

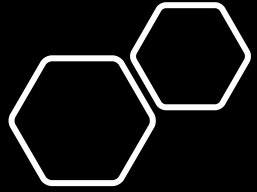
LY Alexandre – CHAU Julien

Robot Mindstorm



Objectif du projet

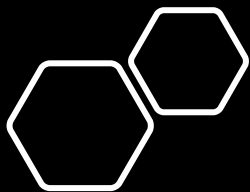
- Programmer un robot pour qu'il puisse suivre une ligne de couleur distincte, qu'importe la forme du circuit.
- Problématique simple mais avec la liberté à beaucoup d'outils d'optimisation (vitesse, qualité du mouvement, correction d'erreurs ...).
- Technologie à usage domestique ou industrielle : transport de produits, de personnes.



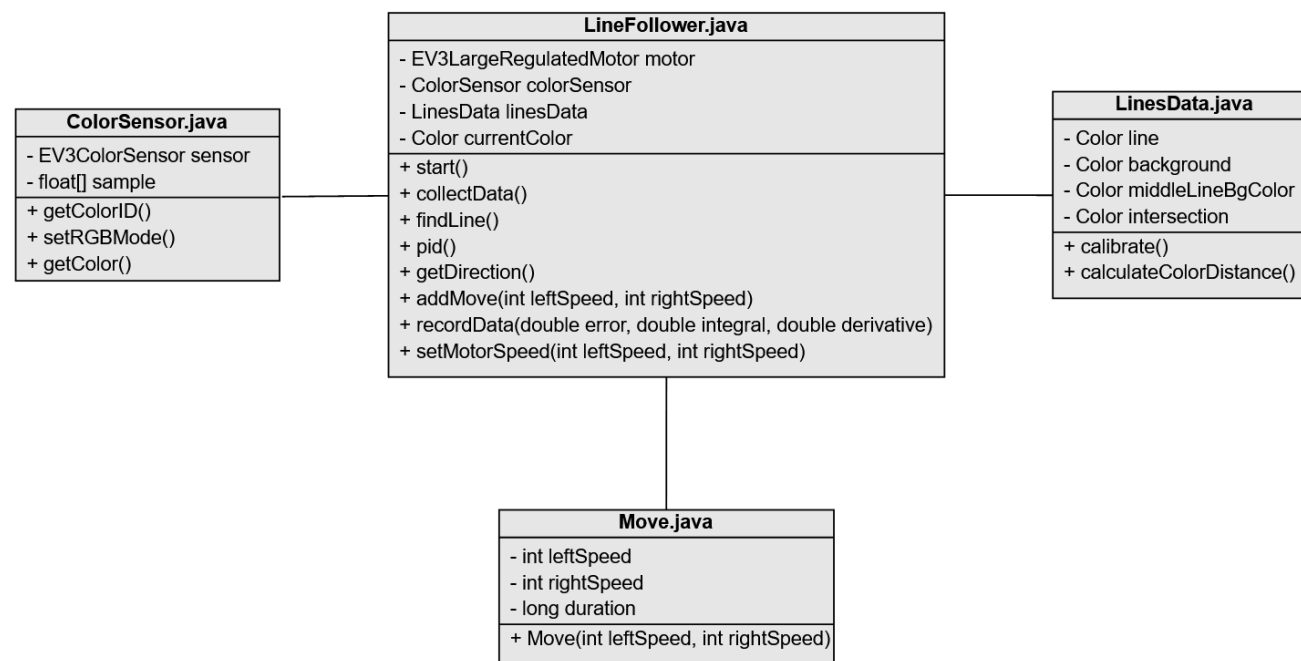
Exemples d'utilisation

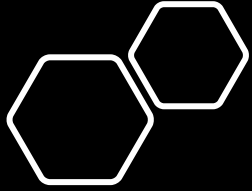


*Robots automatisés de transport dans les entrepôts de Amazon
(CBS News 2021)*



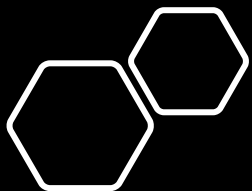
Architecture du projet





Problématiques

- Préparation: installation logicielle, montage du robot
- Apprentissage des couleurs: lire et stocker les couleurs perçues par le capteur
- Suivre une ligne: faire tourner les roues avec le moteur et les données des couleurs précédentes.
- Optimisations: PID, mouvement fluide, limiter les erreurs.

