E.P.I. 4ème

Robot UnoEvo

Thème : **Sciences, technologie et société** **Niveau 4ème 2016-2017**

Sujet : **Robotique**

Intitulé du projet : **Programmer un robot autonome**

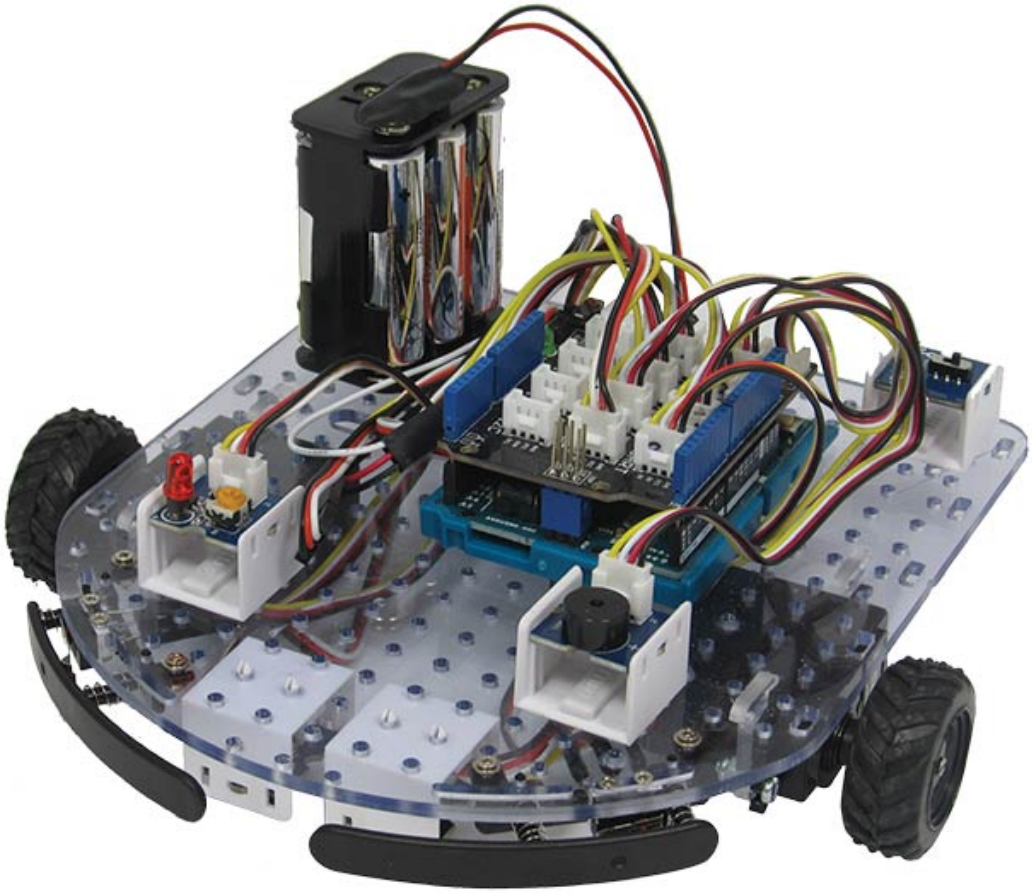
Matières concernées : mathématiques, technologie, physique, anglais

PLAN :

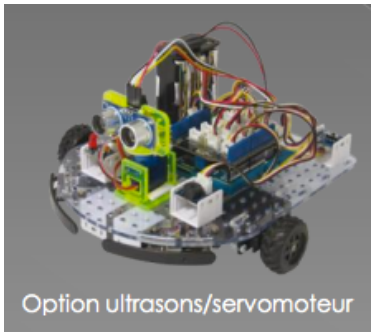
1. Présentation du robot
2. Présentation du projet
3. Simulations du robot sur Scratch
4. Prise en main du robot et programmation des parcours
5. Tests et résultats obtenus lors de l'exercice final
6. Mon opinion personnelle sur le projet et les difficultés rencontrées

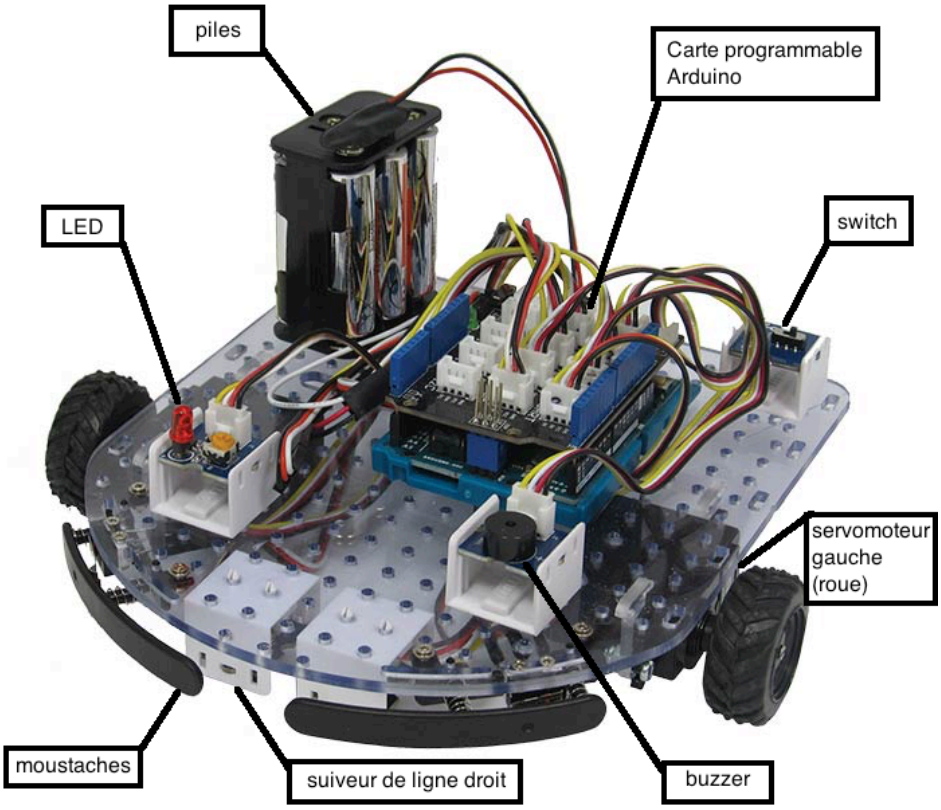
1. Présentation du robot

Voici le robot UnoEvo.



Le robot est évolutif, on peut lui ajouter plusieurs options (suiveur de ligne, capteurs ultrasons, affichage LCD, étage supplémentaire…)





CARTE PROGRAMMABLE
C'est le "cerveau" du robot.

Reçoit des informations,
les analyse,
exécute le programme
et envoie des ordres.

LES CAPTEURS

Relèvent et envoient des informations sur l'environnement.

- les moustaches
- les suiveurs de ligne
- le capteur ultrasons

LES ACTIONNEURS

Reçoivent des ordres et les exécutent pour déplacer le robot.

- les servomoteurs
- la LED
- le buzzer

2. Présentation du projet

Les élèves travaillent en binôme. Quatre robots sont à disposition en classe.

Quels sont les objectifs du projet ?

Le robot est monté mais il doit être programmé avec le logiciel Scratch et/ou Ardublock.

Le robot doit être totalement autonome.

