



---

## Programmation de robots

---

### De 3ème à 6ème primaire

---

## Codage et Robotique

---

---

### Annexes

- Thymio\_DECOUVERTE.pdf
- Thymio\_MISSIONS.pdf
- Scratch\_DECOUVERTE.pdf
- Programmer\_Thymio\_avec\_Scratch.pdf
- Programmer\_Thymio\_avec\_Scratch.pdf

---

### Références & liens utiles

- description et interface de programmation thymio : <http://wiki.thymio.org/fr:thymioapi>
  - calibration des thymios : <http://wiki.thymio.org/fr:thymiomotorcalibration>
  - Scratch mit : <https://scratch.mit.edu/>
-

## Séance 1 : Découverte de Thymio

Objectif : Comprendre le fonctionnement de Thymio et réaliser un programme simple à l'aide d'Aseba VPL.

**Notions abordées :** Robotique, programmation, instructions

- 1 - Observation du Thymio et découverte des modes préprogrammés
- 2 - Communiquer avec Thymio

**Niveau :** de 3ème à 6ème primaire

**Durée :** ~ 1h30

**Dispositif pédagogique :** groupes de 2-3 élèves, 1 ordi + 1 Thymio/groupe

### Matériel

- Thymio (+ dongles USB)
- Ordis avec logiciel ASEBA Studio ou Thymio Suite
- Feuilles d'exercice « Comportements prédéfinis » et « missions »
- Ressources : Thymio\_DECouverte.pdf, Thymio\_MISSIONS.pdf
- Lien utiles : description et interface de programmation – <http://wiki.thymio.org/fr:thymioapi>

### 1 - Observation du Thymio et découverte des modes préprogrammés (cfr Thymio\_DECouverte.pdf)

*Distribution et découverte des Thymio (1 par groupe de 2)*

Leur demander d'observer le robot pour essayer de comprendre comment cela fonctionne. Identifier les différentes composantes.

- ➔ Il possède 2 roues, différents capteurs : capteurs de distance : 5 devant et 2 derrière, 2 capteurs de sol, un capteur de température, un accéléromètre (pour déterminer s'il est droit, penché...), des boutons capacitifs, un micro, un haut-parleur, un récepteur infrarouge pour télécommande, des LEDs (non visibles si éteint), un emplacement pour mettre un marqueur.

Leur demander de trouver comment allumer le robot.

- ➔ Allumer le robot (bouton central) : il s'allume en vert. Si l'on touche les boutons haut/bas ou gauche/droite, les couleurs changent. À chaque couleur correspond un mode préprogrammé. Pour activer un mode, sélectionner la couleur puis presser le bouton central.

### Découverte des couleurs "émotions" du Thymio

Déterminer à quel comportement correspond chaque couleur. Essayer de trouver un adjectif pour caractériser ce comportement. Exemple : Programme vert du Thymio = il va vers la main = amical.

Attention : signaler aux participants qu'ils n'ont pas les accessoires nécessaires pour déterminer le comportement bleu clair (donner le circuit imprimé pour le faire). S'il y a du bruit, ils ne pourront pas non plus déterminer le mode bleu foncé.

Comportements prédéfinis

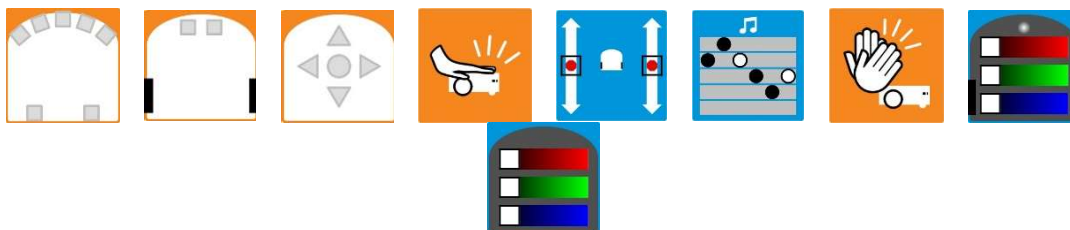
|            |  |
|------------|--|
| Vert       |  |
| Jaune      |  |
| Rouge      |  |
| Violet     |  |
| Bleu clair |  |
| Bleu foncé |  |

➔ Chaque groupe remplit le tableau.

