

### Introduction a la navigation robotique :

Les objectifs :

- Comprendre et mettre en œuvre des techniques de **navigation réactive**.
- Maîtriser les outils de simulation pour l'étude et le développement de techniques de navigation.
- Implanter dans le modèle virtuel et réel, les véhicules de **Braitenberg**.
- **Concevoir et implanter un suivi multi-robot (convoi, Platooning )**

### Navigation :

La navigation robotique est un problème qui concerne le **guidage d'un système robotique vers un objectif**.

Certaines taches de navigation peuvent être effectuées sans faire usage d'une carte. Une des techniques qui permet de naviguer **sans carte** est connue sous le nom de navigation réactive.

Exemples :

- ***S'orienter vers une source de lumière (phototaxie)***
- ***Suivre une ligne au sol***
- ***Naviguer dans un labyrinthe en longeant les murs***
- ***Nettoyer le sol d'une chambre en suivant un chemin aléatoire***

En résumé, un robot effectue une **navigation réactive en associant une réaction à son environnement** par exemple, l'intensité de lumière, la position relative à une ligne au sol ou sa distance par rapport à un mur.

# La Navigation réactive

## Véhicules de Braitenberg

Famille de robots qui se caractérise par **une mise en relation directe entre les capteurs et les actionneurs**. Ces robots n'ont aucune représentation de l'environnement dans lequel ils opèrent et ils n'ont en construisent aucune.

Ces robots associent les actionneurs et les capteurs telles que certaines créatures du vivant. La représentation d'un robot de **Braitenberg** est pourvue de capteurs de lumière reliés directement aux moteurs.

Plus de lumière alors plus vite est actionné le moteur.

Lien direct  $\rightarrow v_L = k I_L \quad v_R = k I_R$

Lien croisé  $\rightarrow v_L = k I_R \quad v_R = k I_L$

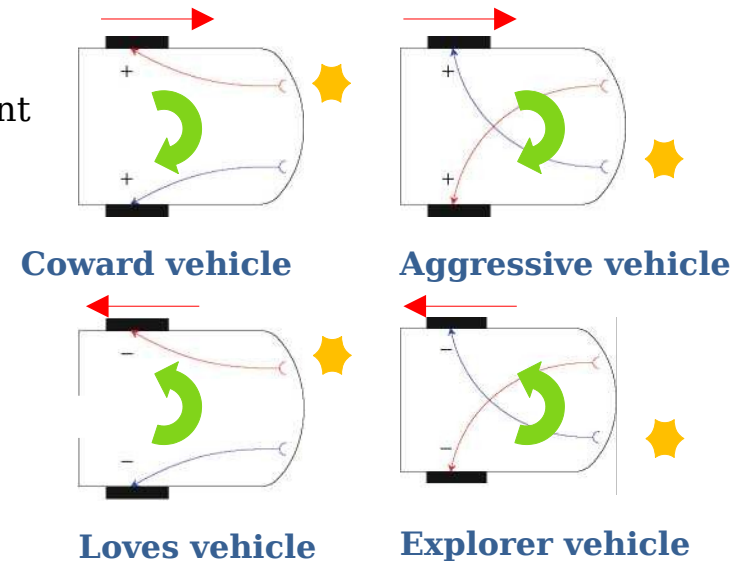
**Coward** : Evite une source de lumière (Peureux, lâche)

**Agressif** : Se dirige vers une source de lumière

**Quelques exemples d'adaptation aux détecteurs d'obstacle**

**Timide** : Ce type de véhicule avance à condition de ne détecter aucun obstacle devant. Si un obstacle est détecté, **alors le véhicule s'arrête**.

**Indécis** : **Le véhicule avance** tant qu'aucun obstacle n'est détecté. Si un obstacle est détecté alors **le véhicule recule**. A la distance limite de détection, le véhicule aura tendance à décrire un mouvement oscillatoire (succession de mouvements vers l'avant et vers l'arrière).



# Navigation réactive

## Machine à états finis

Machine à États Finis (FSM - Finite State Machine) est composée par un ensemble fini d'états et d'un ensemble de transitions définies entre deux états.