Il aura trois parcours à effectuer, dans un minimum de temps :

- un parcours de suivi de ligne,
- un parcours avec détection d'obstacles,
- un parcours avec franchissement d'obstacles.

Les grandes étapes dans la réalisation de ce projet :

- Prise en main du logiciel Scratch sur des applications variées (*voir en annexe*).
- Simulations des différents parcours sur Scratch avec validation des algorithmes.
- Premières manipulations du robot (connecter, faire avancer/reculer/tourner).
- Tests de programmation effectués sur le robot.
- Validation des parcours et compétition entre les groupes au sein d'un évènement organisé au collège.

Quelles compétences mathématiques sont mises en jeu ?

- **Chercher** : manipuler, expérimenter, tester, décomposer un problème en sous-problèmes.
- **Modéliser** : comprendre et utiliser une simulation, résoudre un problème de proportionnalité.
- **Représenter** : utiliser des situations spatiales (figures géométriques).
- **Raisonner** : mener collectivement un travail, résoudre un problème impliquant des grandeurs variées, utiliser un raisonnement logique.
- **Calculer** : calculer avec des entiers relatifs et des nombres décimaux, contrôler ses résultats, calculer en utilisant le langage algébrique.
- **Communiquer** : expliquer à l'oral et à l'écrit sa démarche et son algorithme, argumenter dans l'échange.

3. Simulations du robot sur Scratch

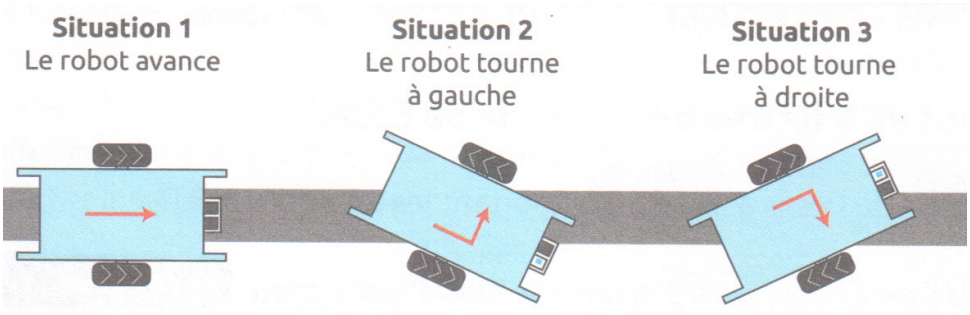
a) Parcours n°1 : le suivi de ligne

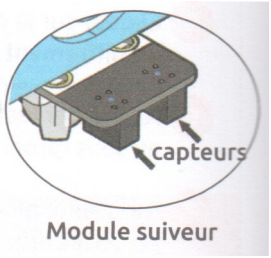
Objectif : écrire un algorithme sur Scratch permettant au robot de suivre, de façon autonome, une ligne noire tracée préalablement de façon quelconque. Le robot devra s'arrêter à la ligne d'arrivée rouge.

Comment ?

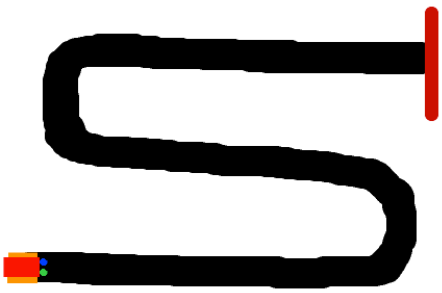
Le robot devra utiliser deux capteurs (vert et bleu) pour suivre la trajectoire noire.

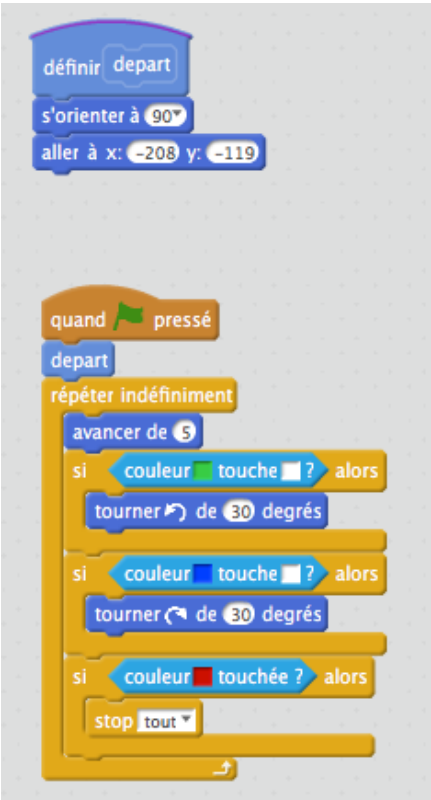
Principe général :



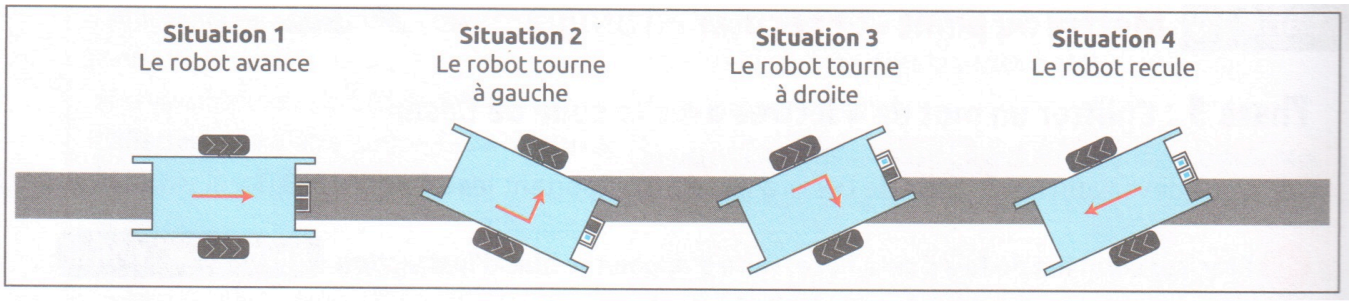


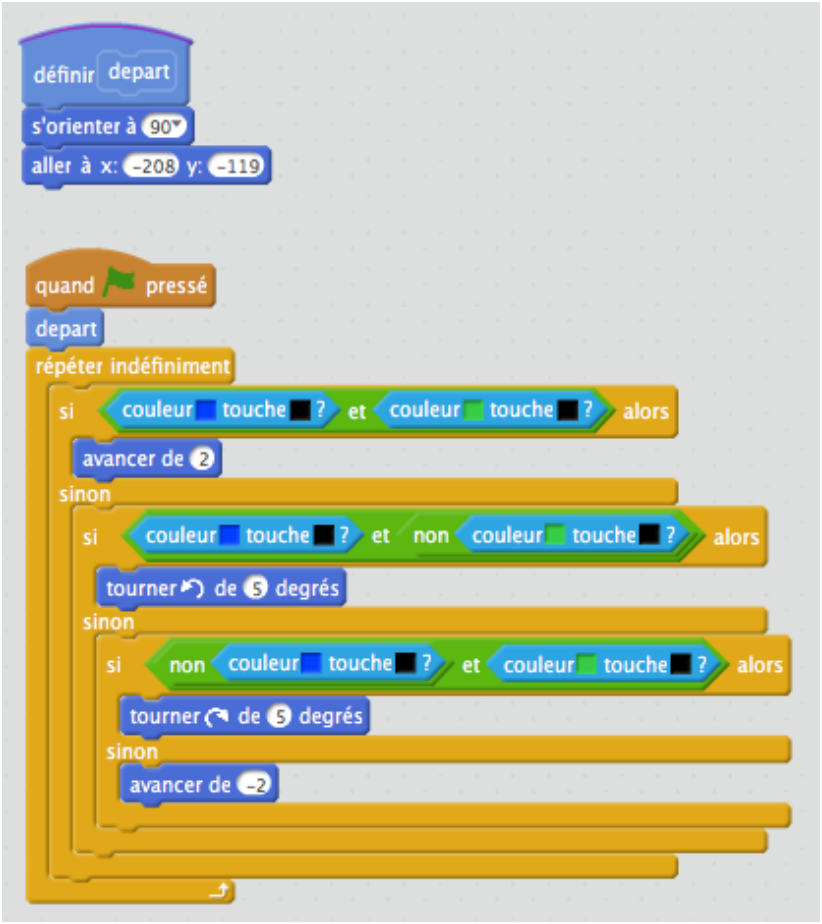
Exemple d'algorithme :





