

# Les systèmes autonomes

---

Matthieu Martel



**UPVD**  
Université de Perpignan *Via Domitia*

# 1. Principes de fonctionnement

---



# Systemes autonomes (ou embarqués)

---

**Systemes autonomes** : Systemes informatiques (logiciel+matériel) conçus pour être intégrés dans des objets industriels

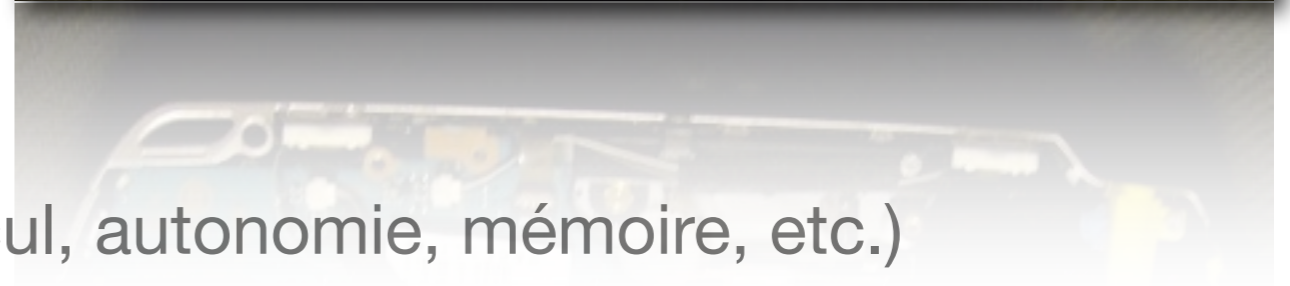
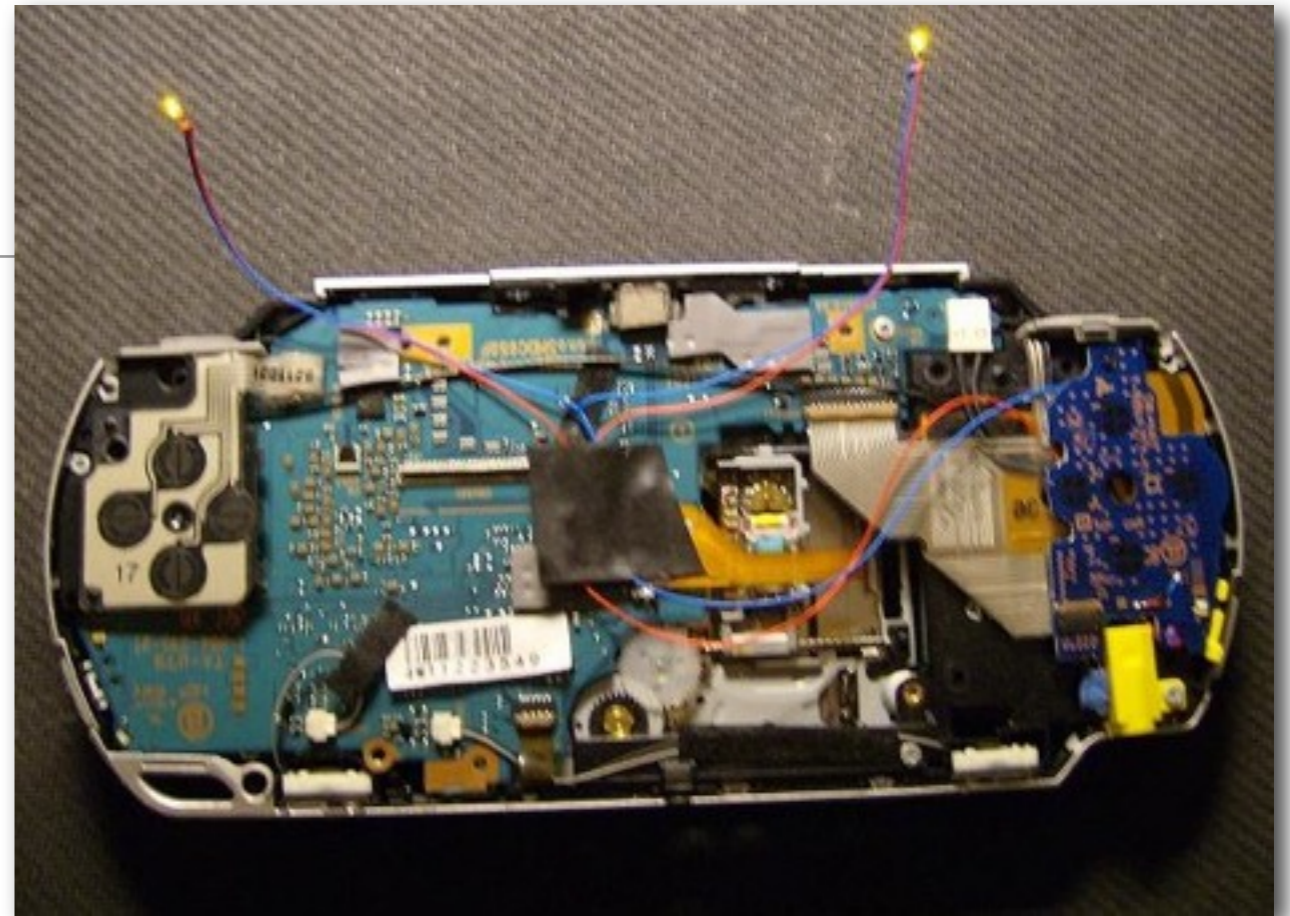


# Systemes embarqués

---

## Quelques caractéristiques :

- Systemes électroniques et informatiques autonomes, sans entrées ni sorties standard (clavier et écran)
- Capacités limitées (puissance de calcul, autonomie, mémoire, etc.)
- Difficilement modifiables après livraison
- Contraintes de coûts :
  - coût de développement (petites séries, ex : avion)
  - économies d'échelle (grandes séries, ex : téléphone)



# Exemple : le kit robot LEGO Mindstorms NXT

---

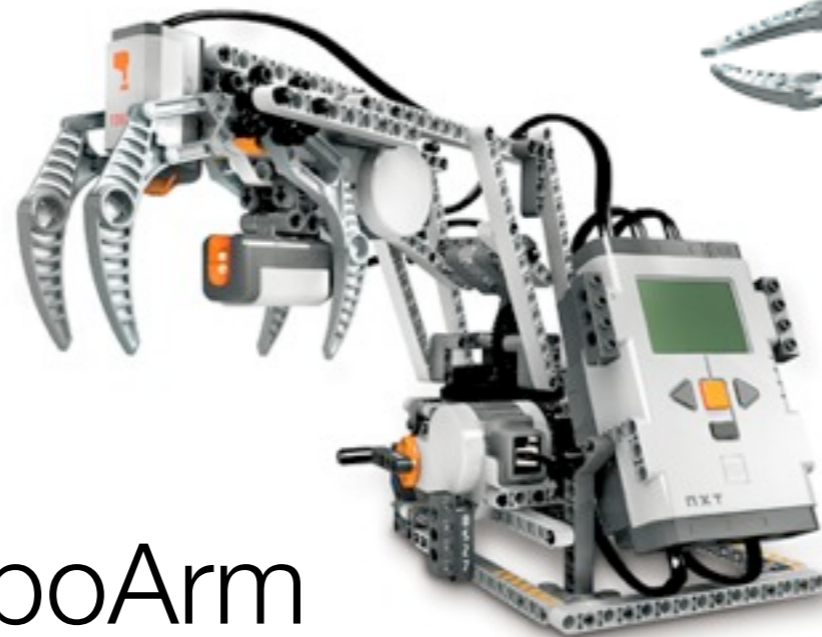
Alpha Rex



Tribot



RoboArm



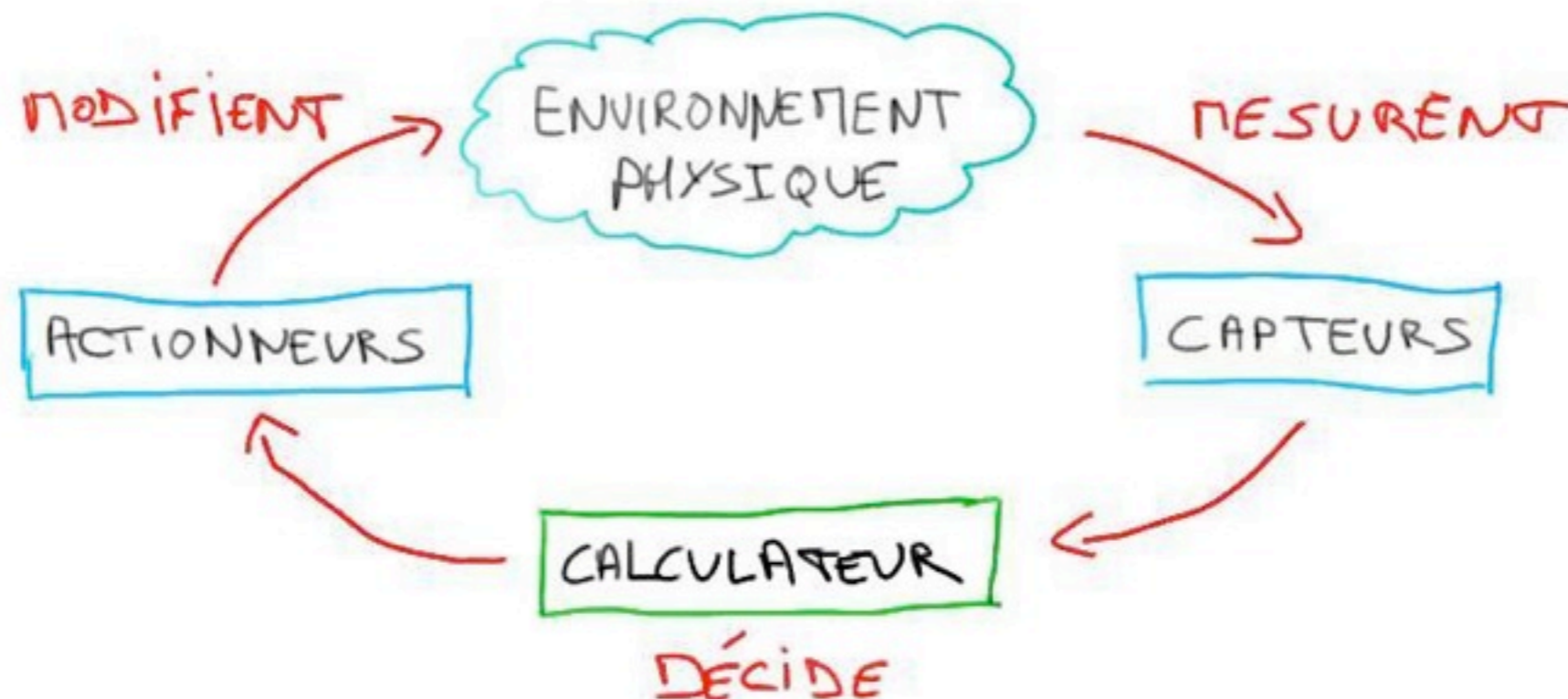
Spike





# Principe de fonctionnement

- **Capteurs** : mesurent des grandeurs physiques et les transmettent au calculateur
  - **exemples** : distances, forces, niveaux sonores et lumineux, etc.
- **Actionneurs** : mécanismes capables de modifier des grandeurs physiques
  - **exemples** : moteurs, chauffage, etc.
- **Calculateur** : pilote le système autonome en modifiant l'environnement grâce aux **actionneurs** en fonction des valeurs fournies par les **capteurs**

