixed project-based learning framework: preparing and developing student competencies in a French Grande Ecole". SEFI-EJEE European Journal of Engineering Education, "Engineering Competencies". Volume 31, Number 1. Pages 83-93. March 2006.
- [2] Documentation C# Microsoft : <https://msdn.microsoft.com/fr-fr/library/618ayhy6.aspx>
- [3] Documentation Unity3D : <http://docs.unity3d.com/Manual/index.html>
- [4] Wiki ROS : <http://wiki.ros.org/>
- [5] Documentation GitHub de ROS : <https://github.com/ros>
- [6] Wiki ROSBridge : http://wiki.ros.org/rosbridge_suite
- [7] Documentation GitHub ROSBridge : https://github.com/RobotWebTools/rosbridge_suite

Si vous avez des problèmes pour utiliser ce modèle, contacter Jean-Yves Floch, jy.floch@telecom-bretagne.eu

Quand cette présentation de votre projet est terminée, **vous la transmettez à l'aide du formulaire accessible sur le lien :**

https://intranet.telecom-bretagne.eu/enseignement/les_ formations/ingenieur/s4-impression/