## 2 - Communiquer avec Thymio

*Apprendre le langage de Thymio (cfr Thymio\_DECouverte.pdf)*

Comment communiquer avec Thymio ? Découverte du langage de Thymio via les étiquettes des « actions » et « événements ». Explication de chaque étiquette en vue de les préparer au logiciel.



Exercice de mise en jeu : si je mets l'évènement *Bouton avant* et l'action *Avancer*, alors quand j'appuie sur le bouton avant, le robot avance.

*Connexion du Thymio et programmation (cfr Thymio\_DECouverte.pdf)*

On connecte le Thymio à l'ordinateur via sa clé USB et découvre le logiciel Aseba VPL : explication des différentes touches et de l'interface.

Expliquer le fonctionnement des capteurs de proximité.


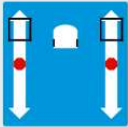

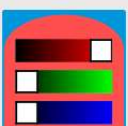

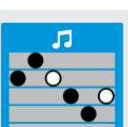


Les missions à réaliser [sélectionner en fonction du niveau de difficulté désiré] :

- Mission 1 : Compléter les phrases par rapport aux programmes proposés.
- Mission 2 : Créer un sélecteur de couleur (à chaque capteur, associer une couleur)
- Mission 3 : Même pas peur ! ; éviter les obstacles.

- Mission 4 : Parcours d'obstacles.
- Mission 5 : Suiveur de ligne.

## Solutions :

### Mission 1 :

| Programme   | Phrase  |
|---|---|
|       | Si appuie sur flèche haut<br>ALORS Thymio avance                |
|       | Si détecte à droite<br>ALORS Thymio s'allume au-dessus en rouge |
|     | Si détecte du noir ou du vide<br>ALORS Joue de la musique       |
|   | Si on tapote le Thymio<br>ALORS Il s'allume jaune en dessous    |

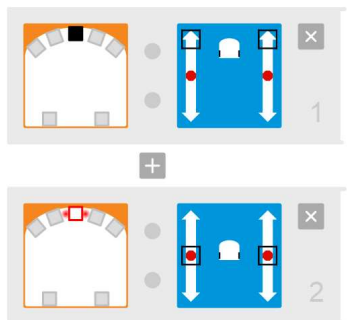
### Mission 2 : Créer un sélecteur de couleur (à chaque capteur, associer une couleur)



Etc...

### Mission 3 : Eviter des obstacles.

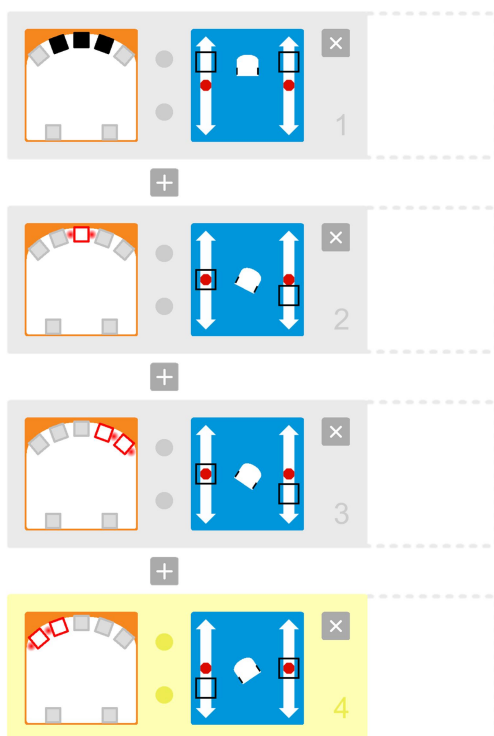
Il faut programmer le Thymio pour qu'il avance quand il ne détecte rien et recule s'il détecte quelque chose. Associer des couleurs à ces actions.



