

# Cartographie

UE: Introduction à la  
robotique mobile

Rapport rédigé par:  
Chida Omar  
Battikh Linda

# Sommaire

<b>I-Stratégies choisies</b>	<b>3</b>
<b>II-Difficultés rencontrées:</b>	<b>3</b>
<b>III-Robot</b>	<b>3</b>
<b>IV-Capteurs</b>	<b>4</b>
<b>V-Cartographie</b>	<b>4</b>
<b>VI-Planification</b>	<b>5</b>
<b>VII-Contrôle</b>	<b>6</b>

## I-Stratégies choisies

Afin de mener à bien le projet, nous avons opté pour le fait de laisser le robot parcourir la salle en utilisant les moteurs pour le faire avancer, tourner à droite et tourner à gauche.

Grâce à ses capteurs, il sera en mesure de percevoir les obstacles (afin de savoir qu'est ce qui se trouve dans la pièce et où exactement ).

Puis mémoriser leur coordonnées dans un tableau afin de ne pas tourner en rond.

Et pour finir il dessinera la carte.

## II-Difficultés rencontrées:

- Avoir un robot qui ne fonctionne pas a été un véritable handicap pour nous. Nous ne pouvions pas vérifier si notre code fonctionnait ou pas. Nous devions attendre jusqu'à ce que l'un des groupe ait fini pour pouvoir emprunter leur robot. Et c'est pour cette raison que nous avons un peu de retard par rapport aux autres.
- Certains capteurs ne réagissaient pas convenablement, cela nous posait un problème car nous ne savions pas si c'était un problème lié au code ou pas.
- Nous avons aussi rencontré des difficultés lors de l'écriture de l'algo qui crée la cartographie.
- Le fait que le robot ne tourne pas sur lui même était un petit problème qu'on a su régler en rendant le rapport cyclique négatif afin que le second moteur tourne dans l'autre sens.
- Nous avons aussi rencontré des difficultés pour dessiner la carte puisque nous travaillons avec un robot et non pas un ordinateur.
- Nous avons eu du mal à trouver la librairie qui nous permettait de générer les images. Mais à la fin nous avons découvert Pygame

## III-Robot

Le robot utilisé est le EV3 Lego Mindstorm qui a 4 ports à l'arrière et 4 ports à l'avant. On a utilisé le capteur ultrason, 2 grands moteurs

auxquels on a lié deux grandes roues afin de permettre au robot de circuler. Et un autre plus petit qui lui faisait bouger le capteur ultrason afin de permettre la détection des obstacles. (Photo 2)

## IV-Capteurs

On a utilisé le capteur ultrason, qui grâce à sa capacité de voir qu'est ce qui se trouve à 2m de long, a contribué à détecter les obstacles présents dans la pièce afin de tout pouvoir mettre sur la carte.

Le touch sensor était utile pour pouvoir arrêter le robot quand nous le voulions, juste en appuyant sur lui. C'était pratique quand nous détectons un bug.

## V-Cartographie

Dessiner la cartographie n'était pas quelque chose de facile pour nous puisque nous sommes encore des débutants en robotique et les lieux que le robot explore sont vraiment des lieux "random" on a pas de critères précis (le lieu peut avoir n'importe quel forme géométrique et les obstacles n'ont pas de formes prédéfinies) pour pouvoir plus ou moins aider le robot à faire la cartographie. Et chaque stratégie qu'on prenait avait des points faibles et des défauts, par exemple au début, nous pensions faire un algorithme de "recherche en profondeur" mais nous nous sommes rendu compte que nous ne pouvions pas l'appliquer sur n'importe quel lieu. Et puisque nous n'avions pas de graphe sur lequel appliquer cet algo, nous nous sommes tourné vers Internet, avec l'espoir de trouver quelque chose qui pourrait nous être utile. Nous avons trouvé beaucoup de bonnes références comme des thèses de doctorat. Mais cela était vraiment long à lire et très compliqué à comprendre. Donc à la fin nous avons fini par écrire un algorithme simple, qui peut sans doute avoir ses propres problèmes. Cela consiste à scanner n'importe quel obstacle que le robot détectait.

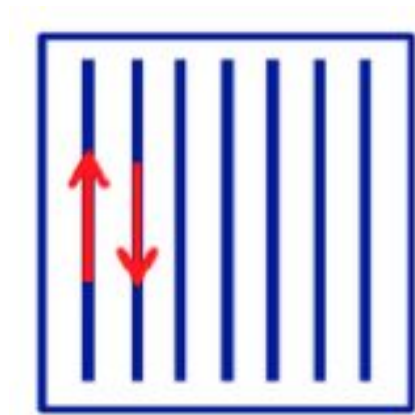


Image explicative (1):  
méthode de parcours de l'obstacle

## VI-Planification

Nous avons essayé de calculer où se trouvent les bords de l'obstacle en cherchant où est-ce qu'on ne le détecte plus pour la première fois après avoir parcouru toute sa surface. Le capteur ultrason fait un tour de 180° afin de pouvoir tout détecter. Après cela, le robot reprend son parcours jusqu'à ce qu'il retrouve un autre obstacle. À chaque détection d'obstacle, les coordonnées sont enregistrées dans un tableau multidimensionnel pour pouvoir les utiliser afin de générer une image 2D qui représente la partie qu'on est en train de parcourir. Générer cette image est facile à faire grâce à la fameuse librairie pygame qui est connue pour être utilisée dans le développement des jeux en python, et heureusement qu'elle nous permet aussi de générer des images.

