

Rapport de la séance 14

19 mars 2024

Objectif de séance :

- Récolter les images pour la banque de données
- Essayer le suivi de ligne par caméra

Information concernant les servomoteurs :

Tout d'abord la dernière séance, j'avais des problèmes avec l'activation des servomoteurs par le biais de la Jetson Nano. Cependant, lors de notre dernier cours d'informatique, nous avons appris à faire communiquer la Jetson Nano avec la carte Arduino avec de l'I2C.

Par conséquent, il semble être une bonne option de rajouter sur notre nouveau robot une carte Arduino, afin de faciliter l'activation des servomoteurs, mais aussi la communication entre nos robots.

En effet, le fait que la Jetson Nano nécessite d'être codé en Python me ralentit dans l'avancée du projet car il ne s'agit pas d'un langage de programmation que je maîtrise.

Ainsi, il semble être une bonne option de rajouter une carte Arduino au robot afin d'utiliser les servomoteurs, les modules de communication RF, et potentiellement d'autres capteurs qui seraient nécessaires au bon fonctionnement du robot dans l'entrepôt miniaturisé.

Suivi de route par caméra : (Source dans le README.md)

Première tentative :

Place maintenant, au travail réalisé dans la séance qui concerne le suivi de route par caméra.

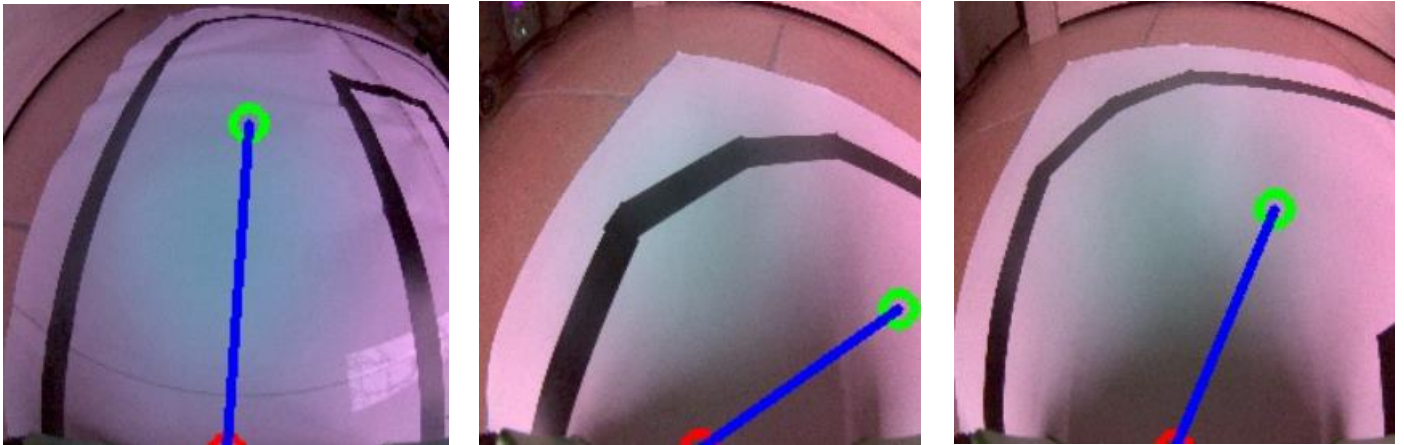
Pour rappel, ce fonctionnement se base sur des Jupyter Notebooks, qui sont l'équivalent de bibliothèques d'informations où se trouve du code et des explications concernant leur fonctionnement. Ces Jupyter NoteBooks sont des publications de Nvidia sur l'apprentissage et la découverte des notions d'intelligence artificielle.

La première étape est de prendre une multitude de photos.

Celles-ci ont pour objectif de créer une banque d'image qui servira à l'entraînement de l'intelligence artificielle. Pour ce premier test, j'ai pris une centaine de photos du robot placé à divers endroits de la piste avec différents angles selon où il se trouve.

De plus, au moment de prendre la photo, il faut utiliser une manette (avec du code présenté lors du dernier rapport), qui permet d'indiquer sur chaque image l'endroit vers lequel devrait se diriger le robot.

Quelques photos en guise d'exemple :



Ensuite, arrive l'étape d'entraînement de l'IA.

Pour se faire, le Jupyter Notebooks se sert de la librairie : torch. Cette librairie permet de réaliser de l'apprentissage machine. C'est donc grâce à cette librairie qu'il est possible de convertir la banque d'image en un réseau de neurone.

Enfin, le test pratique.

Pour cette première tentative, le résultat est peu glorieux. Les seules choses qui semblent légèrement fonctionner sont les virages. En effet, lors d'un virage, le robot effectue une forte rotation conforme à ce qu'il devrait réaliser pour tourner dans le virage. Cependant, il ne parvient pas à continuer sa route après le virage.

En conclusion de cette première tentative, il semble que de légères modifications pourront déjà grandement améliorer la précision du robot.

En effet, je pense qu'agrandir la largeur de la route permettrait au robot de mieux appréhender l'arrivée des virages. De plus, il est préférable de recentrer la piste vers l'intérieur du panneau, car de légères sorties de route conduisent à la chute du robot.

Seconde tentative :

Ainsi, j'ai effectué une deuxième tentative avec une nouvelle piste plus large et plus centrée. J'ai repris les photos, au nombre de 200 cette fois-ci. Et en réalisant la même manipulation que précédemment, le résultat est nettement mieux (voir GIF dans le README.md). En effet, le suivi est quasi fonctionnel, il semble que la caméra devrait être plus relevée afin que l'anticipation des virages s'effectue de plus loin.

Troisième tentative :

Je vais donc réaliser une troisième tentative avec 200 nouvelles photos en relevant la caméra. Cependant, je n'ai pas le temps de finaliser la prise de ces 200 nouvelles photos durant la séance, je n'ai donc pas de vidéo ni d'arguments qui confirme que le fait de relever la caméra améliore la précision du robot.

Objectif de la prochaine séance :

- Avoir un suivi de route parfaitement fonctionnel
- Créer une liaison I2C entre la carte Arduino et la Jetson Nano
- Ajouter des capteurs sur le robot pour qu'il s'arrête aux endroits souhaités (emplacement de dépôt des colis)