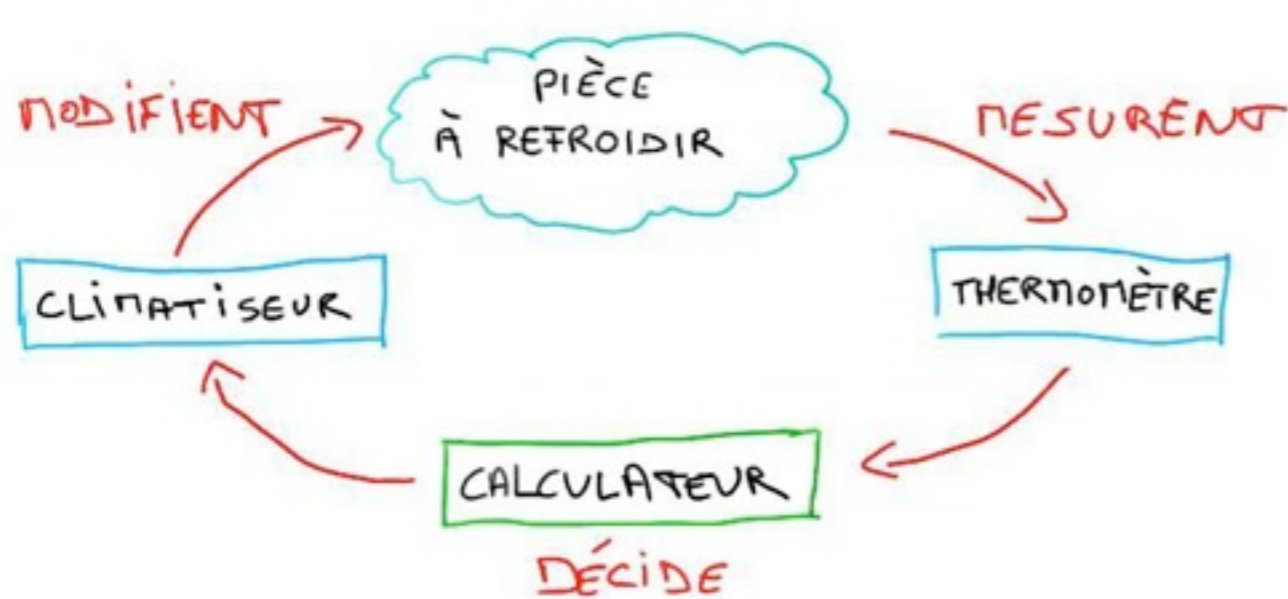




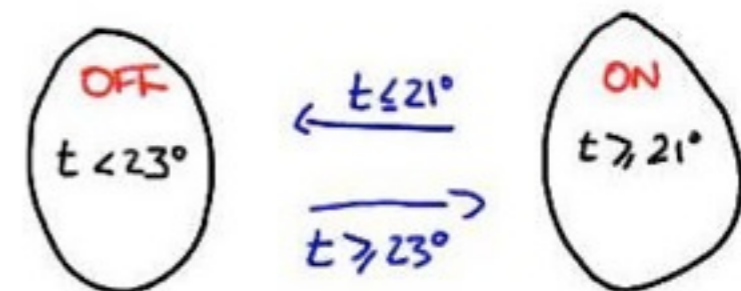
# Exemple

## Système de climatisation :

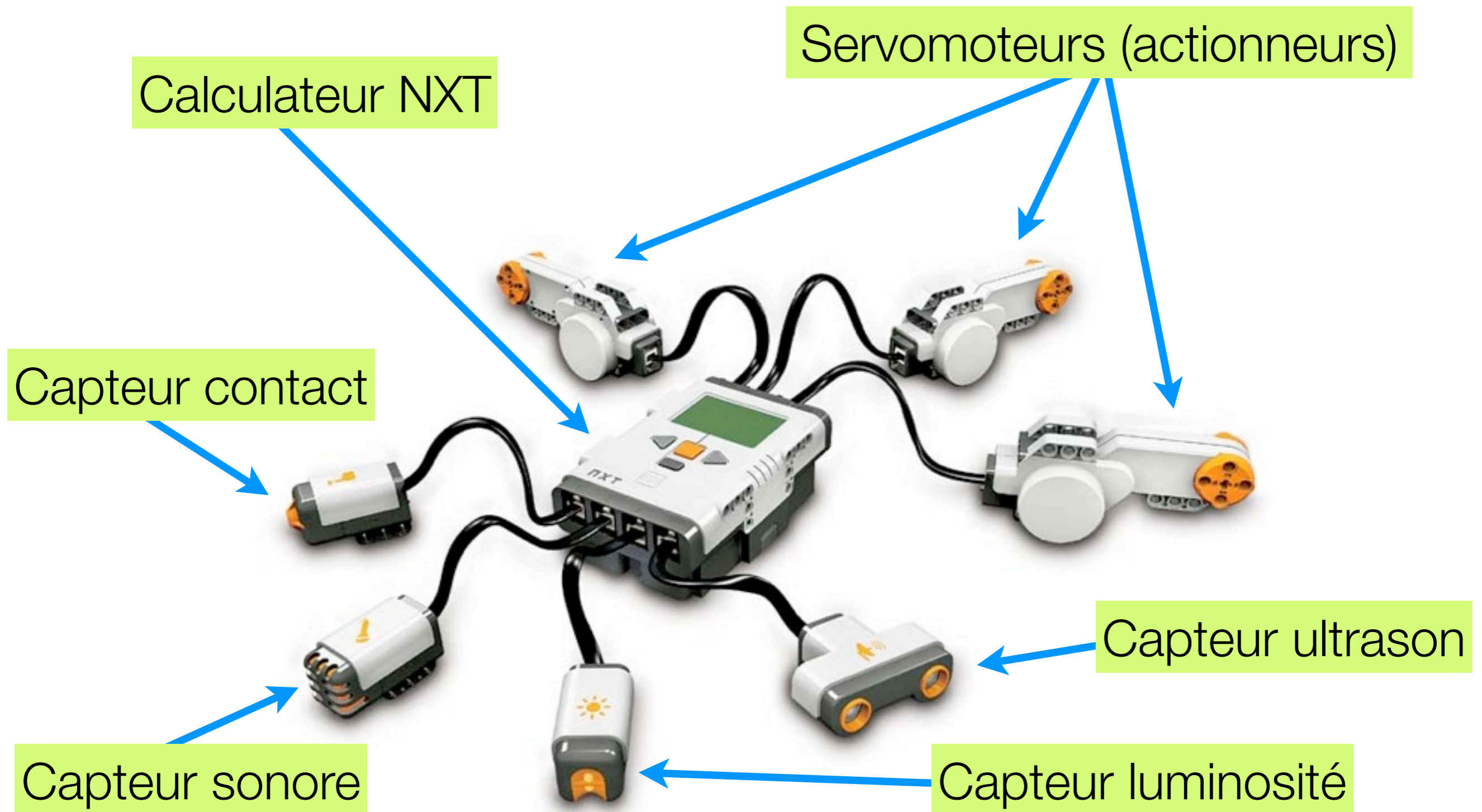
- **capteur** : thermomètre, mesure la température
- **actionneur** : système de refroidissement
- **ordinateur** (thermostat) : si trop chaud (capteur=thermomètre) alors refroidir la pièce (actionneur=système de refroidissement)



COMPORTEMENT DU CALCULATEUR :



# Exemple : le kit robot LEGO Mindstorms NXT



# Les capteurs



## Contact



Détecte lorsque l'on le presse et lorsque l'on le relache

## Son



Détecte le niveau sonore (dB)

## Luminosité



Détecte l'intensité lumineuse

## Ultrason



Détecte la distance à un objet (jusqu'à 2,5 m)



# Les servomoteurs

---

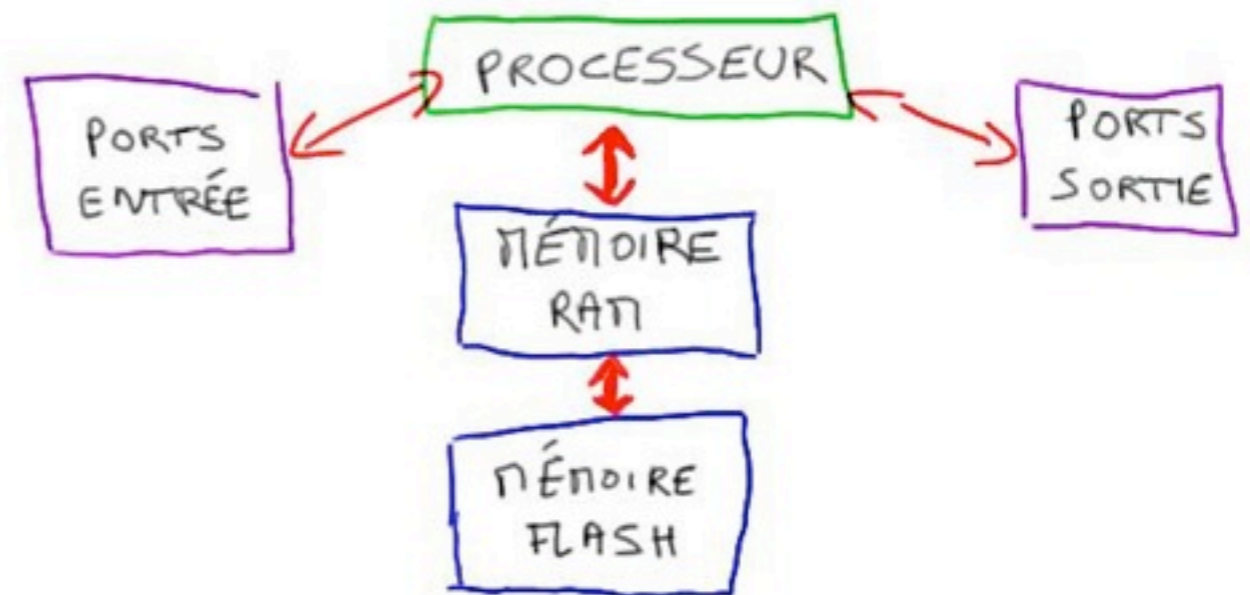
- Moteurs munis de capteurs de rotation
- Permet, par exemple, la synchronisation de 2 moteurs reliés à 2 roues différentes pour avancer droit
- permet de mesurer des angles de rotation



# Le calculateur NXT



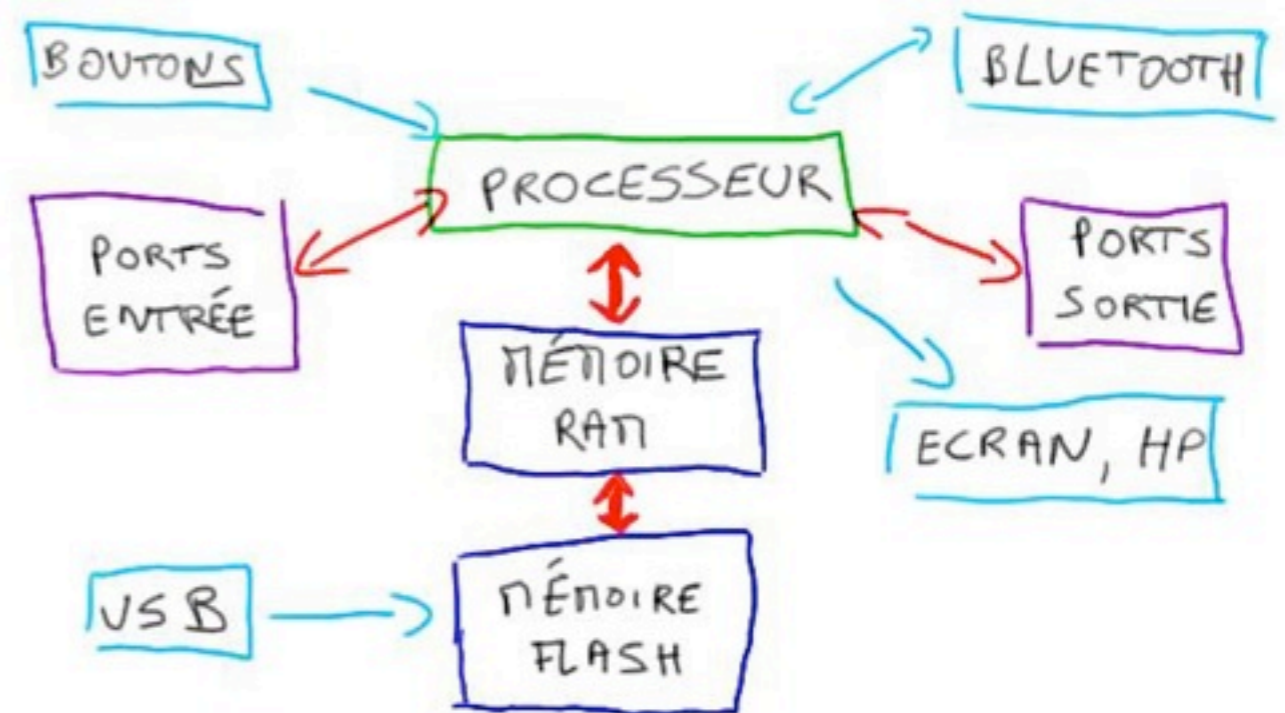
- Processeur 32 bits ARM7
- Mémoire flash 256 KO (persistant)
- Mémoire RAM 64 KO (non-persistant)
- 4 ports d'entrée (capteurs)
- 3 ports de sortie (servomoteurs)
- Communications Bluetooth
- Port USB
- Ecran LCD 100x64 pixels
- Haut-parleur
- Boutons (ON, Flèches, Clear)