### Mission 4 : Parcours d'obstacles

Réaliser un programme qui permet au Thymio de se déplacer parmi des obstacles. Par exemple, un chemin à parcourir sans toucher les bordures.

Plusieurs solutions sont possibles.



## Séance 2 : Découverte de Scratch

Objectif : Réaliser un programme simple (dessiner un carré) à l'aide de Scratch

**Notions abordées :** Programmation, instructions, mathématiques

- 1 - Rappel de la séance précédente
- 2 - Présentation de Scratch
- 3 - Mise en pratique

**Niveau :** de 4ème à 6ème primaire

**Durée :** ~ 1h30

**Dispositif pédagogique :** groupes de 2-3 élèves, 1 ordi/groupe

### Matériel

- Ordis avec logiciel Scratch Desktop
- Présentation (Scratch\_DECOUVERTE.pdf) + dataprojecteur/écran/...

### 1 - Rappel de la séance précédente

### 2 - Présentation de Scratch

#### Avantages et inconvénients de scratch

- La programmation de base est intuitive. Scratch est un véritable langage de programmation.
- Sur le plan didactique, il permet aussi de réaliser des applications (parfois complexes) très attractives pour les enfants.
- Scratch 3 dispose d'extensions : 'Mindstorm', 'Détection vidéo', 'Synthèse vocale', etc.
- Contrairement à d'autres langages de programmation, il est très 'permissif' et ne génère pas d'erreurs lors de l'exécution ... d'où parfois des difficultés pour 'debugger' les scripts.
- L'éditeur de scratch 3 et l'espace de travail sont moins conviviaux et faciles d'utilisation que ceux de scratch 2.

## Configuration requise pour travailler sur Scratch :

- Utilisation de Scratch directement **en ligne** : <https://scratch.mit.edu/>

- *Configuration requise*

Un navigateur relativement récent est nécessaire pour exécuter Scratch 3.0 :

Google Chrome : version 63 ou ultérieure

Mozilla Firefox : version 57 ou ultérieure

Edge : version 15 ou ultérieure

Safari : version 11 ou ultérieure

- *Sans créer de compte*

Cliquer sur « Créer » en haut à gauche pour accéder à l'interface.

Vous pouvez sauvegarder vos projets sur votre ordinateur (« Fichier » en haut à gauche).

- *Créer un compte*

Cliquer sur « Rejoindre Scratch » en haut à droite et suivre les instructions.

Pour accéder à votre compte par la suite, il suffit de vous connecter depuis la page d'accueil (« Connexion » en haut à droite).

Une fois connecté, vos créations sont automatiquement sauvegardées (vous les retrouvez dans la rubrique « Mes projets » de votre compte). Vous pouvez aussi sauvegarder vos projets sur votre ordinateur, comme expliqué précédemment.

- Utilisation **hors ligne** : Scratch Desktop

- *Configuration requise*

L'installation de Scratch Desktop nécessite :

Windows 7+ / MacOS 10.13+ (High Sierra ou ultérieur) / ChromeOS / Android 6.0+

Environ 500 Mo d'espace disque

- *Installation*

Pour utiliser Scratch sans connexion internet, il faut installer l'application Scratch Desktop.

Télécharger gratuitement Scratch Desktop : <https://scratch.mit.edu/download>

Une fois le téléchargement terminé, cliquez sur le fichier téléchargé pour l'exécuter.

Lors de la première ouverture, l'interface est par défaut en Anglais mais on peut facilement changer de langue en cliquant sur le « Globe » en haut à gauche. Par la suite, Scratch s'ouvrira dans la langue sélectionnée.

### 3 - Mise en pratique (avec en support Scratch\_DECouverte.pdf)

- *Découverte interface et fonctionnement :*

Les élèves suivent sur PC ce que l'enseignant explique en projection, on passe en revue les différents espaces de l'interface, tous les onglets thématiques et les outils.

- *Revue des différents espaces de l'interface :*

On s'arrête sur la structure de la scène et les unités de mesure présentes qui peuvent servir de support à appliquer des notions de maths.