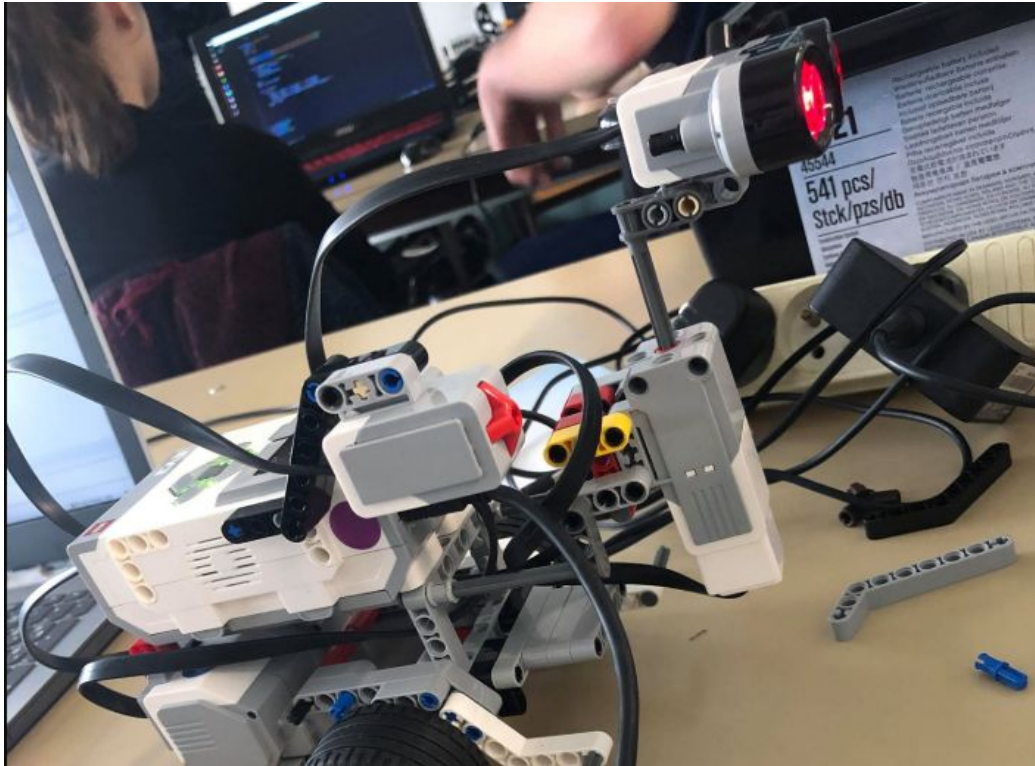


Image explicative (2):

Le robot

## VII-Contrôle

Les actions des grands moteurs sont basées sur les données recueillies par le robot grâce au capteur. Par exemple si le capteur détecte un objet, les moteurs sont automatiquement programmés pour s'arrêter afin de changer la direction du robot pour pouvoir scanner les bords de l'obstacle.

Changer la direction du robot demande en l'occurrence une rotation.

Et pour pouvoir tourner le robot, les deux moteurs doivent tourner dans le sens inverse. Le code ci dessous montre notre approche afin de résoudre le problème pour que le robot puisse tourner sur lui même.

Après avoir scanner les bords de l'obstacle, le robot va retourner au point initial, là où il a trouver l'objet pour la première fois. Vu qu'on a enregistré toutes les coordonnées dans un tableau qui contient tous les points de l'obstacle, on peut très bien détecter le cycle.

Dans le cas où aucun obstacle n'est trouvé, les moteurs vont continuer à aller dans la direction qu'ils ont déjà emprunté.

Le petit moteur a un rôle différent, et est plus facile à programmer puisqu'il n'interagit pas avec les données récoltées. Il se base sur quelle direction le capteur fait face. Le moteur tourne soit à droite soit à gauche pour pouvoir faire un demi cercle pour avoir une vue de 180°.

On a vu bon d'utiliser un code à 3 états: 0,1 ou 2 représentant chacun gauche, droite et devant. C'était plus facile ainsi pour pouvoir voir les choses sans pour autant faire tourner le robot sur lui même. On a juste besoin de faire tourner le capteur pour que le robot puisse détecter les obstacles plus facilement dans toutes les directions et puis enregistrer toutes les coordonnées dans le bon tableau.

Une fois l'exploration finie, on fait appel à la fonction shutdown dans la classe Robot qui est responsable pour arrêter tous les moteurs, même celui qui fait tourner le capteur ultrason.