Algorithme amélioré en prenant en compte tous les cas possibles :





Connaissances et compétences mathématiques :

- Utiliser un raisonnement logique (Si …. Alors …. Sinon ….)
- Manipuler des grandeurs variées (angles / vitesse)

b) Parcours n°2 : parcours aléatoire d'une pièce avec détection d'obstacles

Objectif : le robot doit détecter dans une pièce fermée une pastille rouge posée au sol.

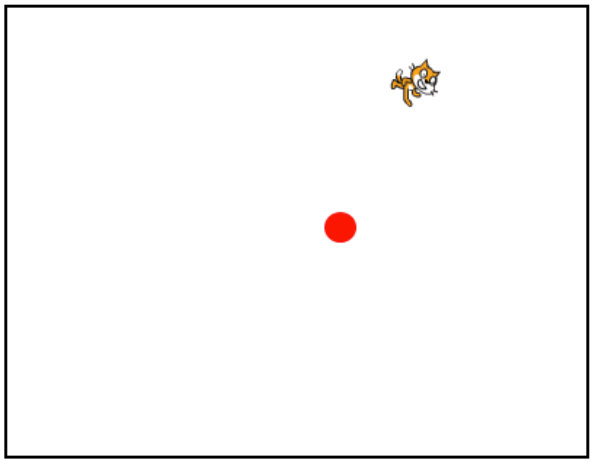
Le parcours sera chronométré.

Comment ?

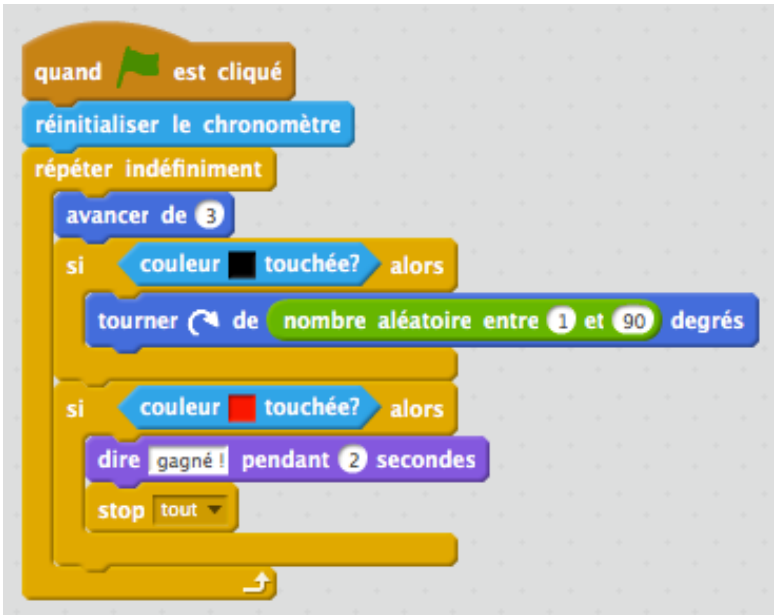
Le robot va avancer et détecter les obstacles (parois verticales délimitant la pièce). Pour éviter les murs, il change de direction aléatoirement.

Principe général :

Le robot va parcourir aléatoirement la pièce jusqu'à détection de la pastille rouge.



Exemple d'algorithme :



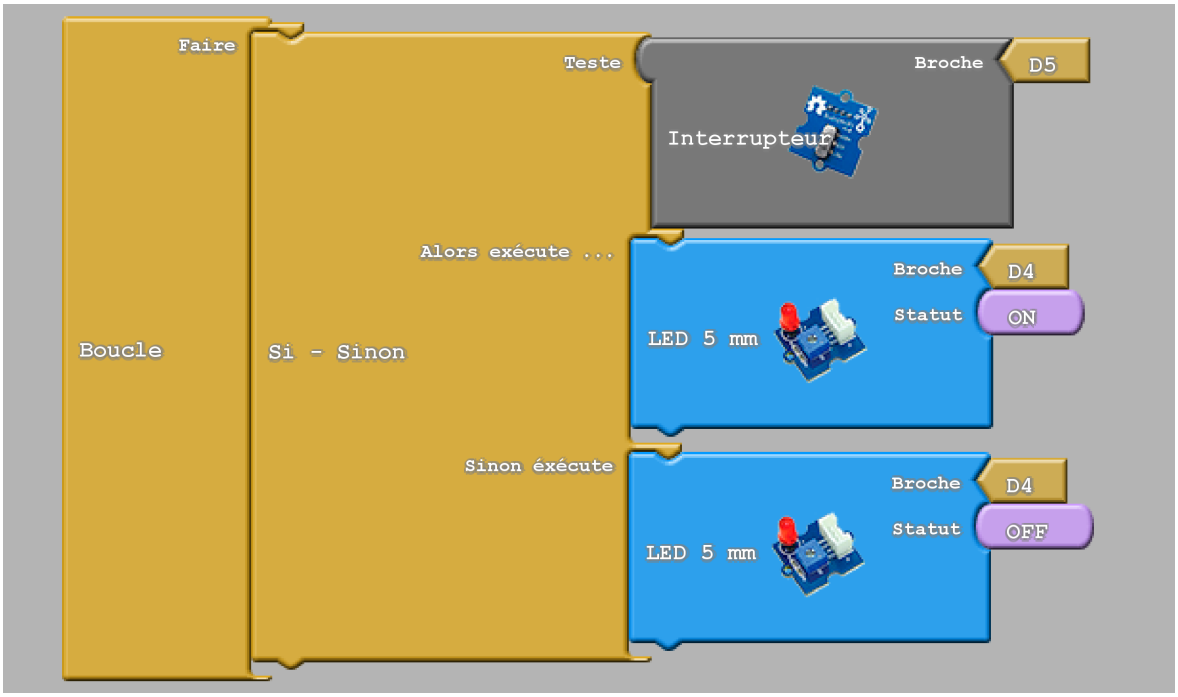
4. Prise en main du robot et programmation