- *Coordonnées dans l'espace avec axes orthonormés  $X,Y$ , défis de déplacement :*

A partir de d'une position  $X,Y = 0,0$  il y a un déplacement possible maximum de 240 unités (correspond au nombre de pixels) pour  $X$  (abscisse) donc  $x$  entre  $-240$  et  $240$  (480 pour toute la largeur en horizontal) ET de 180 unités dans chaque direction pour  $Y$  (ordonnée) donc  $Y$  entre  $-180$  et  $+180$  (360 pour toute la hauteur en vertical).

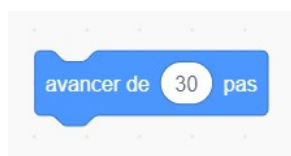
On peut créer de nombreux exercices à partir de ces données. On utilise le stylo pour tracer visuellement le programme de déplacement que l'on crée. On utilise « aller à  $x...$  et/ou  $y...$  » ou si la compréhension est difficile (petites classe) utiliser « Avancer de ... pas ».

- *Exercices*

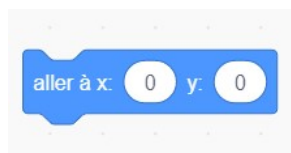
Les élèves réalisent quelques exercices simples pour se familiariser avec l'interface. Au début, corriger au fur et à mesure avec eux au tableau (laisser à ceux qui n'y étaient pas parvenu le temps de faire et tester le programme).

*Instructions de base : Avancer, Tourner*

1) Faire avancer le chat de 30 pas



2) Remettre le chat au centre de la scène



3) Faire avancer le chat de 30 pas et lui faire dire « bonjour »





4) Même chose quand on clique sur le drapeau vert



5) Quand on clique sur le drapeau vert, faire avancer le chat de 30 pas et le faire tourner de 90 degrés



*La boucle : Répéter*

6) Quand on clique sur le drapeau vert : répéter 3 fois faire avancer le chat de 30 pas et le faire tourner de 90 degrés



7) Quand on clique sur le drapeau vert : répéter indéfiniment faire avancer le chat de 30 pas et le faire tourner de 90 degrés

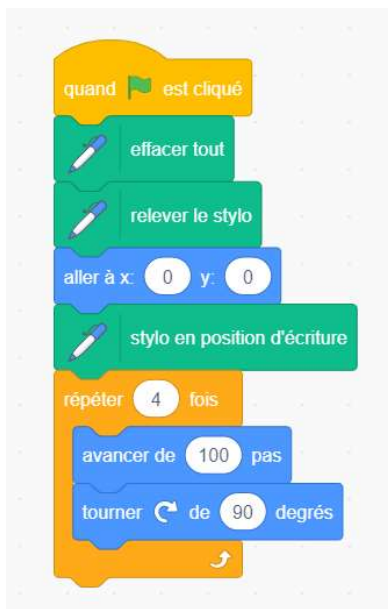


### *Dessiner un carré*

Les élèves doivent programmer le lutin pour qu'il dessine un carré.

Leur faire ajouter l'extension *Stylo* et donner le début du programme (cfr Scratch\_DECouverte.pdf).

Ils essaient de faire des carrés de différentes tailles.



## Séance 3 : Programmation de Thymio avec Scratch (1)

Objectif : Instructions de base : programmer Thymio pour qu'il dessine une forme géométrique donnée

**Notions abordées :** Programmation, instructions, mathématiques

- 1 - Rappel de la séance précédente
- 2 - Scratch pour Thymio : présentation de Thymio Suite
- 3 - Programmation

**Niveau :** de 4ème à 6ème primaire

**Durée :** ~ 1h30

**Dispositif pédagogique :** groupes de 2-3 élèves, 1 ordi + 1 Thymio/groupe

### Matériel

- Thymio (+ dongles USB)
- Ordis avec logiciel Thymio Suite
- Feuilles A3
- Marqueurs pour Thymio
- Projection de l'interface sur écran
- Ressource : *Programmer\_Thymio\_avec\_Scratch.pdf*
- Liens utiles : calibration des moteurs – <http://wiki.thymio.org/fr:thymiomotorcalibration>

**1 - Rappel de la séance précédente :** instructions de base, boucle

### 2 - Scratch pour Thymio : présentation de Thymio Suite (cfr *Programmer\_Thymio\_avec\_Scratch.pdf*)

Pour programmer le Thymio, on utilise Thymio Suite. Les blocs d'instruction pour Thymio se trouvent dans la catégorie « Thymio ».

