« Grains d'Usage robotique »

Labyrinthe quelconque et algorithmes de sortie automatique

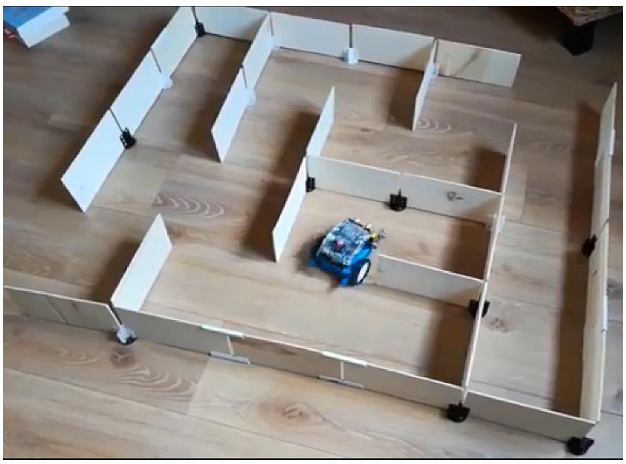
Objectif de la manipulation : concevoir puis implanter un algorithme permettant à un robot de sortir de façon autonome d'un labyrinthe quelconque.

Public concerné : cours d'algorithmique et/ou de programmation (Collèges, lycées et au-delà)

Prérequis : connaissance d'un langage de programmation pour robot mobile (mBot*, Thymio…)

Matériel nécessaire : un robot programmable autonome, un labyrinthe reconfigurable*

**matériels disponibles au SAMS*



1- Le défi :

On dépose de façon **aléatoire** un robot dans un **labyrinthe quelconque** :

• quelles directives donner à notre robot pour qu'il sorte du labyrinthe (à tous les coups) ?

• Comment envisager toutes les situations et lui donner un ensemble de consignes toujours applicables ?

En d'autres termes…quel **algorithme** proposer à un robot pour sortir du labyrinthe de façon autonome ? Tel est le défi à relever ici…

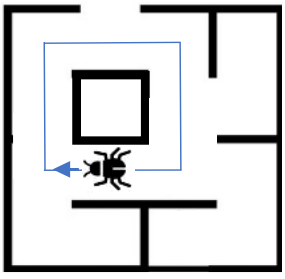
2- Algorithme « des pompiers » :

(Cycle 4 – Lycées)

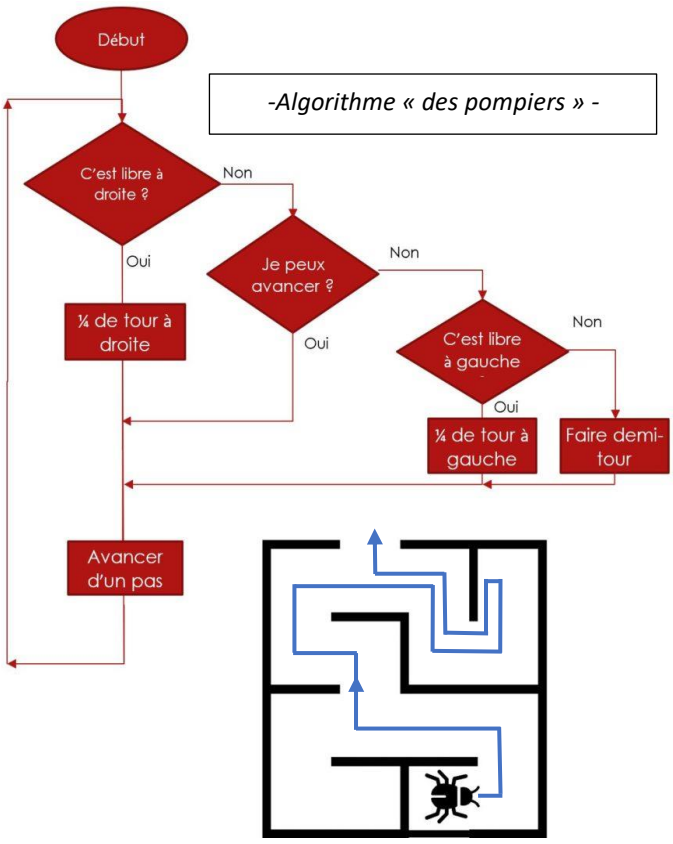
L'algorithme le plus simple à envisager est celui dit « *des pompiers* » : *je longe un mur, toujours le même et -tôt ou tard- je toucherai la sortie* (figure de droite).