a) Connecter, utiliser le switch, allumer la LED

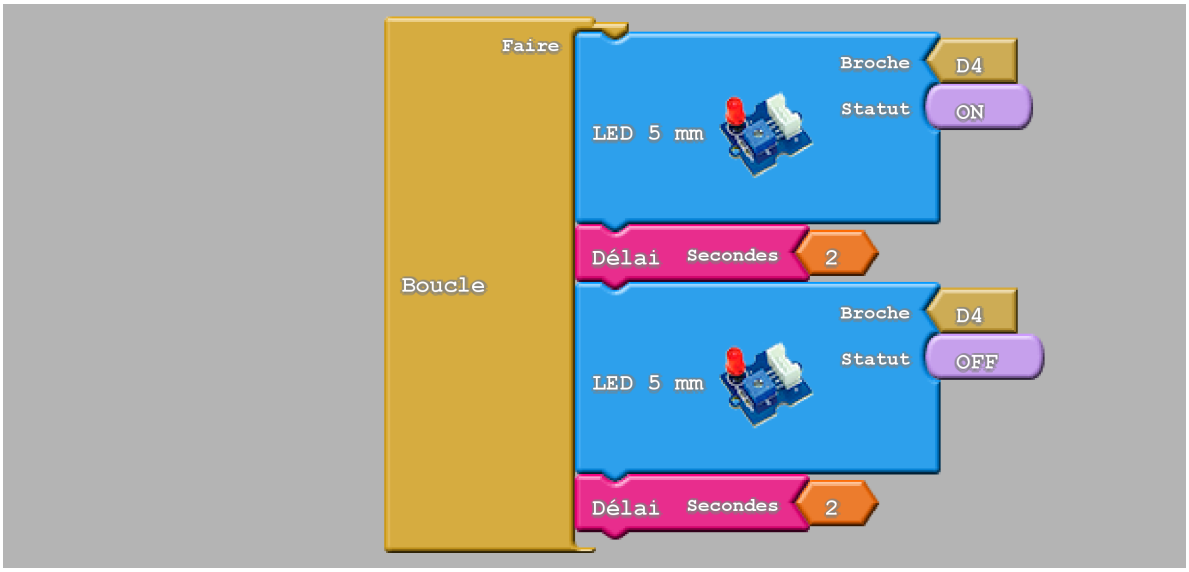
La programmation du robot se fait sur **Ardublock**.
Le robot est connecté à l'ordinateur par câble USB, on charge le programme dans le robot (« téléverser vers l'arduino ») puis on peut débrancher le câble. Le robot est alors autonome.

Les connexions entre les capteurs/actionneurs et la carte Arduino sont précisées en ***annexe 4***.

Voici notre premier programme permettant d'utiliser le switch pour allumer/éteindre la LED :

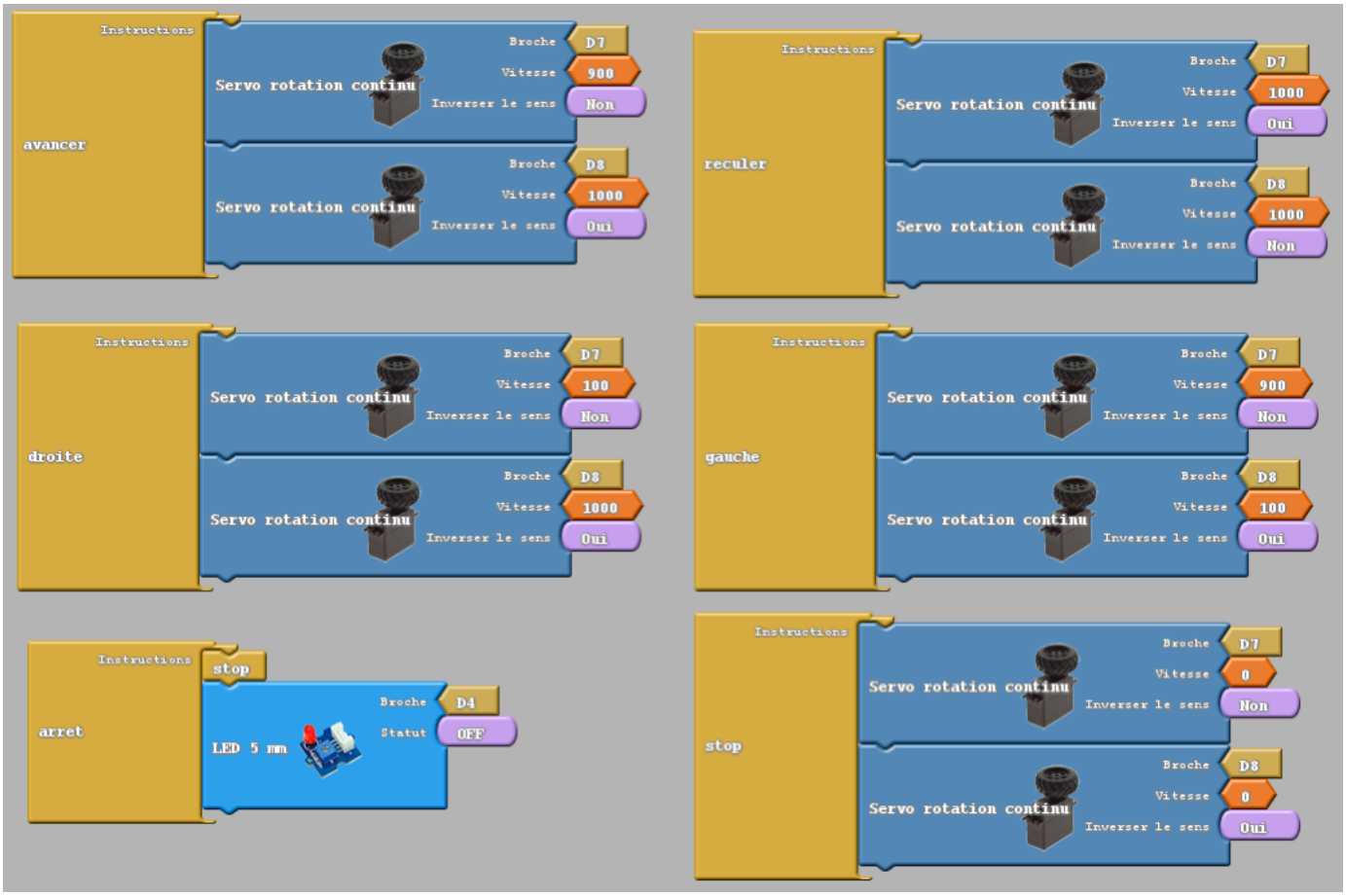


Second programme pour faire clignoter la LED :



b) Programmer les déplacements du robot

Nous avons ensuite créé des sous-programmes pour programmer les déplacements du robot : avancer / reculer / tourner à droite / tourner à gauche / s'arrêter (cf fiche élève en ***annexe 5***).