Présentation des blocs à utiliser pour les instructions de base *Avancer* et *Tourner* + quelques trucs à savoir concernant la programmation de Thymio avec Scratch :

- Distance en mm
- Vitesse : les valeurs peuvent aller de -500 à 500 (unités arbitraires). Une valeur de 500 correspond approximativement à une vitesse linéaire de 20 cm/s.

Pour la suite, afin de minimiser les décalages dans le dessin, il est recommandé d'avoir calibré les moteurs des Thymio pour que le robot aille droit (suivre la procédure décrite ici : <http://wiki.thymio.org/fr:thymiomotorcalibration>) et de fixer la feuille sur la table pour éviter qu'elle bouge pendant que le Thymio dessine.

### 3 - Mise en pratique

Les élèves reçoivent un robot, un ordinateur, une feuille A3 et un marqueur par groupe de 2.

*Adapter le programme Scratch de la séance précédente pour Thymio*

1) Dessiner un carré de 20 cm de côté

*Programmer Thymio pour qu'il dessine un polygone régulier (convexe).*

Expliquer (préalablement ou quand ils ont constaté que ça ne marche pas avec la valeur de l'angle de la forme) comment on détermine l'angle que doit faire Thymio pour dessiner la forme choisie (cfr Programmer\_Thymio\_avec\_Scratch.pdf).

2) Dessiner un triangle équilatéral dont le périmètre vaut 45 cm

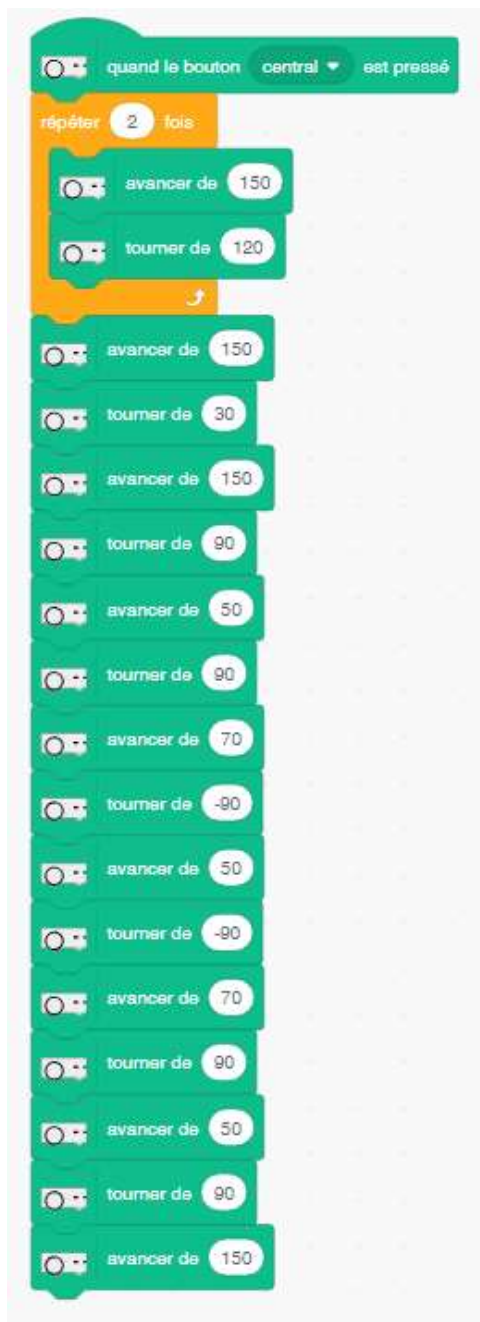
3) Dessiner un hexagone régulier dont le périmètre vaut 60 cm

4) Etc.