# Le calculateur NXT



- Processeur 32 bits ARM7
- Mémoire flash 256 KO (persistant)
- Mémoire RAM 64 KO (non-persistant)
- 4 ports d'entrée (capteurs)
- 3 ports de sortie (servomoteurs)
- Communications Bluetooth
- Port USB
- Ecran LCD 100x64 pixels
- Haut-parleur
- Boutons (ON, Flèches, Clear)



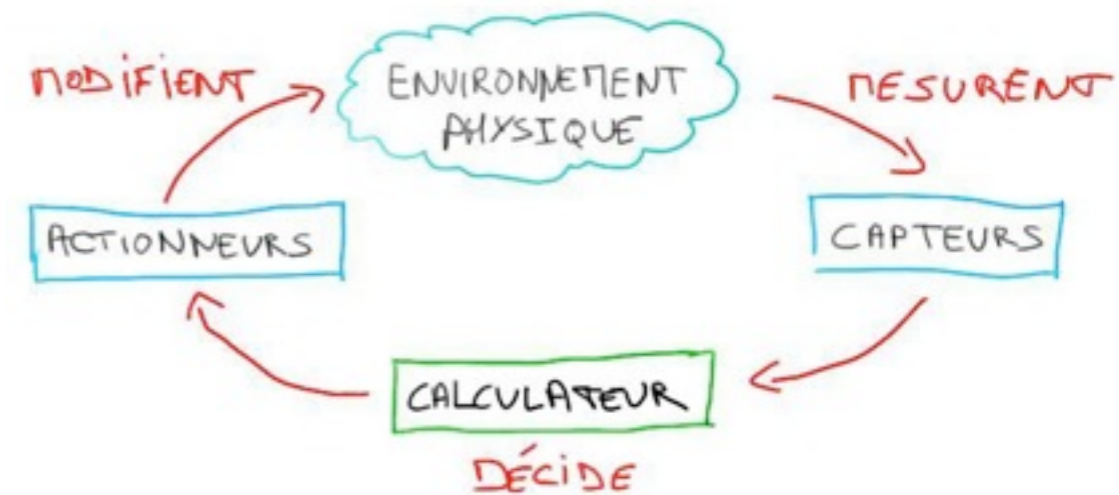
# Applications



- détection et capture d'objets
- atteindre une source lumineuse en évitant les obstacles
- suivre un chemin tracé au sol par exemple
- reconnaître un environnement (carte des lieux)
- etc...