Principe général :

- Pour avancer droit : les deux moteurs doivent tourner en sens inverse avec la même vitesse.
- Pour reculer : le sens de rotation des deux moteurs doit être inversé.
- Pour s'arrêter : la vitesse des deux moteurs est mise à 0.
- Pour tourner à droite : la vitesse du moteur droit doit être largement inférieure à celle du moteur gauche.
- Pour tourner à gauche : la vitesse du moteur gauche doit être largement inférieure à celle du moteur droit.

c) Contrôler la vitesse du robot // non fait

Connaissances et compétences mathématiques :

- Utiliser une situation de proportionnalité.
- Relation entre vitesse moyenne, distance et temps.

d) Tourner d'un quart de tour ou d'un demi-tour // non fait

Connaissances et compétences mathématiques :

- Représenter une situation par des figures géométriques (cercle).
- Utiliser et calculer le périmètre d'un cercle.

e) Utiliser les capteurs (suiveur de ligne et capteur ultrasons)

(Voir fiche élève en ***annexe 6***)

<u>Utilisation des capteurs suiveurs de ligne</u> :

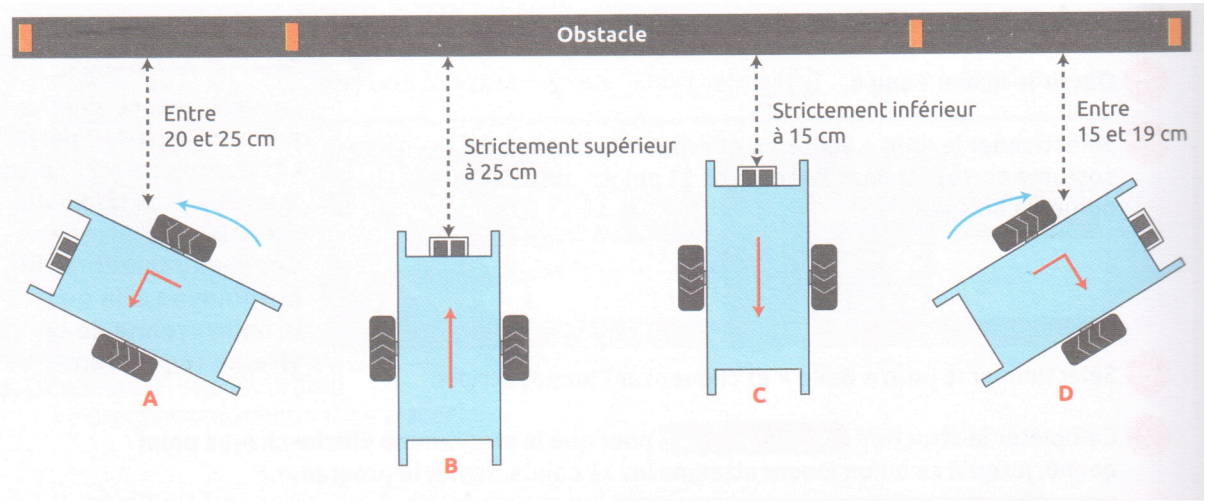
Le robot est équipé de deux capteurs suiveurs de ligne (gauche et droit).

Le capteur envoie l'information VRAI quand il détecte la couleur noire, FAUX sinon.

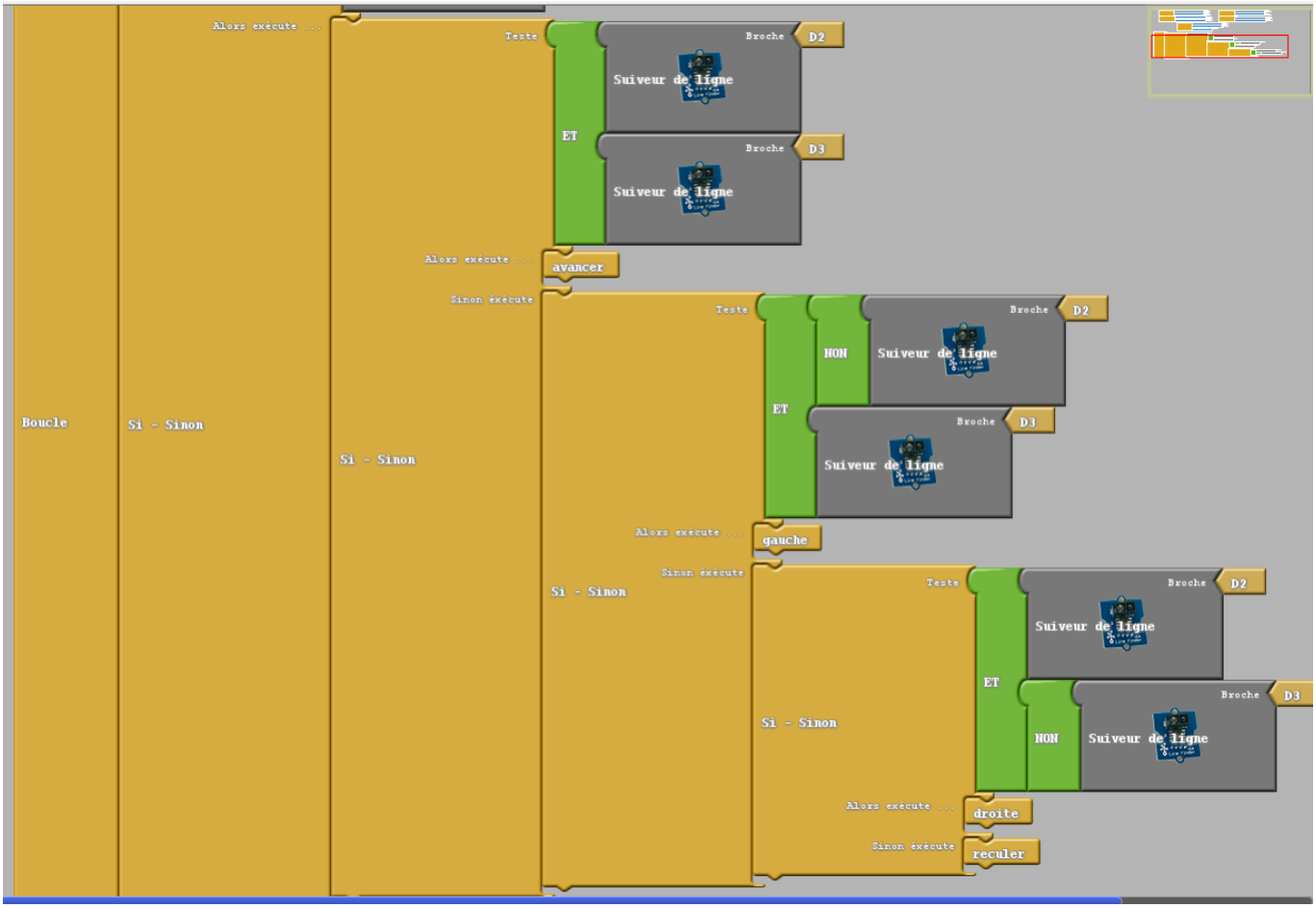
<u>Utilisation du capteur ultrasons</u> :

Le capteur ultrasons envoie la distance (en cm) de l'obstacle qui est devant lui.