Cet algorithme fonctionne…sauf dans un cas particulier : celui ou un ilot isolé (une « tour ») deviendrait un piège autour duquel le robot tournera jusqu'à l'épuisement (voir figure ci-contre).

Piège « de la tour »



Dans tous les autres cas il parviendra à la sortie…mais pas nécessairement de la façon la plus rapide.

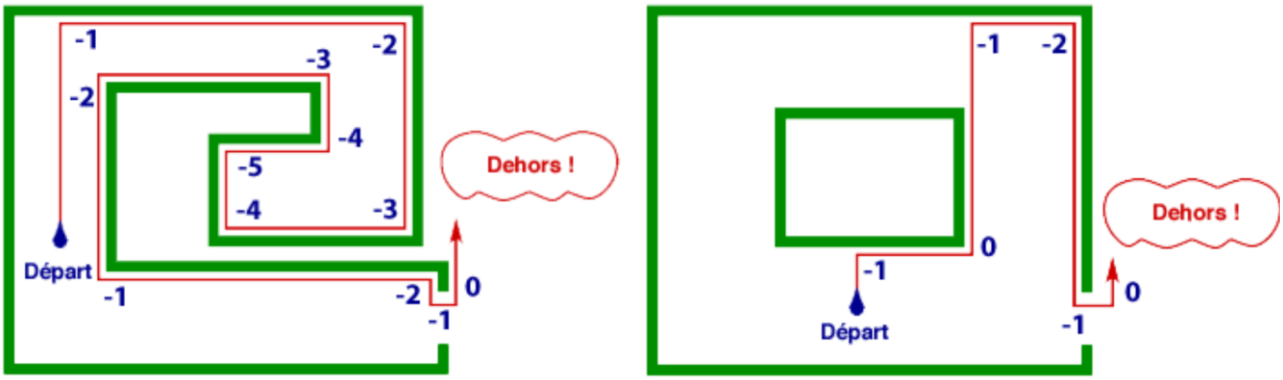


-Algorithme « des pompiers » -

F. Prieur juillet 2020

3- Algorithme de Pledge : *(Lycées – Supérieur)*

L'algorithme de Pledge permet de déjouer le « piège de la tour » évoqué plus haut. Il consiste, toujours en privilégiant un mur (droite ou gauche) à comptabiliser les rotations effectuées tantôt à droite (-1), tantôt à gauche (+1). Si ce décompte passe par 0 alors on part en marche avant jusqu'à un mur et le compte reprend.



Exemple pour un suivi du mur gauche – La « tour » n'est plus un piège - Illustration [1]

A noter que John Pledge n'avait que 12 ans lorsqu'il inventa cet algorithme…

4- Méthodes cartographiées et « par graphes » : *(Lycée – Supérieur)*

D'autres méthodes consisteront à cartographier le labyrinthe inconnu au fur et à mesure de son exploration afin d'éliminer progressivement les culs-de-sac et impasses. Les « graphes » arborescents sont construits et les branches orphelines éliminées [2]. Ces méthodes imposent cependant de savoir se positionner ou se recaler périodiquement. La plupart des robots pédagogiques ne disposant pas d'odométrie ou de dispositifs de positionnement, nous ne les évoquerons que sur dans leur principe.

