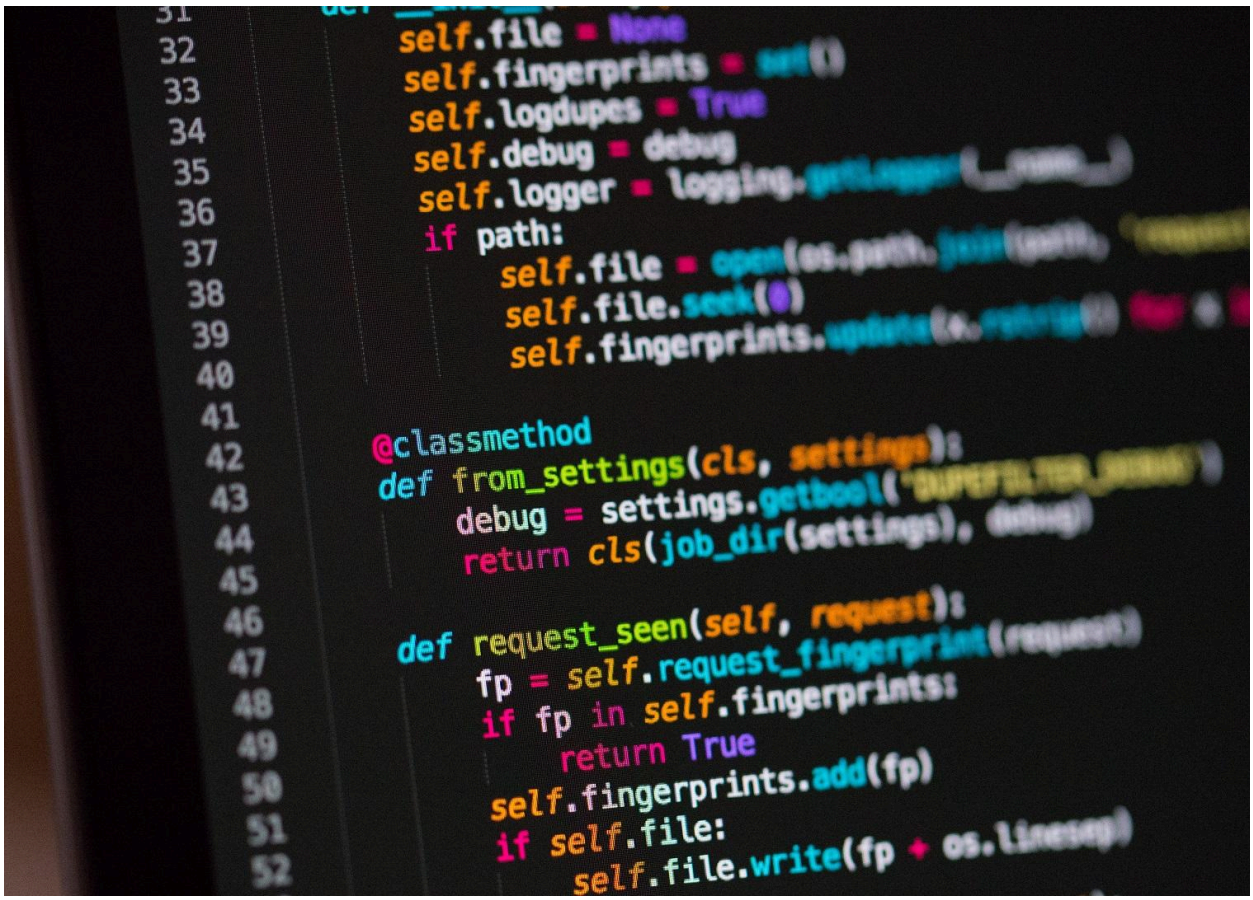
PEAUPARDIN Pierre, NOTTER Nicolas, TONDEUR Vlad, BAROTTIN Thibaut

RAPPORT DE PROJET PYTHON
CODE ORIENTÉ OBJET



Introduction

L'objectif de ce projet est de réaliser une application avec une interface graphique (« GUI » = graphical user interface) qui permet de se connecter au robot. Une fois connecté, l'utilisateur peut visualiser les mouvements du robot et commander ses déplacements.

I. Répartition des Tâches

- Pierre : Mise en place du Code de Base pour la GUI (Graphic User Interface)
- Thibaut : Partie Commande des déplacements via Publisher ROS, une partie de la documentation
- Vlad : Partie communication entre ROS, Python, Gazebo et le TurtleBot
- Nicolas : Amélioration du code du GUI, mise en places de commentaires, une partie de la documentation, Rapport de fin de projet
- Tout le monde : tests unitaires

II. Fonctionnement de l'Application

Le programme principal **main.py** est le programme à exécuter. Ce dernier va faire appel aux autres programmes ainsi qu'aux packages nécessaires (décrites dans requirement.txt). Le programme **Gazebo_launch.py** va lancer Gazebo à chaque utilisation, afin de pouvoir simuler convenablement le TurtleBot que l'on souhaite commander. Les programmes **robot_controller.py** et **command_frame.py** vont respectivement assurer la simulation correcte du robot et de ses déplacements à l'aide de Gazebo, ainsi que le contrôle dudit robot dans la simulation (déplacement linéaire dans une direction, rotation, arrêt, etc). Enfin, **movement_frame.py** est le programme qui va permettre de représenter sous forme de graphe dans l'interface graphique (GUI) les différentes grandeurs que mesure le TurtleBot.

III. Problèmes Rencontrés

La Carte OpenCR de notre TurtleBot étant défaillante, nous ne pouvons hélas pas tester notre programme sur le robot "en réel", et devons donc utiliser Gazebo pour simuler son comportement.

IV. Résultats Finaux

Nous avons au final un projet qui permet effectivement de simuler et de contrôler le Turtlebot dans l'environnement Gazebo, ainsi que de récupérer les grandeurs qu'il nous renvoie pour les afficher sous forme de Graphes. Cependant, en raison de la défaillance technique de notre TurtleBot au niveau de la carte OpenCR, nous ne pouvons plus utiliser efficacement ROS dans une optique de communication entre notre Robot et nos ordinateurs. Nous avons donc un projet opérationnel en théorie, mais que nous n'avons pas pu tester sur le projet réel.

V. Ressources Utilisées

Pour utiliser ROS :

- https://www.ros.org/
- https://wiki.ros.org/rospy
- https://docs.ros.org/en/kinetic/api/rospy/html/index.html
- https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/

Pour utiliser Gazebo :

- https://gazebosim.org/home

Pour importer les packages nécessaires :

- Le Package SetupTools

Le dépôt Github dont nous nous sommes servis :

- https://github.com/AsapNedaLNeB/Projet-Python-Turtlebot-Cormoran

Pour setup ROS avec Gazebo :

- https://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/