EV3 Project - Line follower

Joaquim Lefranc & Jérôme Skoda

Master Informatique - Paris VII

lefrancjoaquim@gmail.com - jerome.skoda@hotmail.fr

Introduction : problématique

Enoncé du problème :

Le robot doit pouvoir suivre une ligne courbe, fracturée, et de couleur arbitraire non constante. Il doit également être capable de s'arrêter automatiquement après avoir accompli deux tours. Le début du circuit est symbolisé par une couleur différente.

Intérêt :

Parkings automatisés avec des tracés, déplacement des bagages dans les gares…

lefrancjoaquim@gmail.com - jerome.skoda@hotmail.fr

Fonctionnalités

Utilisation standard :

- *Calibration : ./Main -c*
- *Scanning : ./Main -s*
- *Fichier de calibration : calibration.calib*

Fonctionnalités :

- *Détecte la couleur de fond, et la première couleur de la ligne.*
- *Suit la ligne pendant deux tours.*

lefrancjoaquim@gmail.com - jerome.skoda@hotmail.fr

Architecture et conception

Considérations techniques :

- *Système EV3Dev (Debian jessie)*
- *Connexion ssh*
- *60Mo de RAM*
- *Architecture ARM 300Mhz*
- *Cross-compilation*
- *Langage C++*

lefrancjoaquim@gmail.com - jerome.skoda@hotmail.fr

Architecture et conception

Architecture du projet :

Common

- *Color*
- *ColorEntry*
- *ColorRGB*
- *Direction*
- *DutyRatio*

Devices

- *ColorSensor*
- *Engine*
- *Robot*

Libs

- *ev3dev*

Main

Makefile avec une règle « make send » qui permet l'envoi direct des nouveaux binaires

lefrancjoaquim@gmail.com - jerome.skoda@hotmail.fr

Capteur de couleur

Calibration :

- *50 échantillons*
- *Sauvegarde du maximum des trois composantes (RVB)*
- *Sauvegarde du minimum des trois composantes (RVB)*

Lecture :

- *5 échantillons*
- *Calcul des distances*

$$c1 = RVB(140,\ 140,\ 140)$$
$$c2 = RVB(150,\ 160,\ 160)$$

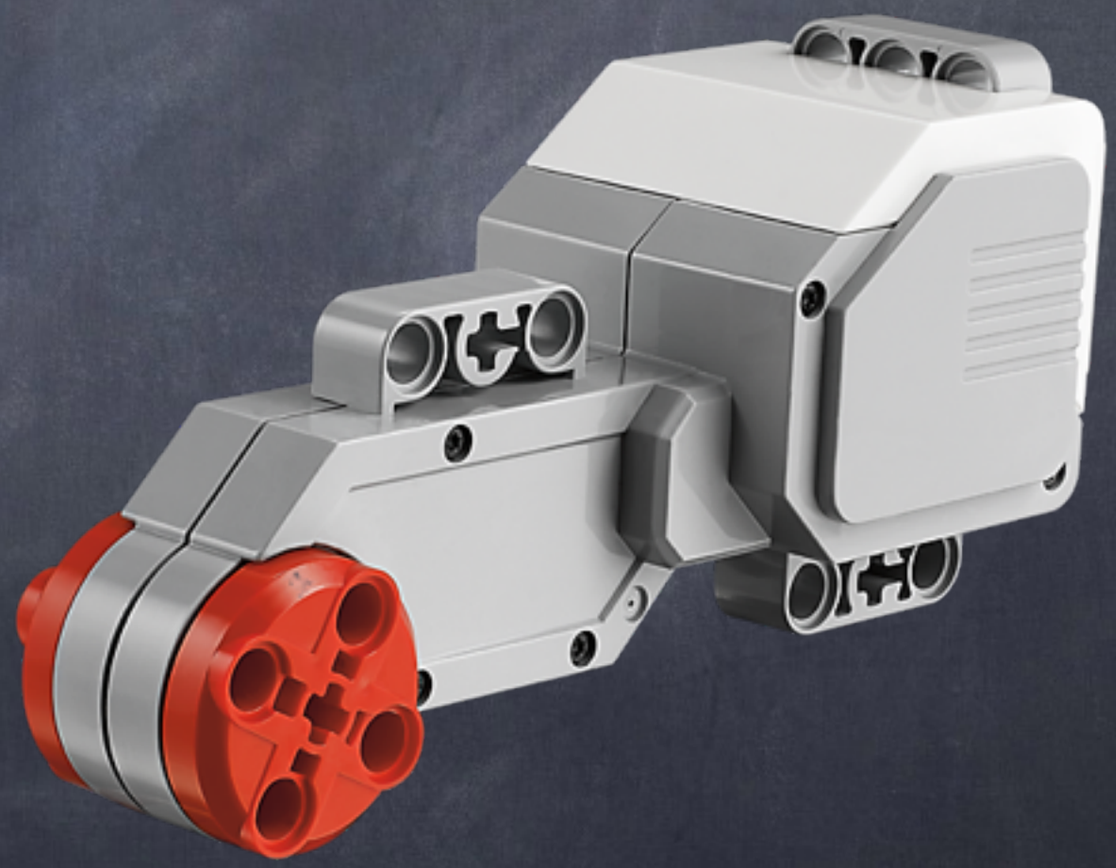
$$distance(c1,\ c2) = \sqrt{(R_{c1} - R_{c2})^2 + (V_{c1} - V_{c2})^2 + (B_{c1} - B_{c2})^2}$$

lefrancjoaquim@gmail.com - jerome.skoda@hotmail.fr

Propulsion et contrôles

Gestion des moteurs :