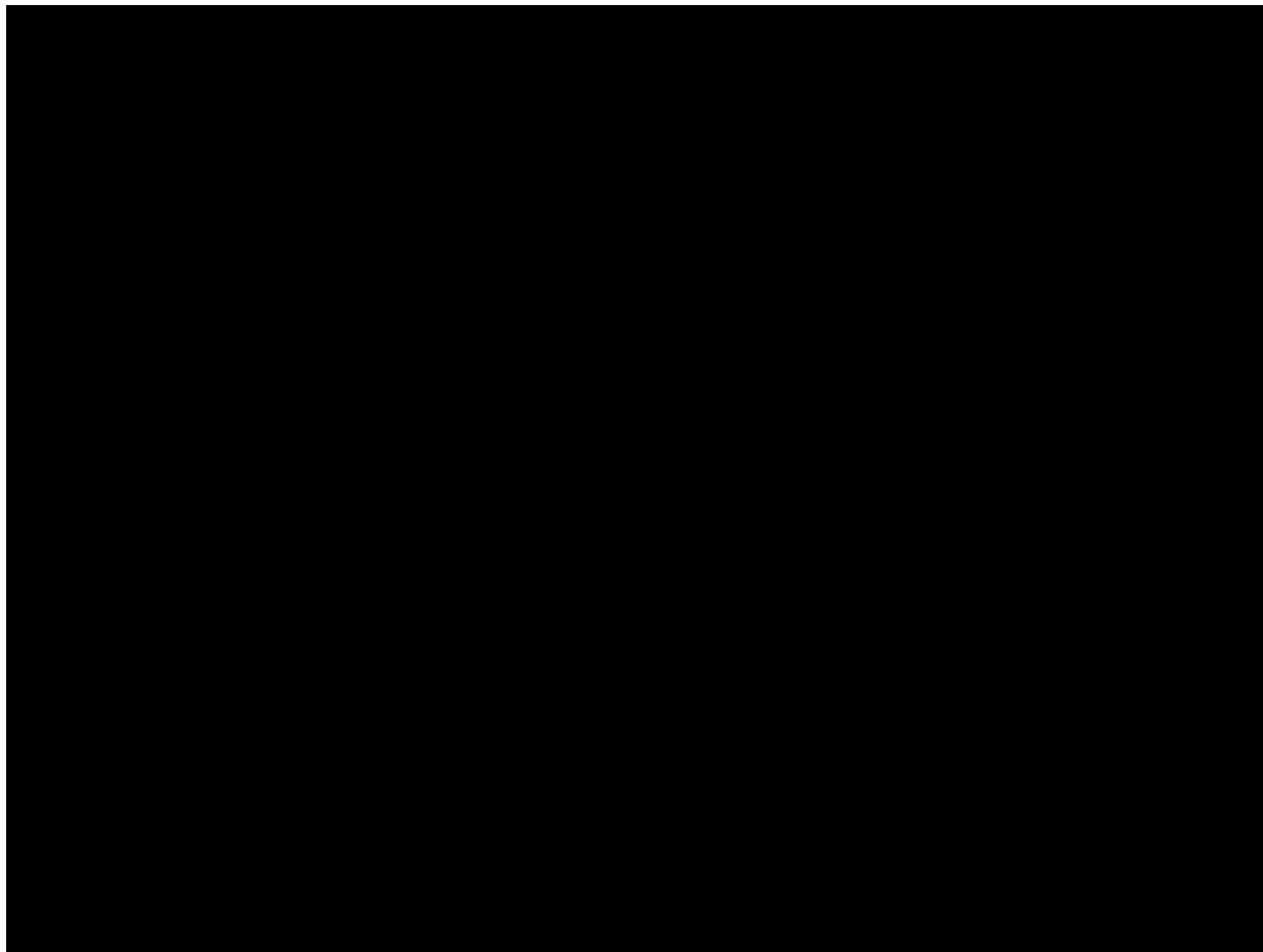


# Applications (suite)



# Applications (suite)

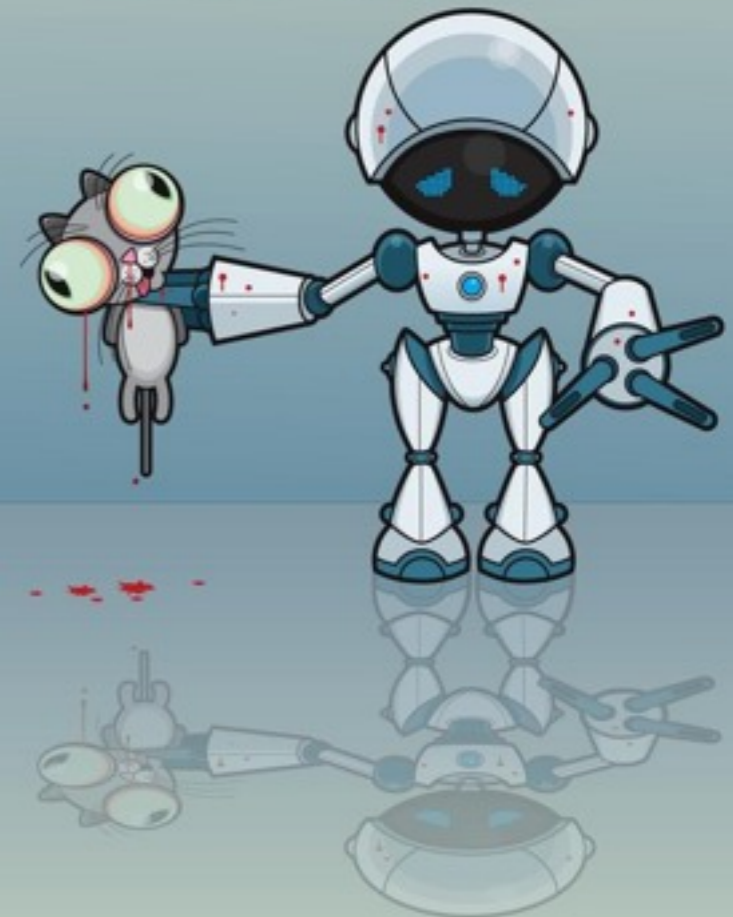
---



## 2. Programmation élémentaire

---

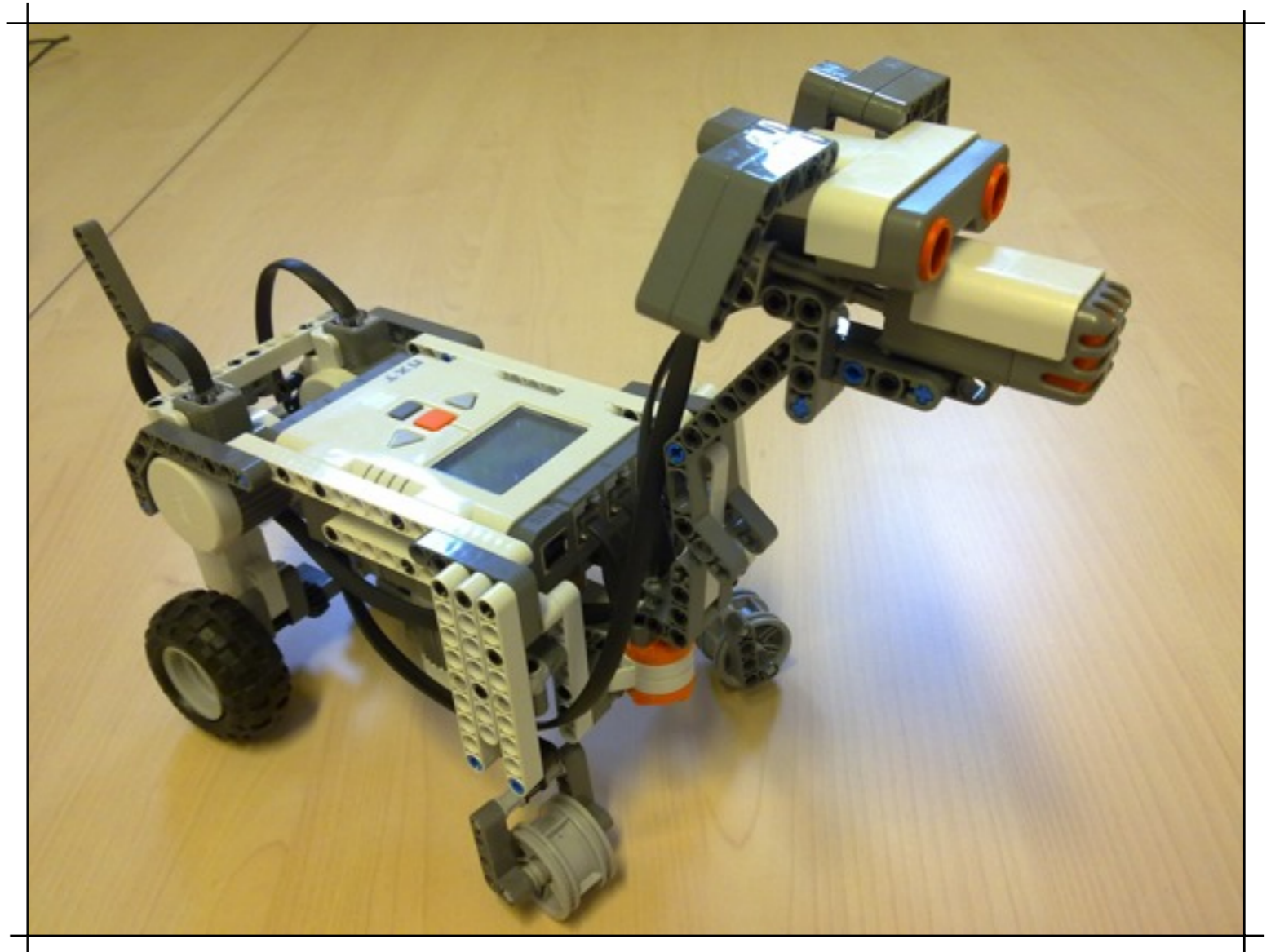
**“BAD ROBOT”**



# Objectif : programmer un robot simple

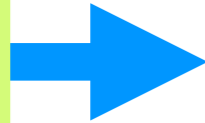
---

- Avancer, avancer en tournant la tête
- Réagir aux stimuli
- Jouer
- ...

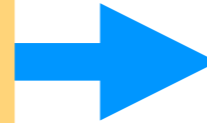


# Programmation

**programme**



**compilation**



**programme  
exécutable**



**téléchargement (usb,bluetooth)**

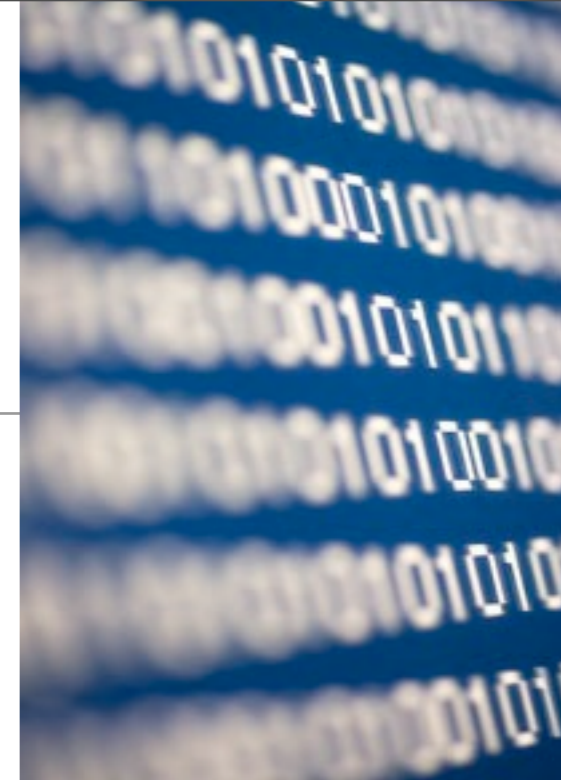


- Programme développé sur ordinateur
- Compilation : transformation du programme en programme directement compréhensible par le robot (programme exécutable en langage machine)
- Langage machine : instructions binaires compréhensible par le processeur
- Téléchargement du programme compilé dans le calculateur via le port USB ou par communication Bluetooth

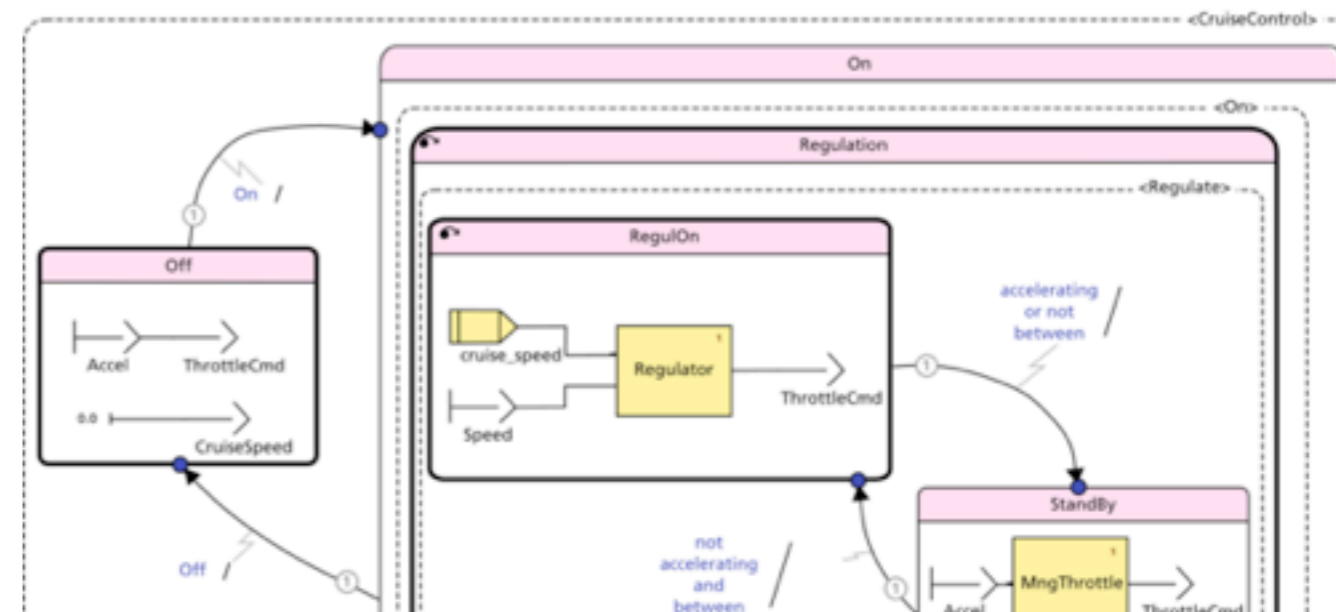


# Langages de programmation

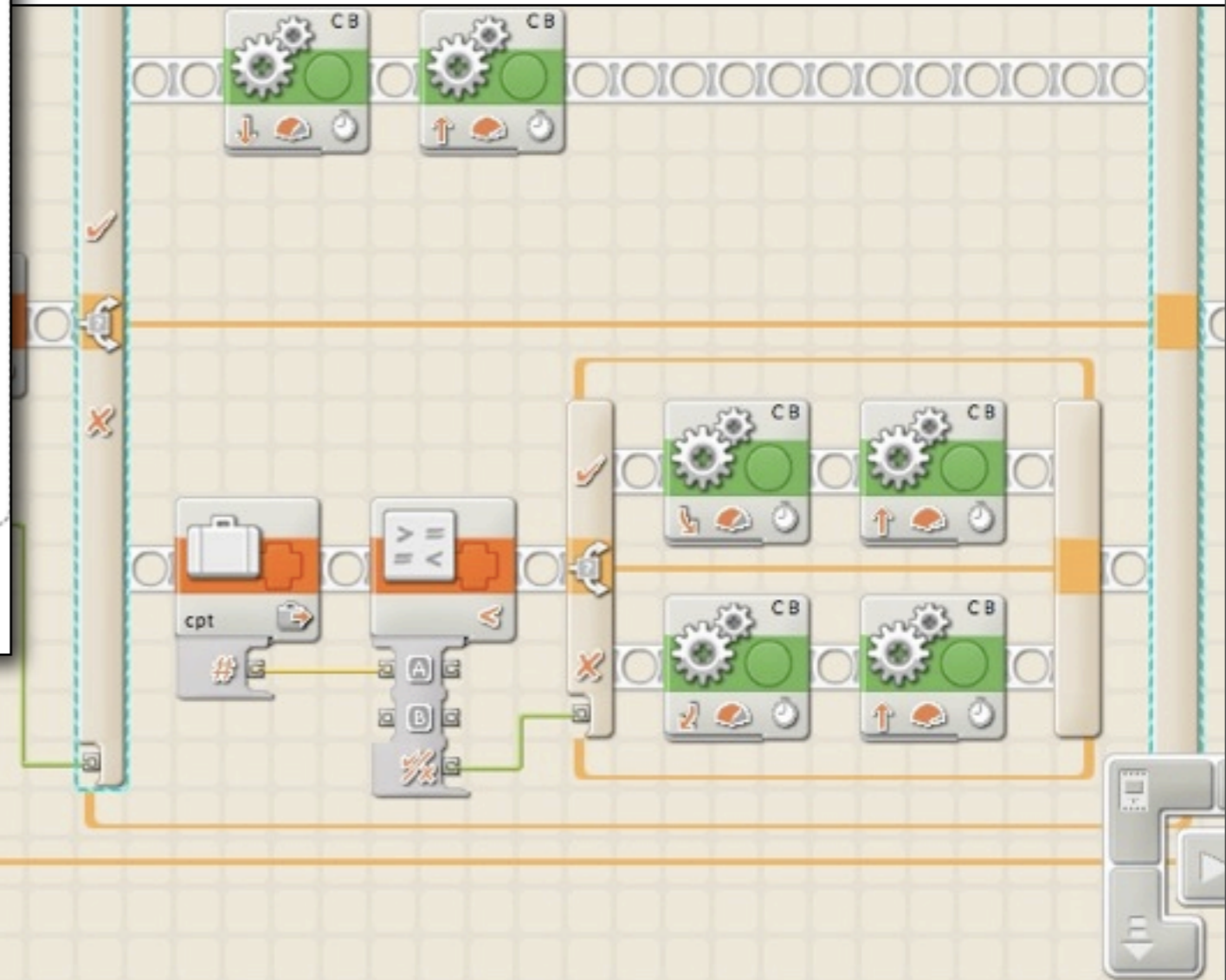
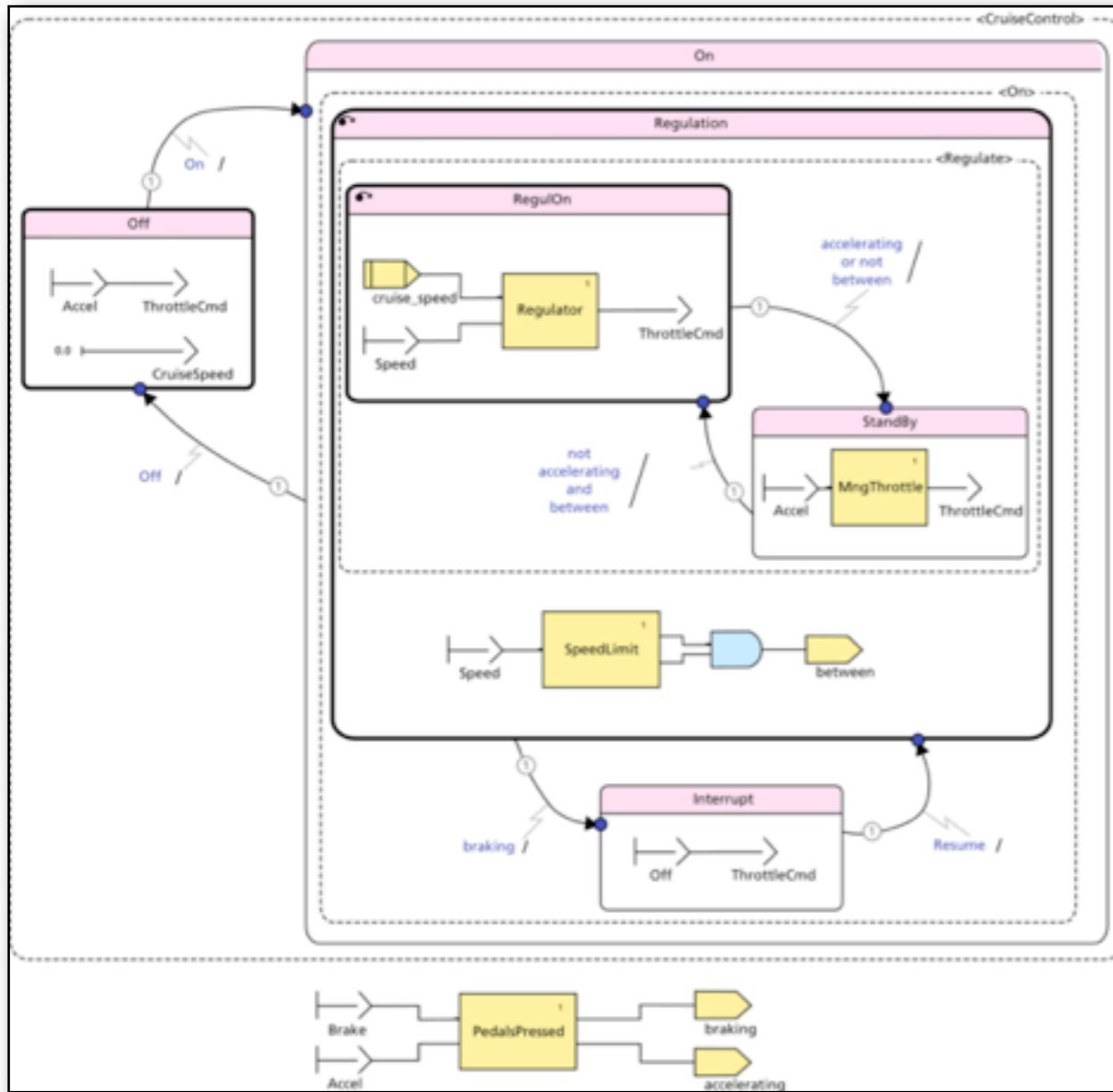
- Assembleur : (presque) directement compréhensible par le calculateur
- C : plus structuré que l'assembleur, simplifie la programmation
- Java : encore plus abstrait que C (objets)
- Langage graphique : plus simple d'utilisation, moins expressif



```
pc@05462EFBB3AE435 ~
$
Your group is currently "nkpaswd". This indicates that
the /etc/passwd (and possibly /etc/group) files should be rebuilt.
See the man pages for nkpaswd and nkgroup then, for example, run
nkpaswd -l [-d] > /etc/passwd
nkgroup -l [-d] > /etc/group
Note that the -d switch is necessary for domain users.
pc@05462EFBB3AE435 ~
```