Pour aller plus loin avec ces acquis : écrire un programme pour faire un dessin plus complexe comme une maison, un château, ...

Exemples :

Dessiner une maison



## Dessiner un château



## Séance 4 : Programmation de Thymio avec Scratch (2)

Objectif : Instructions conditionnelles : Thymio doit éviter les obstacles ou s'arrêter quand il détecte un obstacle

**Notions abordées :** Programmation, instructions, mathématiques, capteurs

- 1 - Rappel de la séance précédente
- 2 - Instructions conditionnelles et blocs d'instruction pour les capteurs
- 3 - Programmation « Si Thymio détecte un obstacle devant ALORS il tourne »
- 4 - Mise en pratique « Si... ALORS... SINON... »

**Niveau :** de 5ème à 6ème primaire

**Durée :** ~ 1h30

**Dispositif pédagogique :** groupes de 2-3 élèves, 1 ordi + 1 Thymio/groupe

### Matériel

- Thymio (+ dongles USB)
- Ordis avec logiciel Thymio Suite
- Projection de l'interface sur écran
- Ressource : [Programmer\\_Thymio\\_avec\\_Scratch.pdf](#)



## 1 - Rappel de la séance précédente : programmation du Thymio avec Scratch

## 2 - Instructions conditionnelles et blocs d'instruction pour les capteurs

### *Introduction de l'instruction conditionnelle*

Demander aux élèves comment ils diraient à Thymio d'avancer quand il n'y a pas d'obstacle et de tourner quand il y a un obstacle.

→ Si Thymio ne voit rien ALORS il avance tout droit. Si Thymio voit un obstacle ALORS il tourne

Est-ce qu'on peut le dire en une seule phrase ?

→ Si Thymio voit un obstacle ALORS il tourne SINON il avance tout droit (ou inversement).

### *Rappel*

Comment Thymio « voit »-il les obstacles ? Fonctionnement des capteurs ?

### *Utilisation des capteurs avec Scratch (cfr Programmer\_Thymio\_avec\_Scratch.pdf)*

Dans Scratch, on utilise la valeur de l'intensité reçue par le récepteur pour dire quand Thymio voit un obstacle :

Valeur 0 → il n'y a pas d'obstacle

Valeur petite → l'obstacle est loin

Valeur grande → l'obstacle est proche

## 3 - Programmation « Si Thymio détecte un obstacle devant ALORS il tourne »

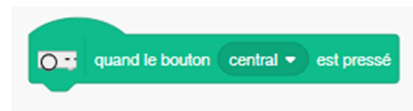
Le programme est réalisé en direct via projection sur écran. Les élèves le font sur leur ordinateur en même temps.



## Sélection des instructions

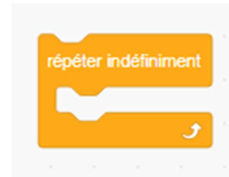
### 1. Instruction de démarrage

- Dans catégorie THYMIO



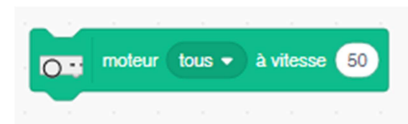
### 2. Boucle

- Dans catégorie CONTRÔLE



### 3. Mettre les moteurs de THYMIO en route pour qu'il avance tout droit

- Dans catégorie THYMIO



### 4. Condition SI ... ALORS ...

- Dans catégorie CONTRÔLE



### 5. Instruction pour laquelle la condition s'applique

Ici : si la valeur du capteur central est supérieure à une valeur choisie

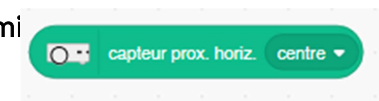
#### 5.1. Bloc « supérieur à »