- *Deux moteurs dans l'objet Engine*
- *Fonctions de contrôle (run , stop, setDirection, …)*



lefrancjoaquim@gmail.com - jerome.skoda@hotmail.fr

Test et difficultés techniques

Tests :

- En salle et sur circuit perso

Difficultés rencontrées :

Problème de fixation de la roue directionnelle

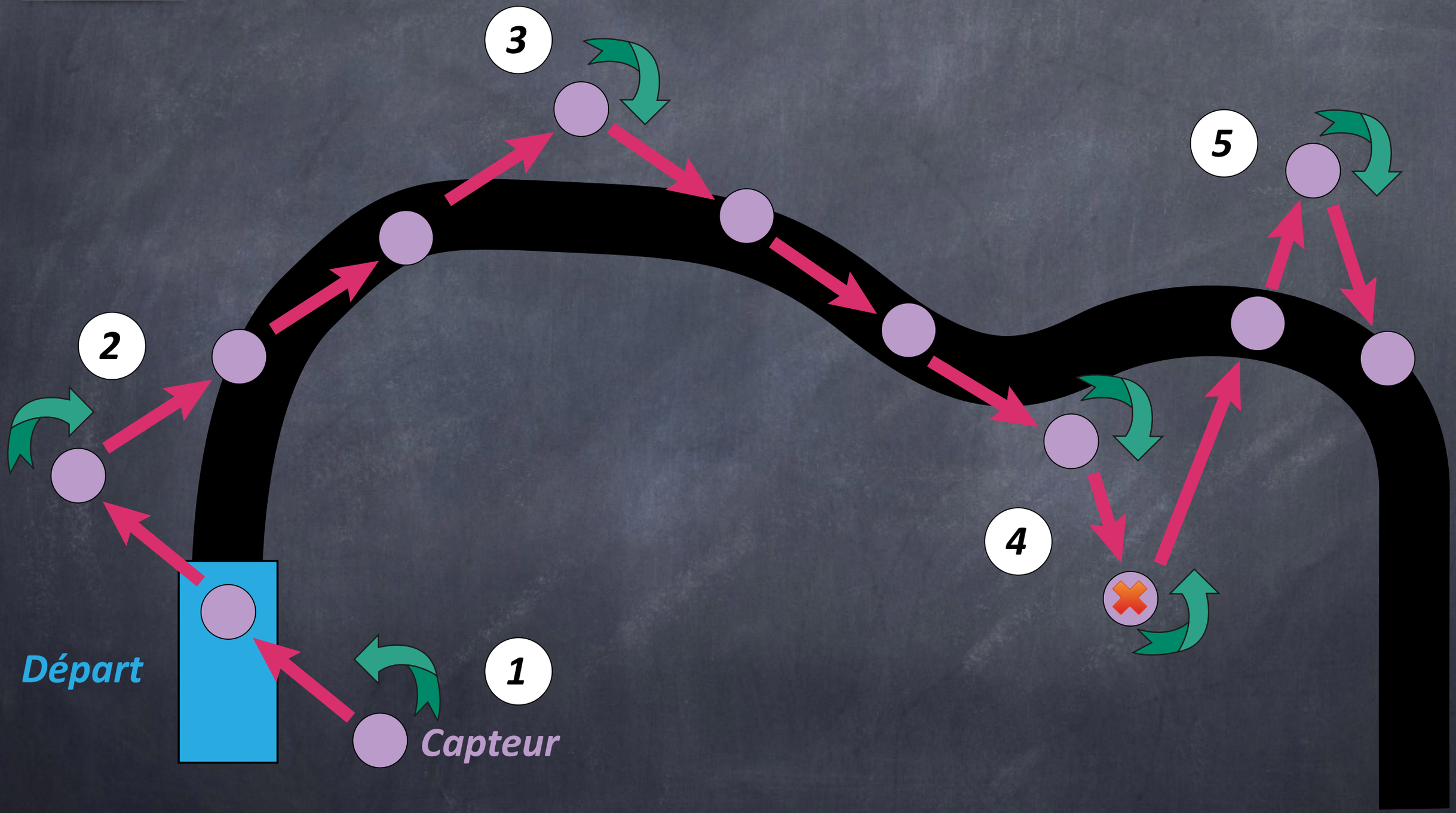
Ajout de blocs pour diminuer l'effet de levier

Exception lors de la lecture avec le capteur

- *Try catch brutal évitant la fin du programme*
- *Ajout d'un sleep entre chaque capture*

lefrancjoaquim@gmail.com - jerome.skoda@hotmail.fr

Algorithme de suivi



lefrancjoaquim@gmail.com - jerome.skoda@hotmail.fr

Conclusion

Apports de compétences :

- Mise en pratique des acquis du cours de C++ / Système

- Subir les contraintes propres aux moteurs / capteurs

Améliorations possibles :

- Deuxième capteur pour avoir des informations plus pertinentes et une fluidité accrue.

- Adaptation des différents paramètres (vitesse, temps) en fonction de la tension des piles.

Expérience exaltante, la robotique c'est magique.

lefrancjoaquim@gmail.com - jerome.skoda@hotmail.fr