# Programmation graphique



# Programme élémentaire

The screenshot shows the LEGO MINDSTORMS NXT software interface. The main workspace is a grid with a program flow starting from a 'Début du programme' (Start of program) block, indicated by a red arrow and the handwritten text 'Début du programme'. This block leads to a motor control block, also indicated by a red arrow and the handwritten text 'Contrôle moteurs'. The motor control block is configured with port B, a duration of 1 rotation, and the 'Freiner' (Brake) action. The bottom panel shows the configuration for the selected motor control block, including port selection (A, B, C), direction (up, down, stop), and action (Freiner, Continuer). The top bar shows the user profile as 'Par défaut'.

# Commande des moteurs



choix moteurs

Puissance



rotation  
robot

← temps  
de marche  
moteur

# 1<sup>ère</sup> étape : avancer en tournant la tête

The screenshot shows the LEGO MINDSTORMS NXT software interface. The main workspace displays a gear-based mechanism with four motor blocks. The first block is labeled 'BC', the second 'A', and the last two 'A'. A red arrow points to the 'BC' block, another to the first 'A' block, and a third to the second 'A' block. An orange box highlights the second and third 'A' blocks, with a red arrow pointing to it from the word 'Boucle' written in red. The interface includes a toolbar on the left with various icons, a top menu bar with 'Profil d'utilisateur : Par défaut', and a bottom status bar with a help message: 'Vous avez besoin d'aide ? Amenez le pointeur de la souris au-dessus d'un objet pour afficher une infobulle qui explique sa fonction. Pour plus d'informations, cliquez sur le lien « Aide supplémentaire ».

Profil d'utilisateur : Par défaut

Commun

essai

Port :  A  B  C

Alimentation  40

Direction :  ↑  ↓  ↻

Durée : 30 Degrés

Diriger : A

Action suivante :  Freiner  Continuer

Port :  A  B  C

Alimentation  40

Direction :  ↑  ↓  ↻

Durée : 15 Degrés

Diriger : A

Action suivante :  Freiner  Continuer

Port :  A  B  C

Alimentation  75

Direction :  ↑  ↓  ↻

Durée : 1800 Éliminé

Diriger : B C

Action suivante :  Freiner  Continuer

**Vous avez besoin d'aide ?**  
Amenez le pointeur de la souris au-dessus d'un objet pour afficher une infobulle qui explique sa fonction. Pour plus d'informations, cliquez sur le lien « Aide supplémentaire ».

[Aide supplémentaire »](#)

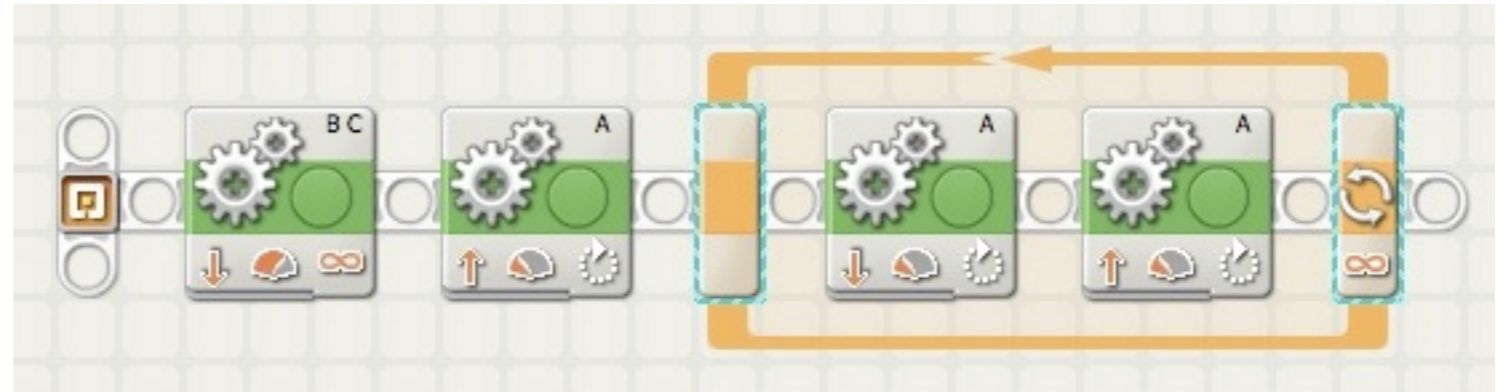


# Boucles et séquences

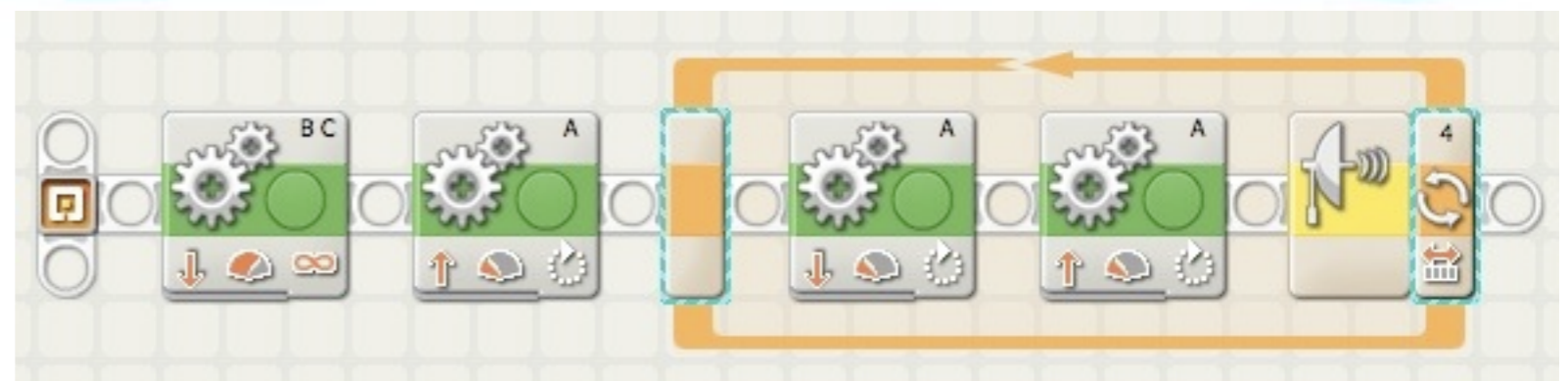
- Séquence d'instructions : instructions exécutées les unes après les autres



- Boucle : permet de répéter plusieurs fois une séquence d'instructions



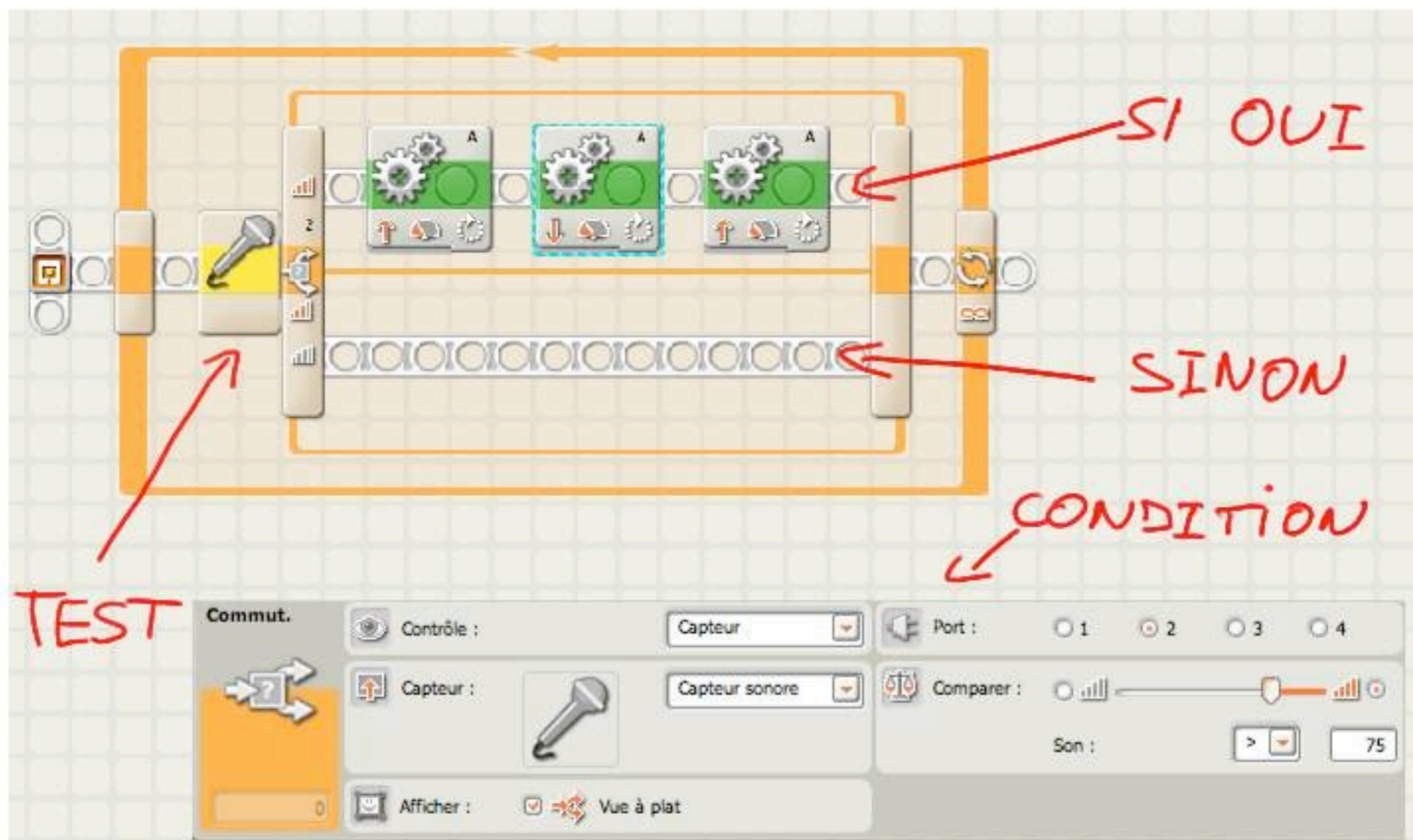
- Répétition : nombre de fois déterminé ou jusqu'à ce qu'une condition soit remplie