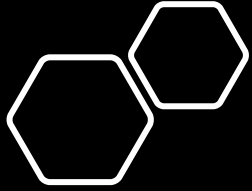


Compétences/Outils
utilisés

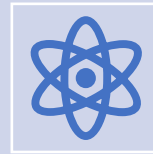
Java comme langage de
programmation.

Librairie LeJOS: firmware de
programmation pour la brique EV3.

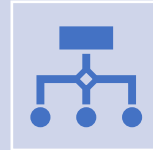
Git.



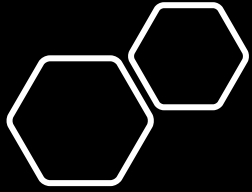
PID



Proportional-integral-derivative mechanism.

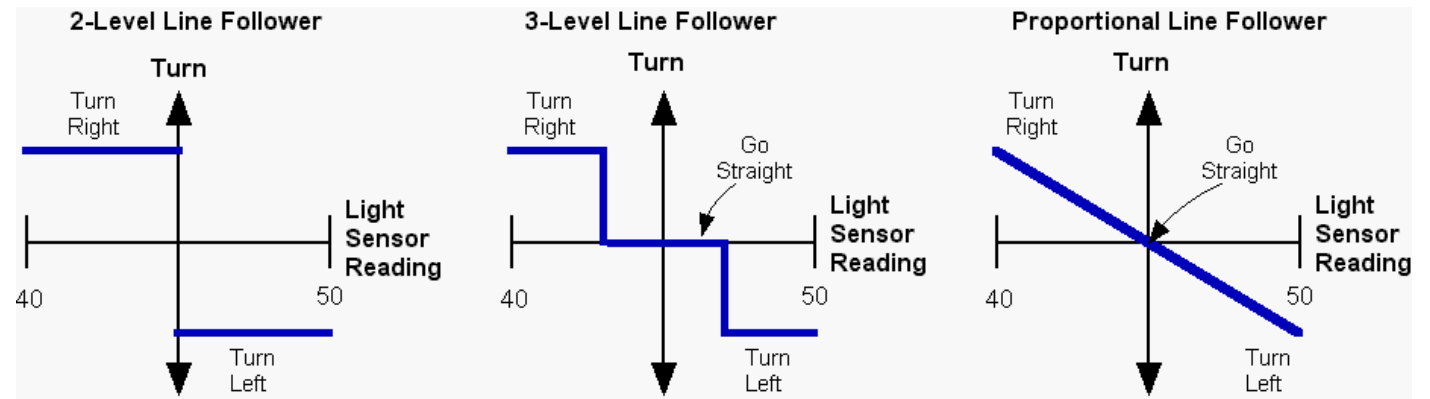


Un algorithme de correction d'erreurs très utilisé dans l'industrie.

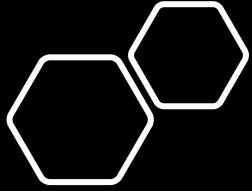


Principes

Proportional



- > Expliquer l'algo
- > Pourquoi avoir opté pour le P controller ? (avantage, désavantage)



Pseudocode

```
previous_error := 0
integral := 0
loop:
    error := setpoint - measured_value
    proportional := error;
    output :=  $K_p \times$  proportional
    previous_error := error
    wait(dt)
    goto loop
```

https://en.wikipedia.org/wiki/PID_controller

Conclusion

- Qu'avons-nous appris ?
 - Programmer un robot.
 - Utiliser et traiter les données d'un capteur.
 - Travail en groupe.
 - Traitement des couleurs.
- Des améliorations ?
 - Optimiser et rendre le robot plus compétitif.
 - Réfléchir à une méthode de correction d'erreurs plus efficace.