5- Mise en œuvre pratique et recommandations :

- **Constitution du labyrinthe :** uniquement des angles droits, parois constituées d'éléments simples ou multiples entiers. En effet les mouvements seront basés sur des rotations élémentaires (avance, rotations +90°,-90°,180°)

- **Adaptation du robot mBot :**
  - mBot ne dispose à l'origine que d'un seul capteur de distance frontal. Cela impose pour explorer l'environnement (droite ou gauche) de réaliser une rotation sur place avant de faire un choix. Cette contrainte peut être contournée en installant un second capteur US à droite ou à gauche (selon le côté privilégié), ce capteur sera relié sur l'une des entrées libres de mBot. Un espace se présente à droite ? Le capteur le détectera aussitôt et pourra alors engager la rotation du côté toujours privilégié.
  - Les rotations temporisées sont peu fiables sur un robot ne disposant pas d'odomètrie (mBot…). Une rotation temporisée dépendra en effet pour partie de l'état de charge du robot. Il y a là un intérêt très net à l'emploi du capteur latéral cité plus haut, celui-ci permettra de contrôler la distance au mur rejoint une fois la rotation faite.

- **Aspects liés à la programmation :**
  - Les essais et la mise au point devra se faire en « mode déconnecté » pour éviter les délais induits par la liaison série.
  - Il est hautement souhaitable de décomposer en routines les mouvements élémentaires (quart de tour droite, gauche, demi-tour). Cette façon de faire permettra de structurer le programme et de transcrire l'algorithme de façon directe.

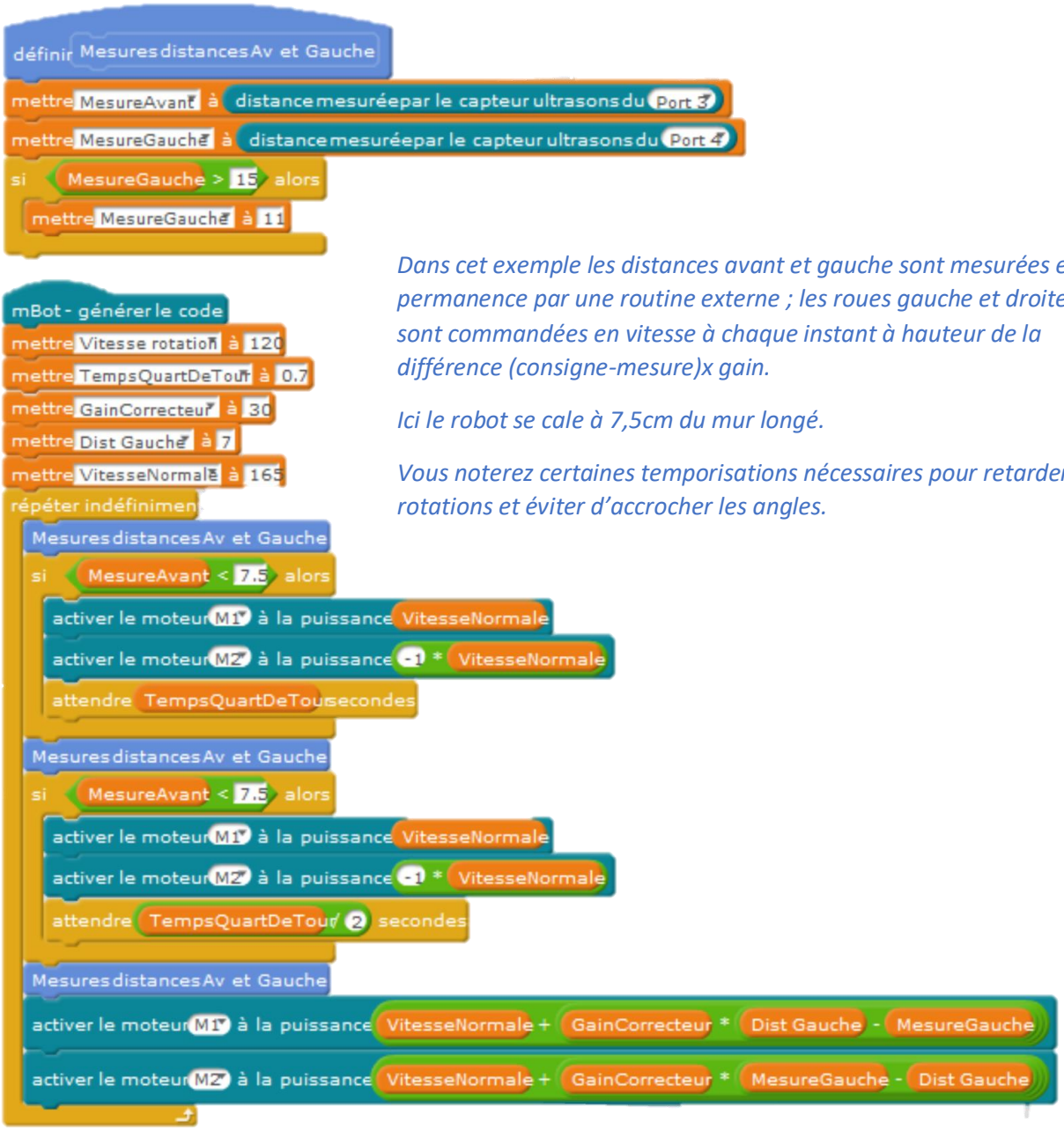
6- Apports pédagogiques :