# Instructions conditionnelles

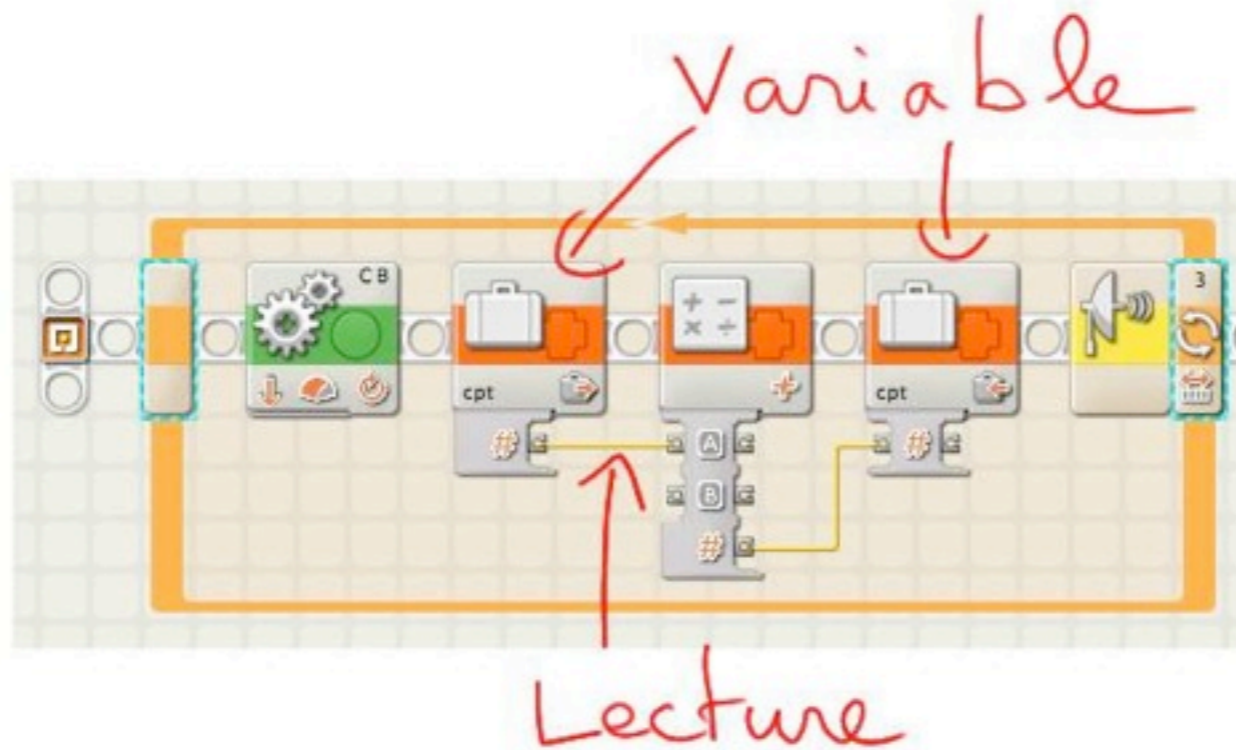


- Instructions conditionnelles : permettent d'exécuter une séquence d'instructions ou une autre en fonction d'une certaine condition
- Exemple : bouger la tête s'il y a du bruit

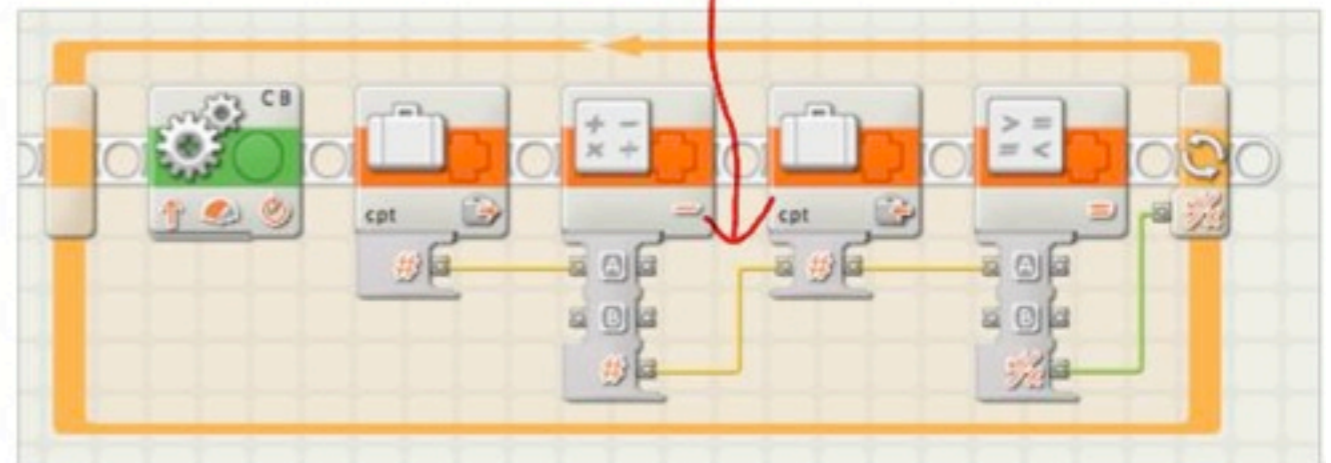


# Variables

- Variables : permettent de mémoriser des valeurs
- Les variables peuvent être lues ou écrites



écriture



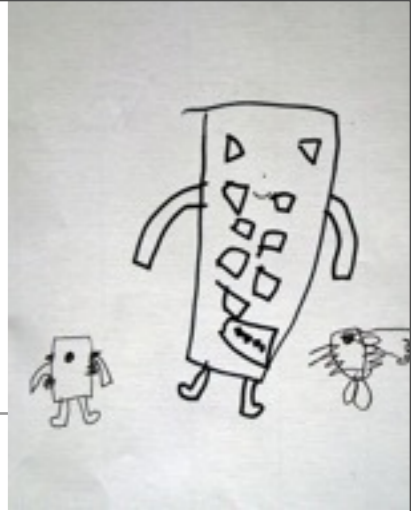
ex : avancer jusqu'à rencontrer un obstacle puis revenir au point de départ