- Dans catégorie OPÉRATEURS



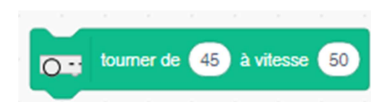
#### 5.2. Bloc « capteurs de proximi

- Dans catégorie THYMIO



### 6. Faire tourner Thymio

- Dans catégorie THYMIO

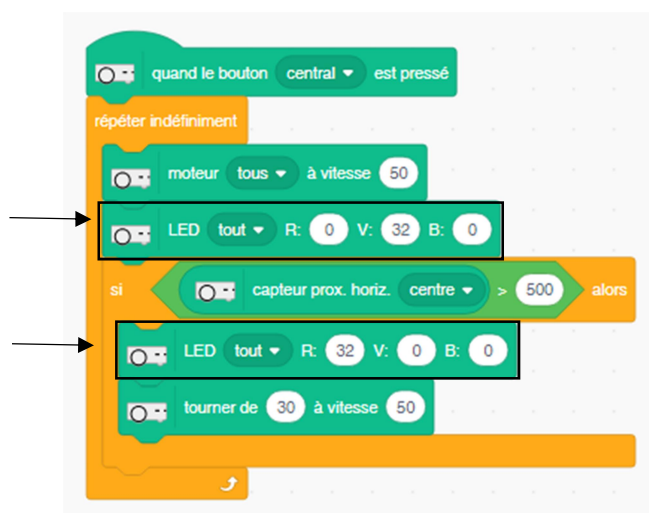


Placer les blocs pour construire le programme :



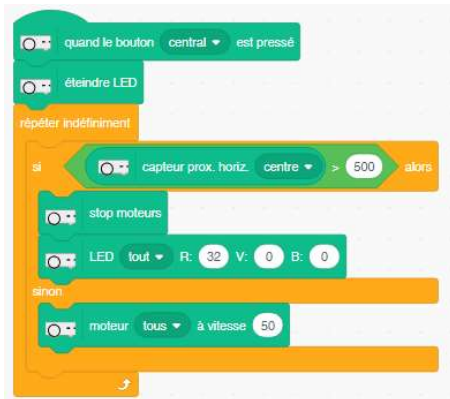
Tester le programme : appuyer sur le bouton central de Thymio et vérifier que Thymio avance tout droit quand il n'y a pas d'obstacle et tourne vers la droite quand il y a un obstacle. Ajuster les valeurs (capteur, angle, vitesses) en fonction du résultat désiré.

On ajoute des changements de couleur : quand Thymio avance, les LEDs sont vertes ; quand Thymio détecte un obstacle et tourne, les LEDs sont rouges.



#### 4 – Mise en pratique « SI... ALORS... SINON... »

Les élèves doivent programmer le Thymio pour qu'il s'arrête et devienne rouge s'il détecte un obstacle : « SI Thymio détecte un obstacle ALORS il s'arrête et devient rouge SINON il avance ».



## Séance 5 : Programmation de Thymio avec Scratch (3)

Objectif : Programmer Thymio pour qu'il franchisse un parcours d'obstacle

**Notions abordées :** Programmation, instructions, mathématiques, capteurs

- 1 - Rappel de la séance précédente
- 2 - Défi : programmer Thymio pour qu'il franchisse un parcours sans toucher les obstacles
- 3 - Révision de l'ensemble des séances

**Niveau :** de 5ème à 6ème primaire

**Durée :** ~ 1h30

**Dispositif pédagogique :** groupes de 2-3 élèves, 1 ordi + 1 Thymio/groupe

### Matériel

- Thymio (+ dongles USB)
- Ordis avec logiciel Thymio Suite
- De quoi faire un parcours d'obstacles (Kapla®, ...)
- [Projection de l'interface sur écran]

**1 - Rappel de la séance précédente :** instructions conditionnelles

**2 - Défi : programmer Thymio pour qu'il franchisse un parcours sans toucher les obstacles**

Les élèves partent du programme « Thymio évite les obstacles » de la séance précédente et modifient les paramètres, complètent les instructions, etc. pour programmer Thymio pour qu'il traverse le parcours en évitant les obstacles.

**3 - Révision de l'ensemble des séances**

Demander aux élèves d'expliquer tout ce qu'ils ont fait/appris au cours des 6 séances.