### Éléments composant une machine à états

- Les états

- Les transitions : sont composées par un couplage condition/action

- Condition : Élément indiquant la transition de l'état
- Action : procédure qui commence dès la validation de la condition

### Convention graphique

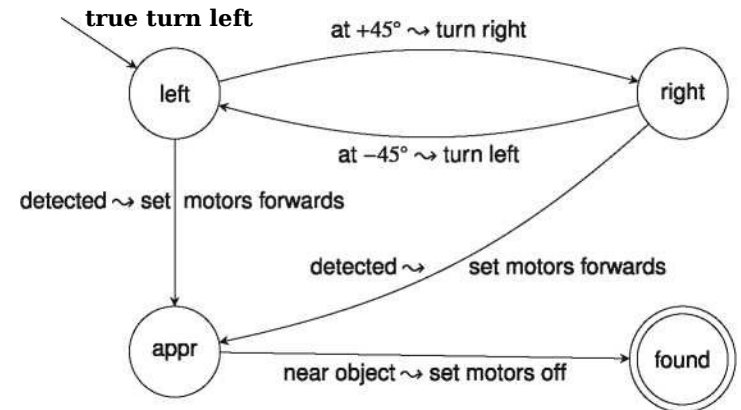
Un état est représenté par un nœud labellisé

Une flèche rentrante indiquera l'état initiale de la machine

Une transition se représente par un lien avec un flèche indiquant le sens de la transition.

Les actions d'une transition n'est pas (ne doit pas être) un processus **bloquant**.

### Machine à états : Recherche et rapprochement



**left** = robot turning left to search  
**right** = robot turning right to search  
**appr** = robot approaching object  
**found** = robot found object

# Projet robotique

## Cahier des charges

### Le projet se structure en trois phases :

- **Phase 1** : Programmation en Python et virtualisation d'un système robotique sur Webots
- **Phase 2** : Implantation des véhicules de **Braitenberg** et consolidation d'un suivi de convoi virtuel
- **Phase 3** : Développement et implantation d'une démonstration de suivi d'un convoi de trois robots physiques (**Thymio**)

# Projet robotique

## Cahier des charges

### Phase 1

Cette phase doit être abordée et consolidée en autonomie à l'aide d'une synthèse de ressources :

- Exercices à exécuter : introduction à python (Exercices 1 à 6 sous environnement virtuel python)
- Des informations disponibles sur internet  
Lien [openclassrooms](https://openclassrooms.com) - Apprenez à programmer en Python (Partie 1)

Des exemples fournis dans les contrôleurs du robot Thymio II

Voir fichiers du site du projet :

- **Simulation\controllers\thymio\_exemple\thymio\_exemple.py**
- **Simulation\controllers\thymio\_manual\_control\thymio\_manual\_control.py**

# Projet robotique

## Préparation de l'environnement de développement

- 1- Ouvrir un terminal sous linux
- 2- Créer un environnement de travail  
`mkdir Projet_Robotique`
- 3- Environnement virtuel python  
`python3 -m venv .venv`  
`source .venv/bin/activate`
- 4- `mkdir seance_1 seance_2 seance_3`
- 5- `# Initialiser Git`  
`git init`  
`# Créer un fichier .gitignore (pour éviter fichiers inutiles comme le venv)`  
`echo "venv/" > .gitignore`  
`# Premier commit pour fixer la structure`  
`git add .`  
`git commit -m "Initialisation de la structure du projet : séances 1, 2 et 3"`  
`#Créer votre dépôt Github`  
`git remote add origin git@lien vers votre repository`  
`git push -u origin main`  
`# Fin Séance i`  
`git add .`  
`git commit -m "Rendu séance i" # i=1,2 ou 3`

# Projet robotique

## Cahier des charges

### Phase 2

Cette phase doit être abordée et consolidée sur webots avant de passer à l'implantation sur robot réel.

L'adaptation tient particulièrement compte des capteurs et des actionneurs disponibles dans le robot.

Pendant cette phase chaque groupe devra faire l'implantation de **trois différents véhicules** (personnalités) dans le contrôleur du Thymio.

Les véhicules à implanter vous seront communiqués individuellement.

### Livrable de la phase 2 : (60% de la note finale)

- Un script contrôleur intégrant trois différents véhicules de Braitenberg. Le script permettra à l'aide de touches du clavier de choisir la personnalité du robot (35%)
- Un environnement de simulation (world) permettant de valider l'implantation (10%)
- Une vidéo illustrant le fonctionnement de chaque véhicule de Braitenberg implanté sur robot (15%)

Véhicule	Description
Timide	Ce type de véhicule avance à condition de ne détecter aucun obstacle devant. Si un obstacle est détecté, alors le véhicule s'arrête.
Indécis	Le véhicule avance tant qu'aucun obstacle n'est détecté. Si un obstacle est détecté alors le véhicule recule. À la distance limite de détection, le véhicule aura tendance à décrire un mouvement oscillatoire (succession de mouvements vers l'avant et vers l'arrière).
Paranoïaque A	Quand le robot détecte un objet, il avance. Quand aucun obstacle n'est détecté alors il tourne à gauche.
Paranoïaque B	Quand le robot détecte un objet devant, il avance. S'il détecte un objet sur la droite, il tourne à droite. S'il détecte un objet sur la gauche alors il tourne à gauche.
Paranoïaque C	Quand le robot détecte un objet, il avance. Quand aucun obstacle n'est détecté alors il tourne en alternant de gauche à droite à chaque seconde.
Obstiné A	Ce type de véhicule avance à condition de détecter un objet devant. Si un objet est détecté à l'arrière, alors le véhicule recule.
Obstiné B	Ce type de véhicule avance à condition de détecter un objet devant. Si un objet est détecté à l'arrière, alors le véhicule recule. Si aucun objet n'est détecté alors le véhicule s'arrête.
Anxieux	Quand aucun objet n'est détecté à gauche, le robot tourne à gauche. Si un objet est détecté à gauche alors le robot tourne à droite. Ce type de personnalité permet au robot de longer un mur sur la gauche.