### 3. Application 1 : jeu de balle

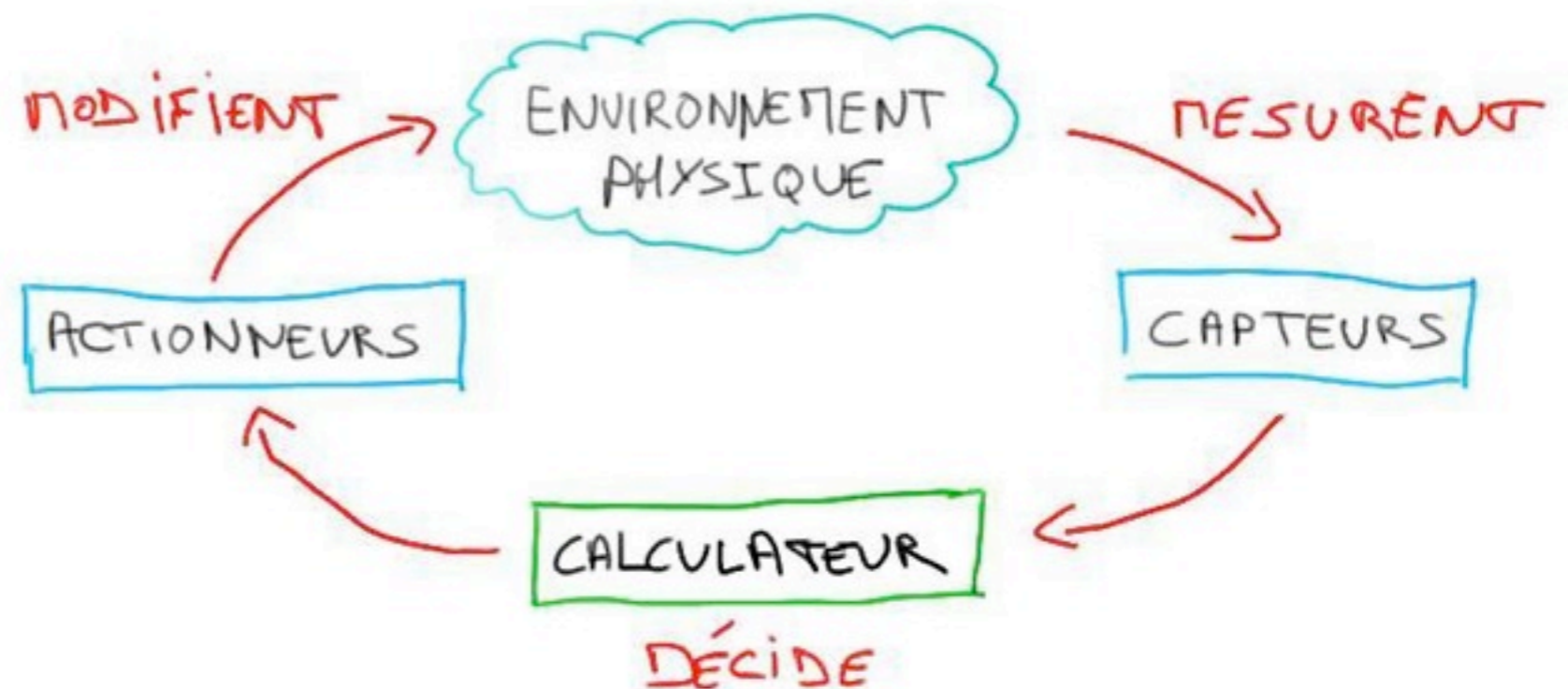
---



# Description informelle

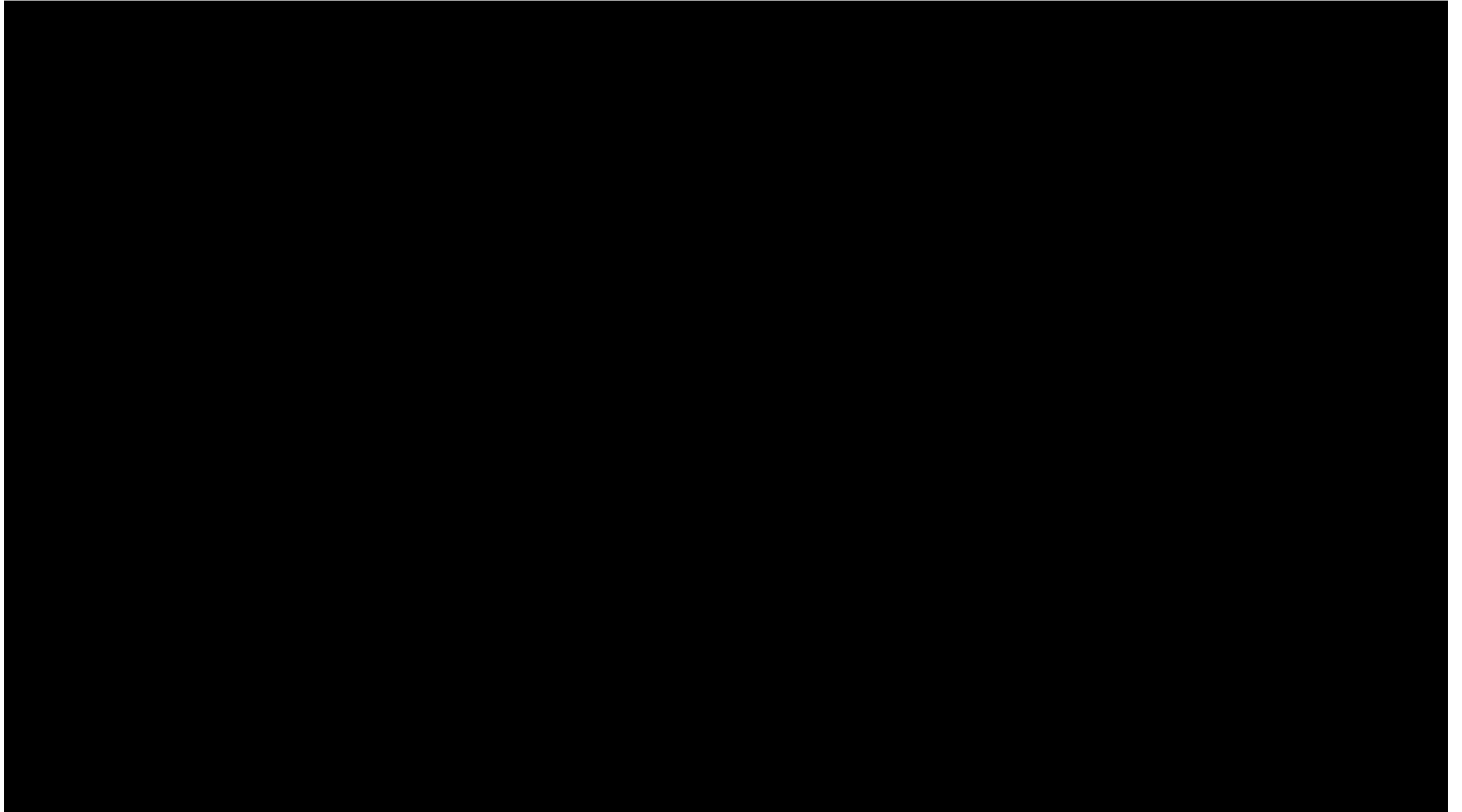


- Regarder à gauche et à droite
- Lorsqu'un objet apparaît :
  - avancer vers lui
  - revenir à son point de départ
- recommencer

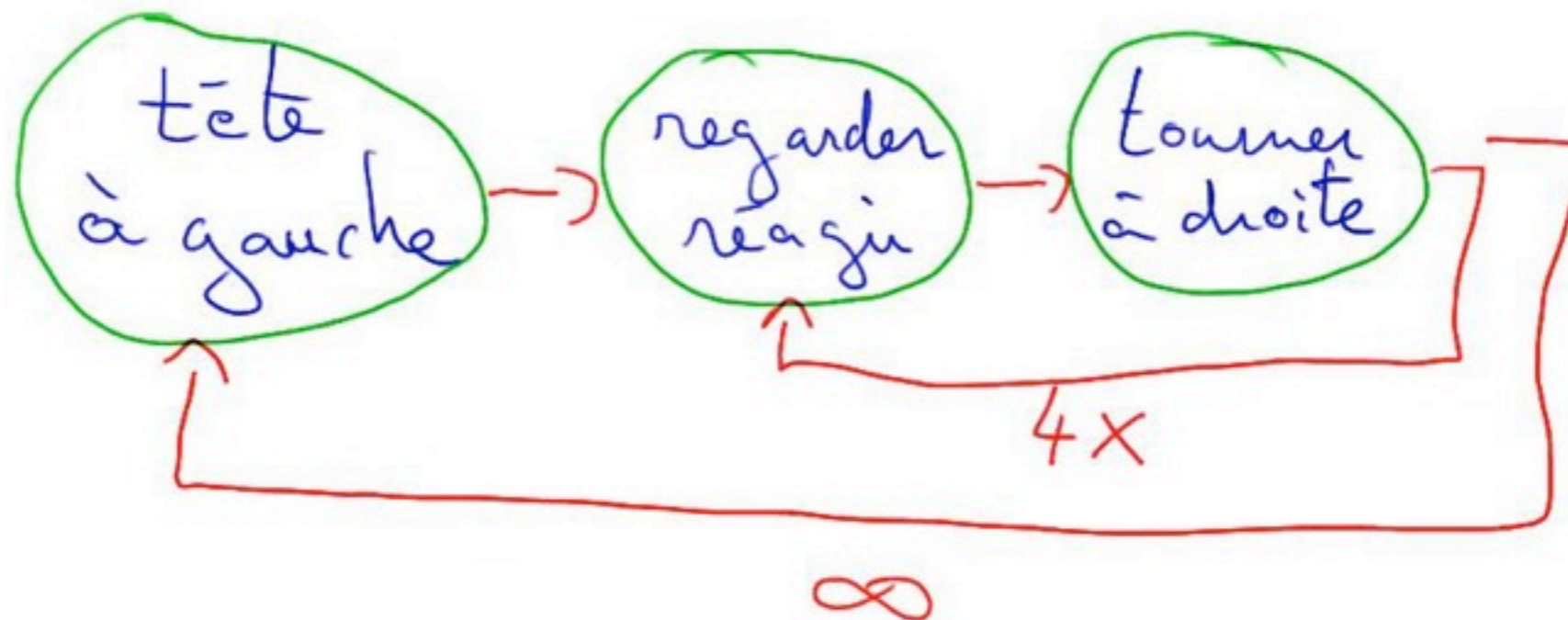
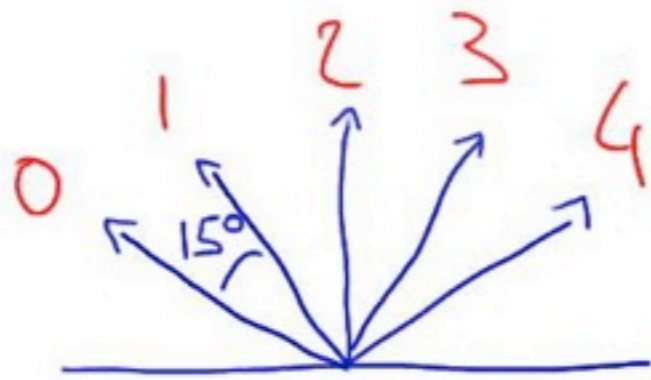
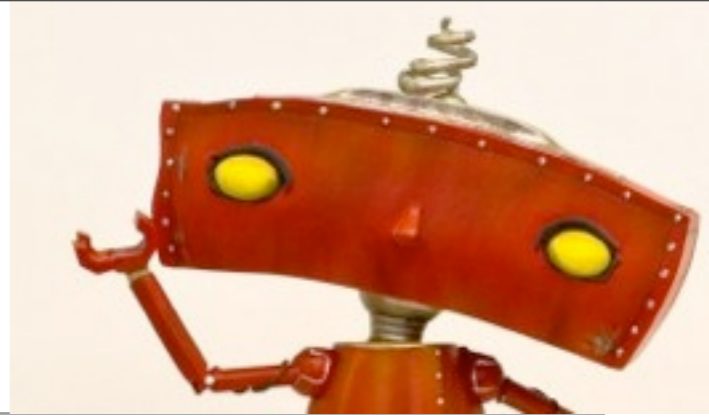


