

Compte rendu : Keyboardmission

I- Définition de la mission :

La mission **KeyboardMission** permet à l'utilisateur de contrôler le drone via une interface interactive. Elle offre la possibilité de piloter le drone à l'aide des boutons du clavier (up, down, right, left, b, h, e, et a) , et aussi d'effectuer un retour automatique à la position initiale en cas d'urgence. cette mission peut être utiliser dans les applications suivantes :

Surveillance de sites industriels → Inspection des infrastructures et détection d'anomalies en temps réel.

Recherche et sauvetage → Localisation de personnes en détresse avec un retour vidéo en direct.

Sécurité d'événements → Monitoring aérien pour prévenir les incidents et améliorer la gestion des foules.

II- Les entrées et la sortie de la mission :

1- **Les entrées** : pour fonctionner cette mission tu n'a pas besoin des entrées

2- **Les sorties** : Le code génère plusieurs types de sorties pour l'utilisateur. Dans la console, des messages de statut tels que "Connexion au drone réussie", "Évitement d'obstacles activé", "Décollage en cours..." et "Atterrissage en cours..." sont affichés pour informer de l'état du drone. Lorsqu'une action est effectuée, un retour détaillé est donné, par exemple, "Retour à la maison..." ou "Position initiale enregistrée : {position_initiale}". En parallèle, le flux vidéo du drone est affiché dans une fenêtre OpenCV, permettant à l'utilisateur de visualiser en temps réel ce que le drone filme. Enfin, une interface graphique Tkinter permet de contrôler le drone via des boutons, et chaque interaction, comme un mouvement vers le haut ou le bas, génère une sortie visible dans la console, par exemple, "Démarrage du mouvement vers le haut...".

III- Explication du code de la mission :

Ce script Python permet de contrôler un drone Parrot en utilisant une interface graphique Tkinter et des commandes clavier. L'utilisateur peut piloter le drone à l'aide de boutons ou de touches du clavier pour effectuer des mouvements comme monter, descendre, avancer, reculer, tourner à gauche ou à droite. Il est également possible d'activer l'évitement d'obstacles et de diffuser un flux vidéo en temps réel via RTSP. En cas d'urgence, le drone peut revenir automatiquement à sa position initiale grâce à une fonction dédiée. Le script établit la connexion avec le drone, effectue un décollage, et permet à l'utilisateur de contrôler le drone tout au long de la mission. Après avoir terminé la mission, le drone effectue un atterrissage automatique et se déconnecte. L'interface graphique offre une façon simple et intuitive de contrôler le drone, tout en fournissant un retour en temps réel sur les actions effectuées.

IV- Explication des fonctions du script :

obtenir_position_initiale() : Cette fonction récupère la position actuelle du drone en utilisant la méthode `get_state` pour obtenir les informations de position à partir du message `PositionChanged` de l'API Olympe.

recuperer_position() : Cette fonction lance un thread pour récupérer et afficher la position initiale du drone dès qu'il est connecté et prêt, en utilisant la fonction `obtenir_position_initiale()`.

fonction_retour_maison() : En cas d'urgence, cette fonction permet au drone de revenir automatiquement à sa position initiale en utilisant la bouton **e** pour retourner aux coordonnées enregistrées.

start_video_stream() : Cette fonction ouvre un flux vidéo RTSP du drone, le redimensionne et l'affiche en temps réel via OpenCV dans une fenêtre pour permettre à l'utilisateur de voir ce que filme le drone.

fonction_haut() : Cette fonction déplace le drone vers le haut en utilisant la bouton **h**, ce qui augmente l'altitude du drone de 10 cm (0.1 m).

fonction_bas() : Cette fonction déplace le drone vers le bas en utilisant la bouton **b**, réduisant l'altitude du drone de 10 cm (0.1 m).

fonction_avance() : Cette fonction déplace le drone vers l'avant de 10 cm en utilisant la bouton **up** pour avancer horizontalement.

fonction_arriere() : Cette fonction déplace le drone vers l'arrière de 10 cm en utilisant la bouton **down** pour reculer horizontalement.

fonction_gauche() : Cette fonction fait tourner le drone de 5° vers la gauche en utilisant la bouton **left**, modifiant son orientation en radians.

fonction_droite() : Cette fonction fait tourner le drone de 5° vers la droite en utilisant la bouton **right**, modifiant son orientation en radians.

fonction_atterrissage() : Cette fonction ordonne au drone d'atterrir en utilisant la bouton **a**, puis se déconnecte et ferme l'interface graphique après l'atterrissage.