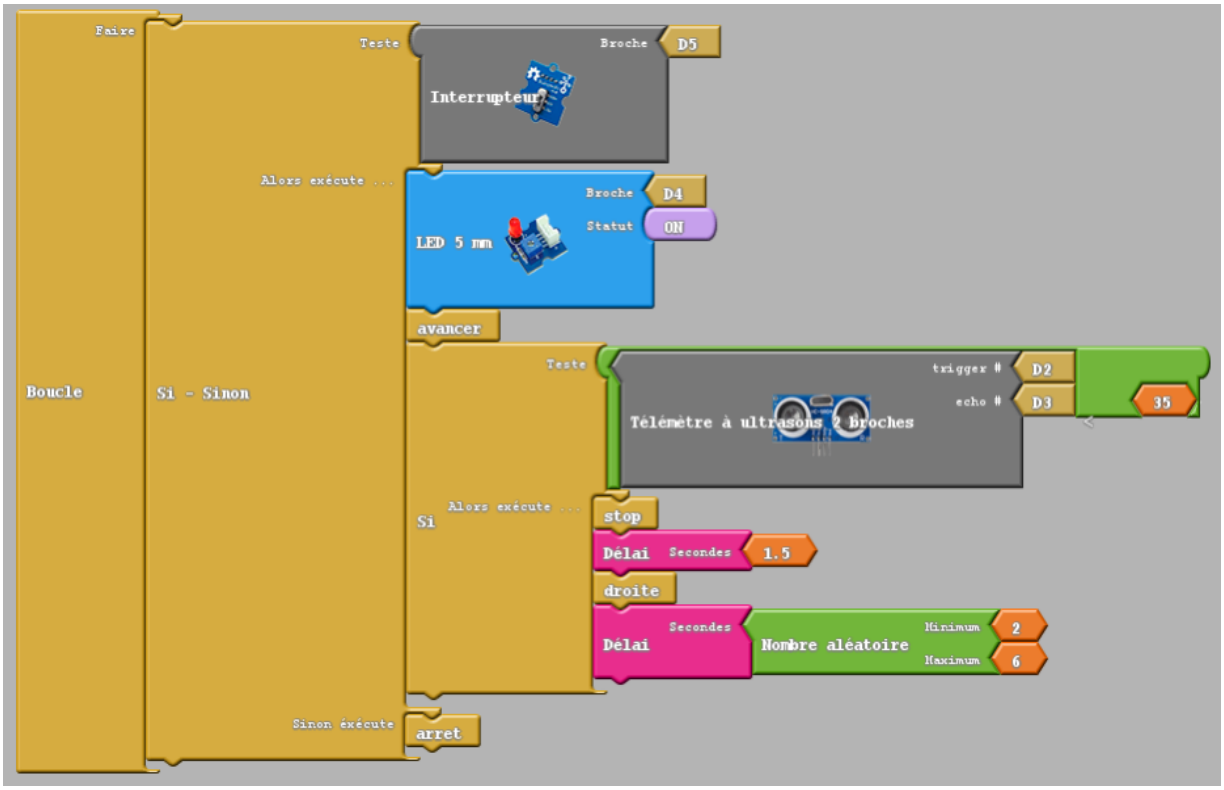
Pour tester ce capteur, nous avons programmé les déplacements du robot en fonction de la distance de l'obstacle comme l'explique le schéma ci-dessous :



f) Rédiger et tester un algorithme pour réaliser le parcours n°1



g) Rédiger et tester un algorithme pour réaliser le parcours n°2



5. Tests et résultats obtenus

6. Mon opinion personnelle sur le projet et les difficultés

Ai-je aimé le projet ?

Pourquoi l'avoir choisi pour mon oral ?

Quelles difficultés ai-je rencontrées ?

Cela m'a-t-il donné envie de réaliser d'autres jeux, d'autres programmes… par moi-même ?

Le projet m'a-t-il donné un aperçu du métier de programmateur, de concepteur de jeux vidéo et d'ingénieur en informatique ?

ANNEXE 1

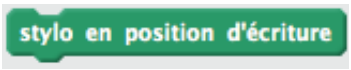
Première initiation à Scratch : tracés de figures géométriques

Objectifs:

- Construire un script qui répond à un évènement.
- Programmer des scripts se déroulant en parallèle.
- Faire appel à un sous-programme.

PARTIE A:

1. En utilisant les commandes 



tracer un carré de côté 100.
2. Ajouter une commande permettant de s'orienter aléatoirement avant de tracer le carré.
3. Insérer la commande 

.
4. Ajouter un bloc > Créer un bloc nommé "Tracer un carré".
5. Remplacer la commande 

par la commande 

.

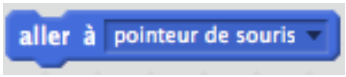
PARTIE B:

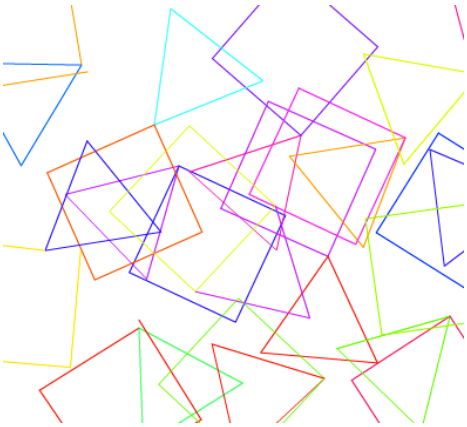
1. Créer un nouveau lutin.
2. Programmer un script qui lui permet de tracer un triangle équilatéral de côté 150.
3. Créer un bloc nommé "Tracer un triangle équilatéral".
4. Cliquer sur  pour exécuter en parallèle les différents scripts des deux lutins.

PARTIE C:

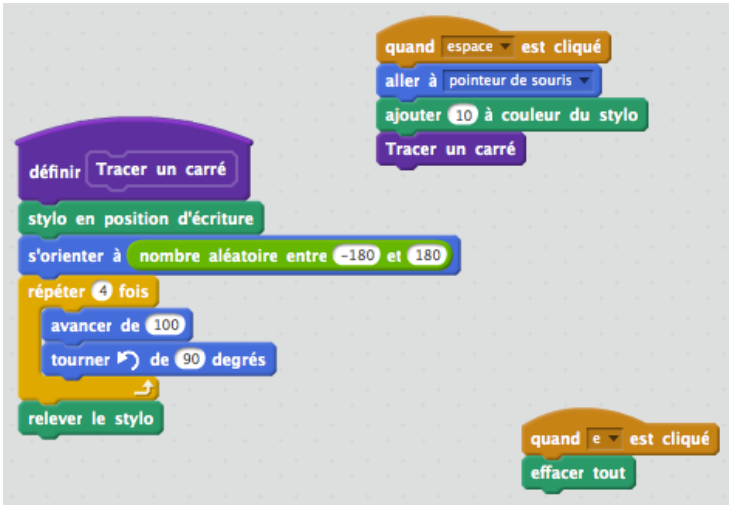
1. Ajouter le script suivant :

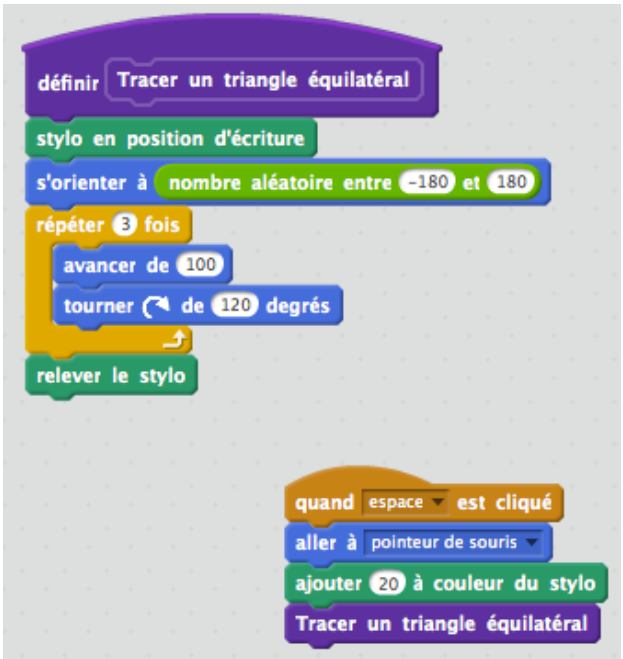
2. Pour chaque lutin, construire un script permettant de réaliser des carrés (ou des triangles) de positions, de couleurs et d'orientations différentes. Pour cela, vous utiliserez les commandes suivantes:





SOLUTIONS :



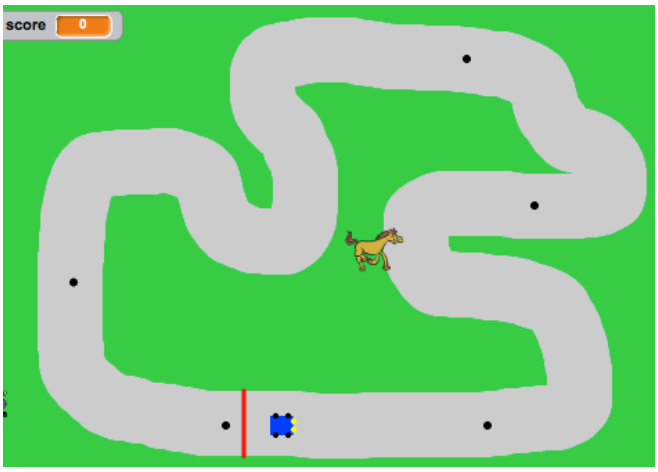


ANNEXE 2

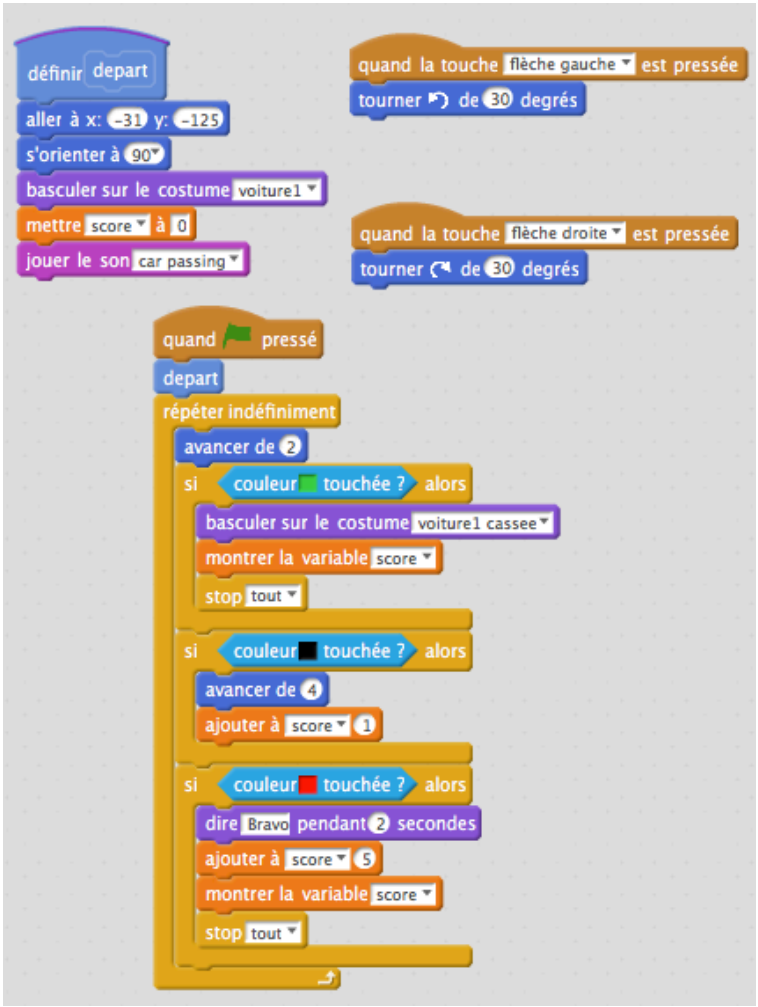
Programmation d'un jeu de course automobile

Objectifs :

- programmation évènementielle,
- conditions (si… alors…),
- introduction d'une variable



But du jeu : le joueur guide la voiture avec le clavier. Il doit réaliser le circuit sans sortir de la piste. Les points noirs servent de « boost ».



ANNEXE 3

Parcourir un labyrinthe de façon autonome

Objectif n°1: le lutin doit trouver, de façon autonome, un chemin lui permettant d'atteindre la ligne d'arrivée rouge sans franchir les obstacles placés aléatoirement.

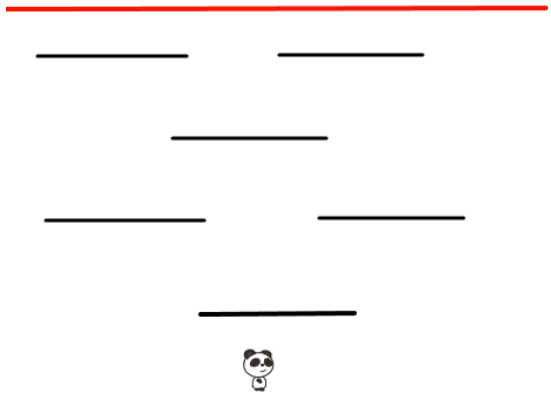
Objectif n°2: le lutin doit contourner les obstacles rencontrés successivement par la gauche et par la droite.

Comment ?

On utilise une variable de direction permettant au lutin de mémoriser la façon dont il a contourné l'obstacle précédemment franchi.

Principe général :

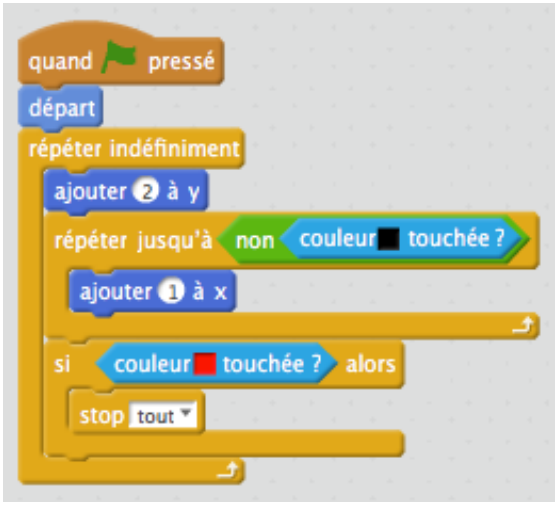
Le lutin avance. S'il touche un obstacle, il se déplace sur la droite (ou sur la gauche) tant que l'obstacle est toujours devant lui.