Selon les niveaux considérés, on pourra travailler avec ce thème bien des aspects de l'algorithmique et de la robotique : algorithmique -bien évidemment-, identification et traitement de cas particuliers, stratégie et optimisation, problèmes de localisation dans l'espace. Commande « proportionnelle », lissage de trajectoire, robustesse à l'environnement et aux perturbations…

Vous pourrez utiliser des labyrinthes automatiques pour amener à la réflexion vos élèves sur des exemples quelconques [(3)].

7- Exemple de programme en Scratch avec commande « proportionnelle »:

La commande « proportionnelle » consistera à asservir la distance au mur de manière à corriger tout défaut d'alignement d'axe en sortie de virage (dérive angulaire liée à une rotation approximative). Cette commande permettra par ailleurs plus de fluidité de mouvement dans les changements de direction en décrivant des arcs de cercles plutôt que des « angles droits ».



Dans cet exemple les distances avant et gauche sont mesurées en permanence par une routine externe ; les roues gauche et droites sont commandées en vitesse à chaque instant à hauteur de la différence (consigne-mesure)x gain.

Ici le robot se cale à 7,5cm du mur longé.

Vous noterez certaines temporisations nécessaires pour retarder les rotations et éviter d'accrocher les angles.

8- Vidéo de démonstration :

C'est l'***algorithme des pompiers*** qui est ici en œuvre « avec priorité à gauche ». Vous noterez (au-delà de la présence d'un sérieux perturbateur…) que la commande des moteurs est ici « proportionnelle » et que le robot arrondi ses trajectoires dans les virages. La robustesse en est améliorée et les trajectoires du robot sont fluidifiées.

Lien : https://urlz.fr/dQnX

9- Ressources et liens utiles :

Les matériels (plaques labyrinthe capteur et mBot) sont disponibles au SAMS (Lycée Joffre Montpellier) pour une mise en œuvre rapide dans votre classe.

N'hésitez à me contacter au besoin : Frédéric PRIEUR frederic.prieur@ac-montpellier.fr

(1) Algorithme de Pledge : un très bon article traitant de la problématique et proposant une simulation : https://interstices.info/lalgorithme-de-pledge/

(2) Labyrinthes et représentations mathématiques : https://fr.wikipedia.org/wiki/Mod%C3%A9lisation_math%C3%A9matique_de_labyrinthe

(3) Générateur de labyrinthes automatiques : http://www.mazegenerator.net/ permet de générer des labyrinthes quelconques.