# Projet robotique

## Phase 3

Implantation sur robot des modèles de navigation programmés et validés sur Webots.

Utilisation de la bibliothèque :

<https://pypi.org/project/thymiodirect/>

### Communication par bus série

Adaptation des modèles pour usage réel.



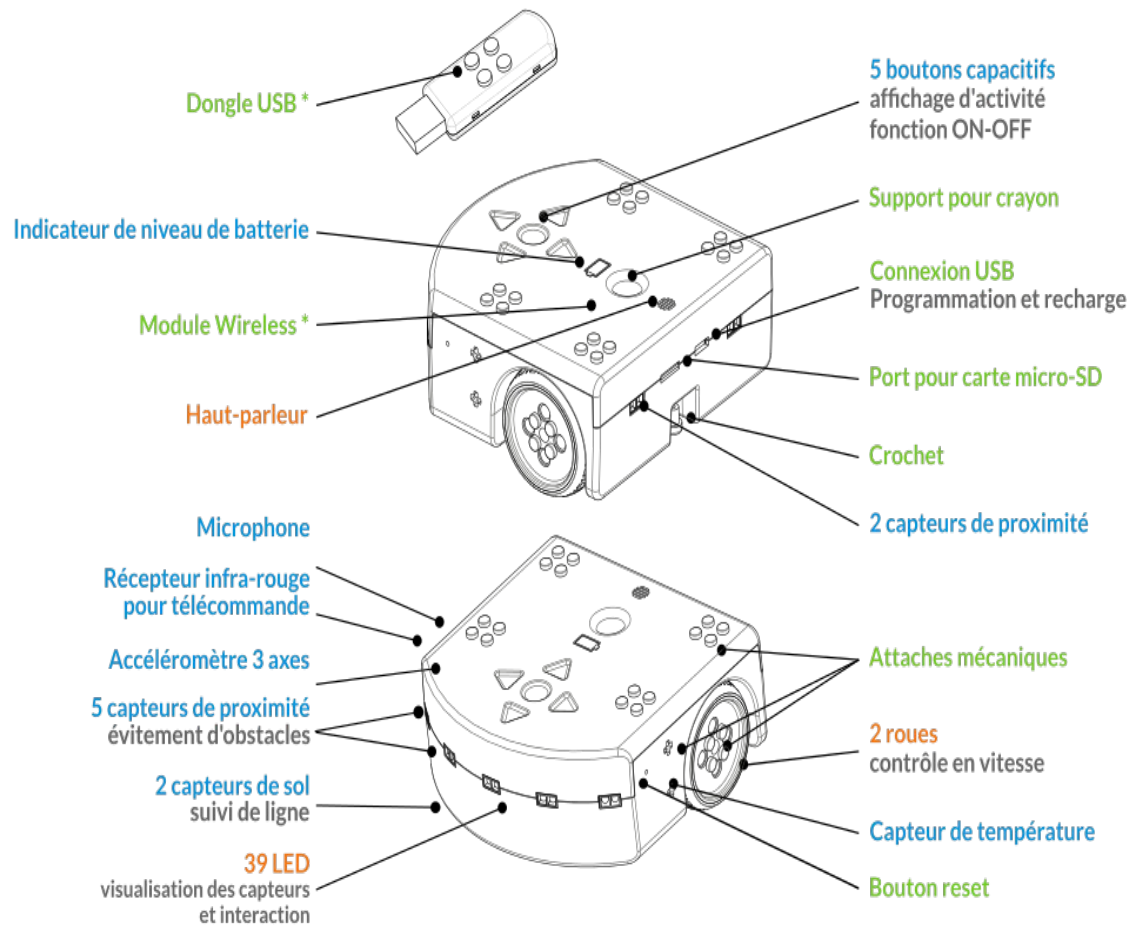
### Livrable de la phase 3 : (40% de la note finale)

- Les scripts des contrôleurs des trois robots du convoi (20%)
- Un environnement de simulation (world) permettant de valider l'implantation du convoi de robots (10%)
- Les scripts et une vidéo illustrant le fonctionnement du suivi du convoi sur robot réel (10%)





# Projet robotique



\* disponible uniquement avec Wireless Thymio



# Projet robotique

## Caractéristique capteur de proximité (Thymio)