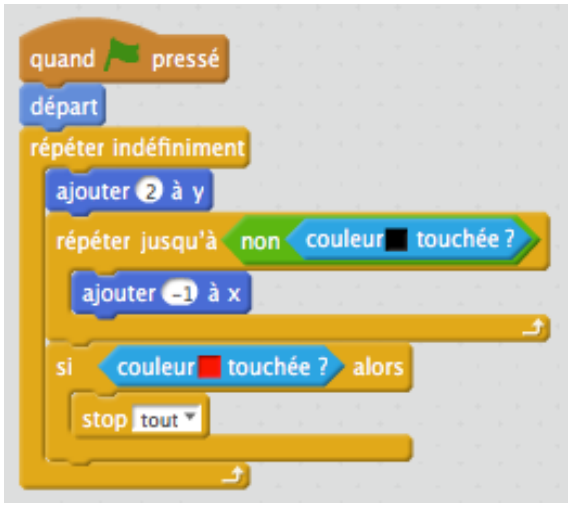
Exemple(s) d'algorithme :

Algorithmes répondant à l'objectif n°1

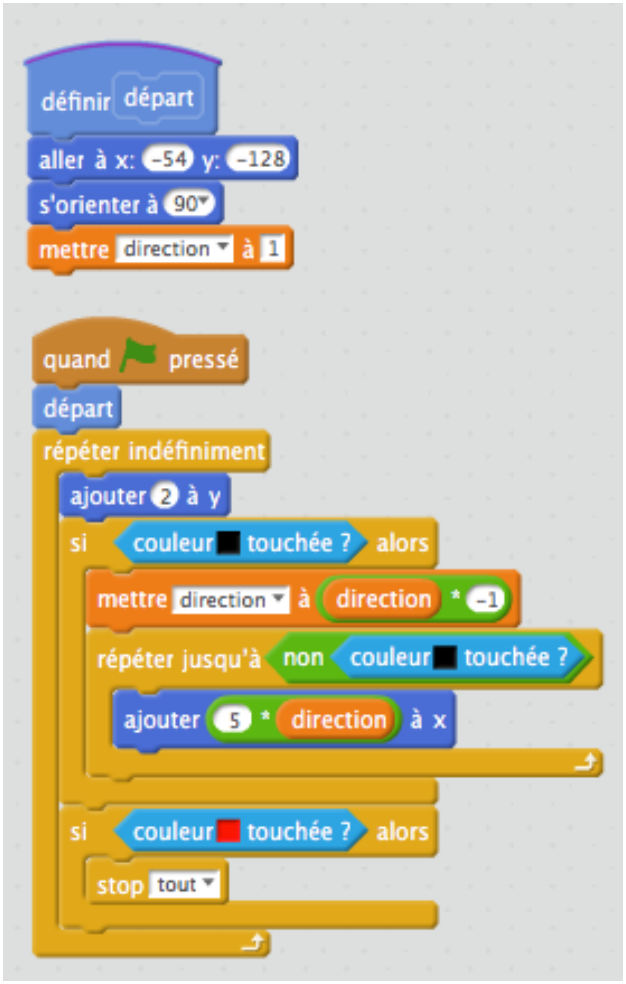
(Contournement par la droite)



(Contournement par la gauche)



Algorithme plus performant répondant à l'objectif n°2



Connaissances et compétences mathématiques utilisées:

- Se repérer dans le plan (abscisses et ordonnées)
- Utiliser un raisonnement logique (Si … alors … / Tant que … / évènement contraire)
- Utiliser une variable (pour mémoriser un état passé)
- Utiliser des entiers relatifs pour se déplacer dans un repère
- Calculer en utilisant le langage algébrique
- Multiplier des nombres relatifs

ANNEXE 4

Connecter les composants (capteurs et actionneurs) à la carte Arduino

Connexions des capteurs :

- Le suiveur de ligne droit sur la broche D2
- Le suiveur de ligne gauche sur la broche D3

Ou

- Le capteur ultrasons sur la broche D2 (D2 TRIG et D3 ECHO)
- Le switch sur la broche D5

Connexions des actionneurs :

- La LED sur la broche D4
- Les servomoteurs sur la broche D7 (D7 moteur droit et D8 moteur gauche)

ANNEXE 5

Utiliser les servomoteurs

Les servomoteurs doivent être branchés en D7 (D7 moteur droit et D8 moteur gauche).

Créer un nouveau document sur Ardublock. Le nommer « deplacements ».

PARTIE 1 : Avancer

1. Ecrire un sous-programme « avancer » permettant au robot d'avancer à la vitesse 1000.
2. Ecrire un sous-programme « arret » permettant au robot de s'arrêter.
3. Ecrire un programme permettant au robot d'avancer lorsque le switch est actionné et de s'arrêter sinon.

Tester puis recopier votre programme.

PARTIE 2 : Reculer

1. Ecrire un sous-programme « reculer » permettant au robot de reculer à la vitesse 1000.
2. Ecrire un programme permettant au robot de reculer lorsque le switch est actionné et de s'arrêter sinon.

Tester puis recopier votre programme.

PARTIE 3 : Tourner à droite

1. Ecrire un sous-programme « droite » permettant au robot de tourner à droite.
2. Ecrire un programme permettant au robot de tourner à droite pendant 4 secondes.

Tester puis recopier votre programme.

PARTIE 4 : Tourner à gauche

1. Ecrire un sous-programme « gauche » permettant au robot de tourner à gauche.
2. Ecrire un programme permettant au robot de tourner à gauche pendant 4 secondes.

Tester puis recopier votre programme.

ANNEXE 6

Utiliser les capteurs

PARTIE A : les capteurs suiveurs de ligne

Les capteurs suiveurs de ligne doivent être branchés en D2 et D3.
La LED est connectée en D4.

Créer un nouveau document sur Ardublock. Le nommer « test capteur suiveur ».

Consigne : écrire un algorithme permettant à la LED de s'allumer lorsque le robot est posé sur une ligne noire (et de s'éteindre sinon).

Tester puis recopier votre programme. Sauvegarder votre travail.

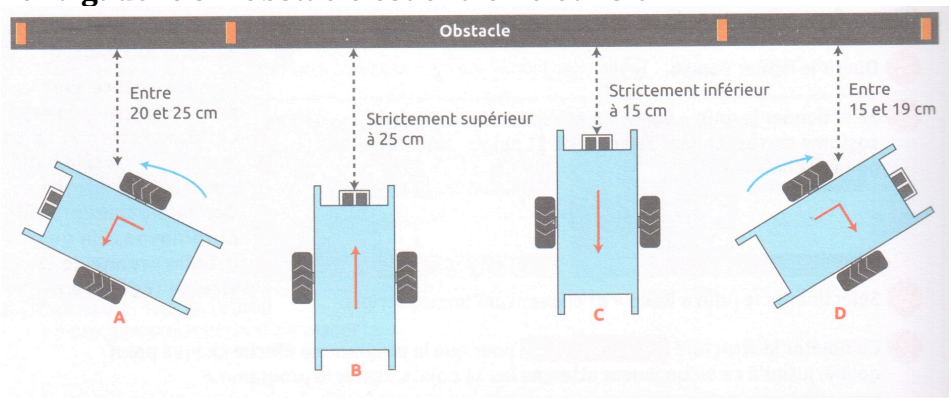
PARTIE B : le capteur ultrasons

Le capteur ultrasons doit être branché en D2.
Les servomoteurs doivent être branchés en D7 (D7 moteur droit et D8 moteur gauche).

Ouvrir le document « deplacements » et l'enregistrer sous un autre nom « test ultrasons ».
Effacer le programme principal en ne gardant que les sous-programmes.

Consigne : écrire un algorithme permettant au robot:

- **d'avancer si l'obstacle est à plus de 25 cm,**
- **de reculer si l'obstacle est à moins de 15 cm,**
- **de tourner à droite si l'obstacle est entre 15 et 20 cm,**
- **de tourner à gauche si l'obstacle est entre 20 et 25 cm.**



Placer votre robot à diverses distances d'un obstacle et tester plusieurs fois votre programme.
Recopier votre programme.