# Aperçu

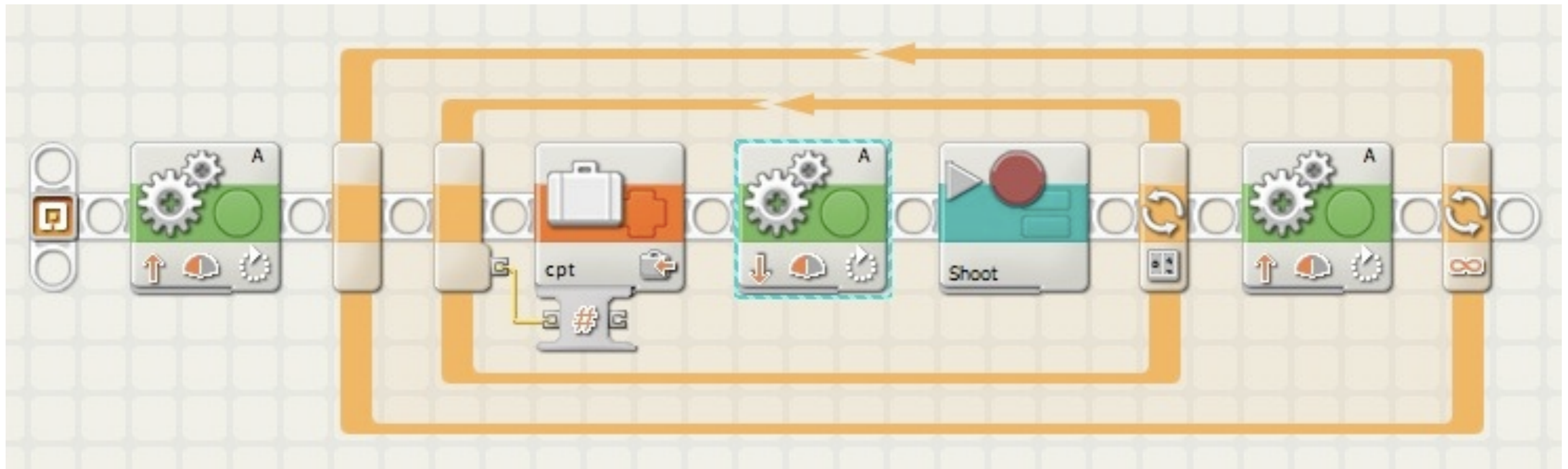
---



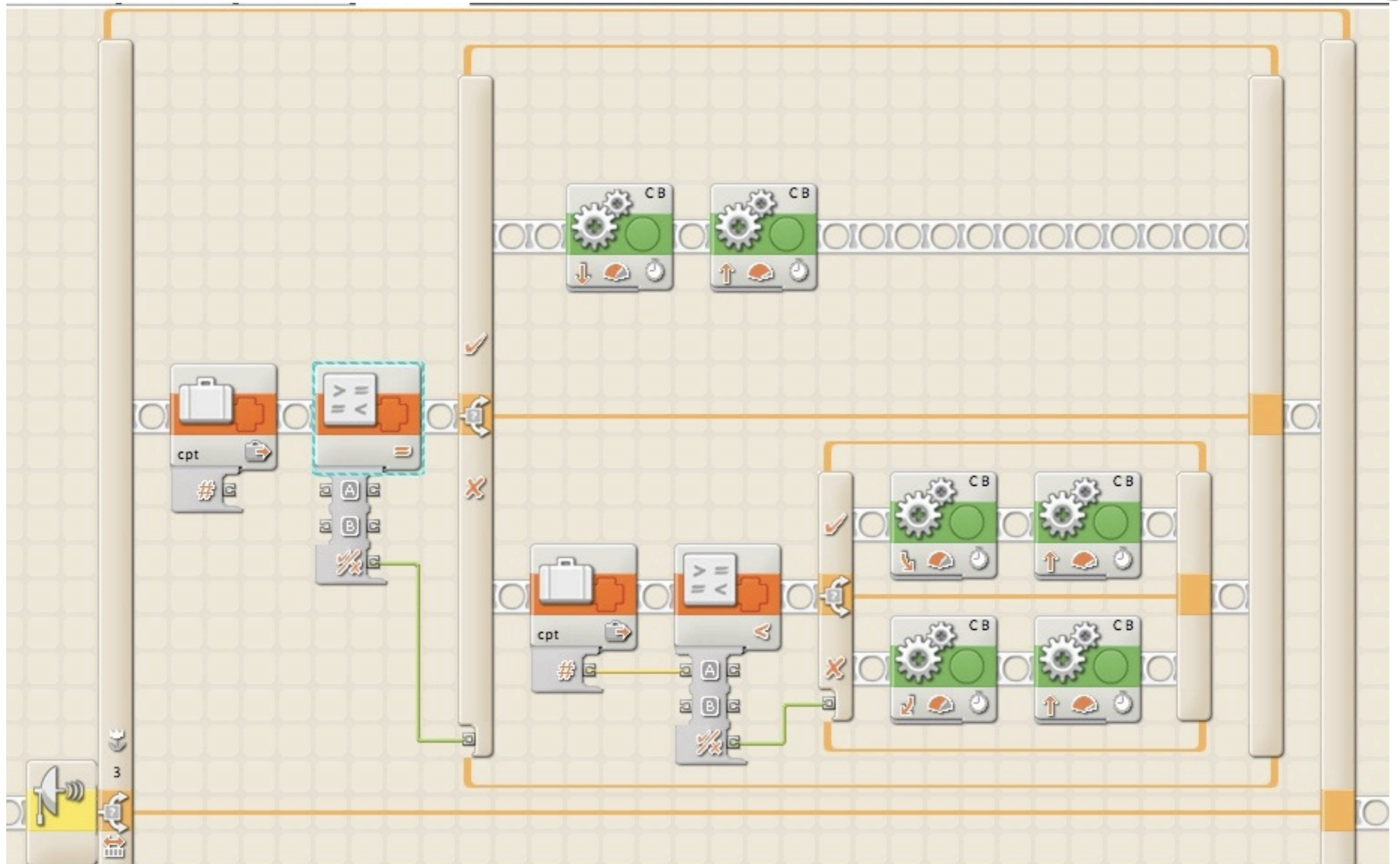
# Principe de fonctionnement



# Vue d'ensemble



# Shoot



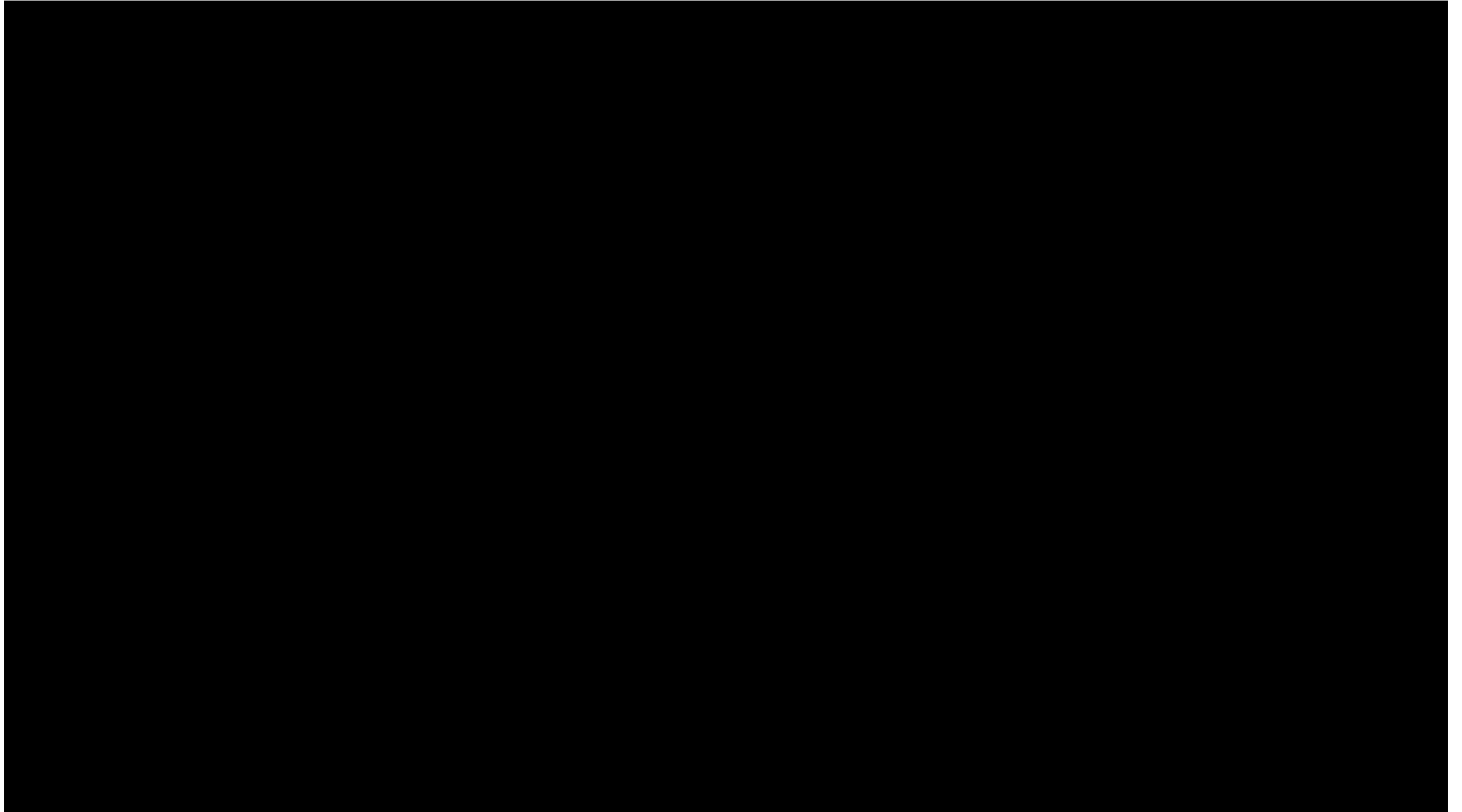
### 3. Application 2 : garder ses distances

---

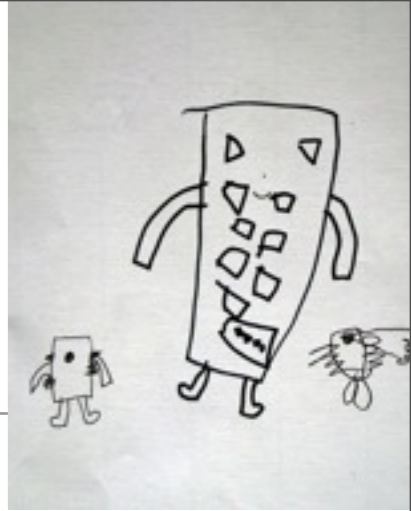


# Aperçu

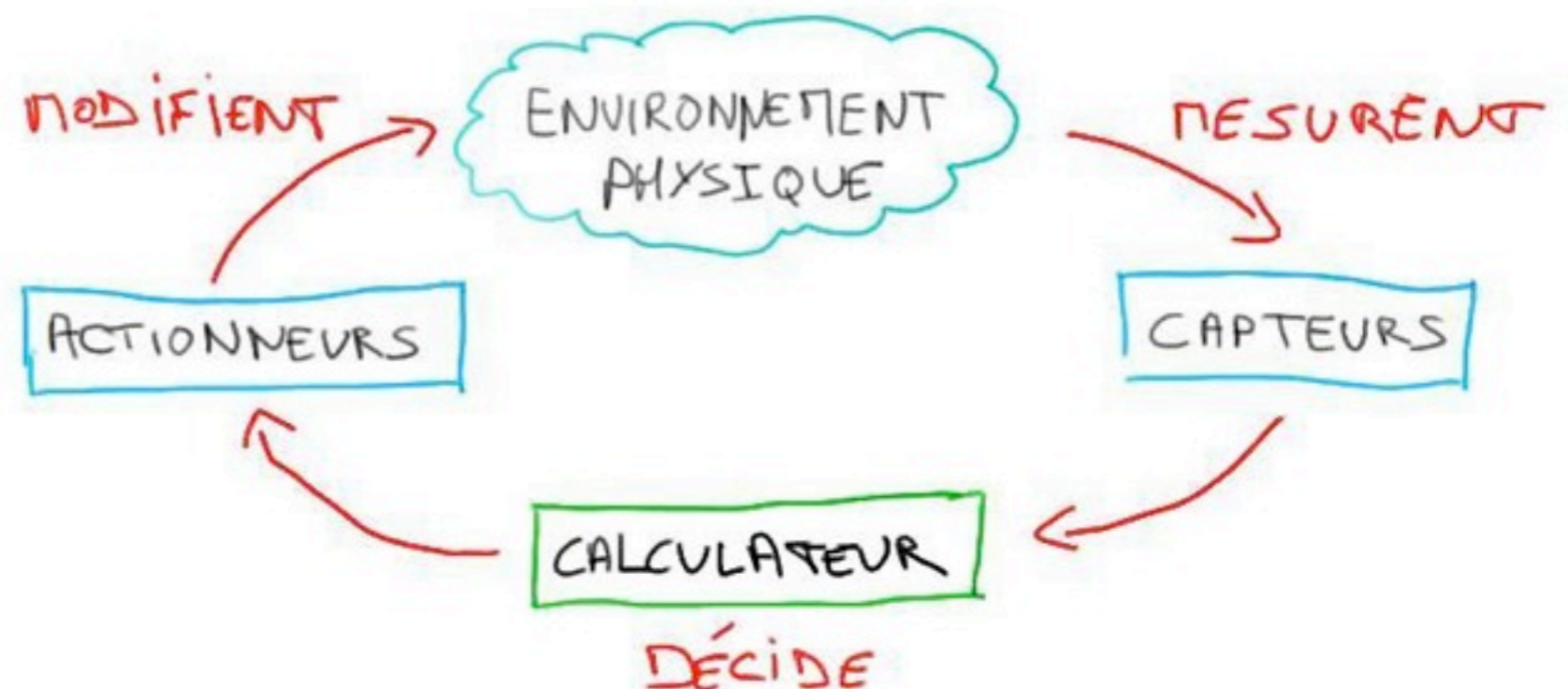
---



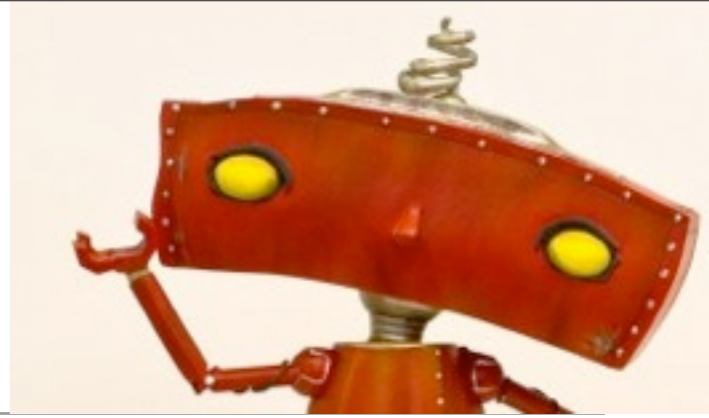
# Description informelle



- Avancer en regardant devant soi (en tournant légèrement la tête)
- Lorsqu'un objet apparaît à moins de x centimètres
  - reculer pour le ramener à distance de x centimètres
  - réagir + ou - fortement selon la distance à l'objet



# Principe de fonctionnement



$d$  : distance à obstacle

$D$  : distance idéale

$e = |D - d|$  erreur ( $e = 0$  si OK)

---

réponse  $y$  : mouvement à effectuer

$$y(t) = y_p(t) + y_i(t)$$

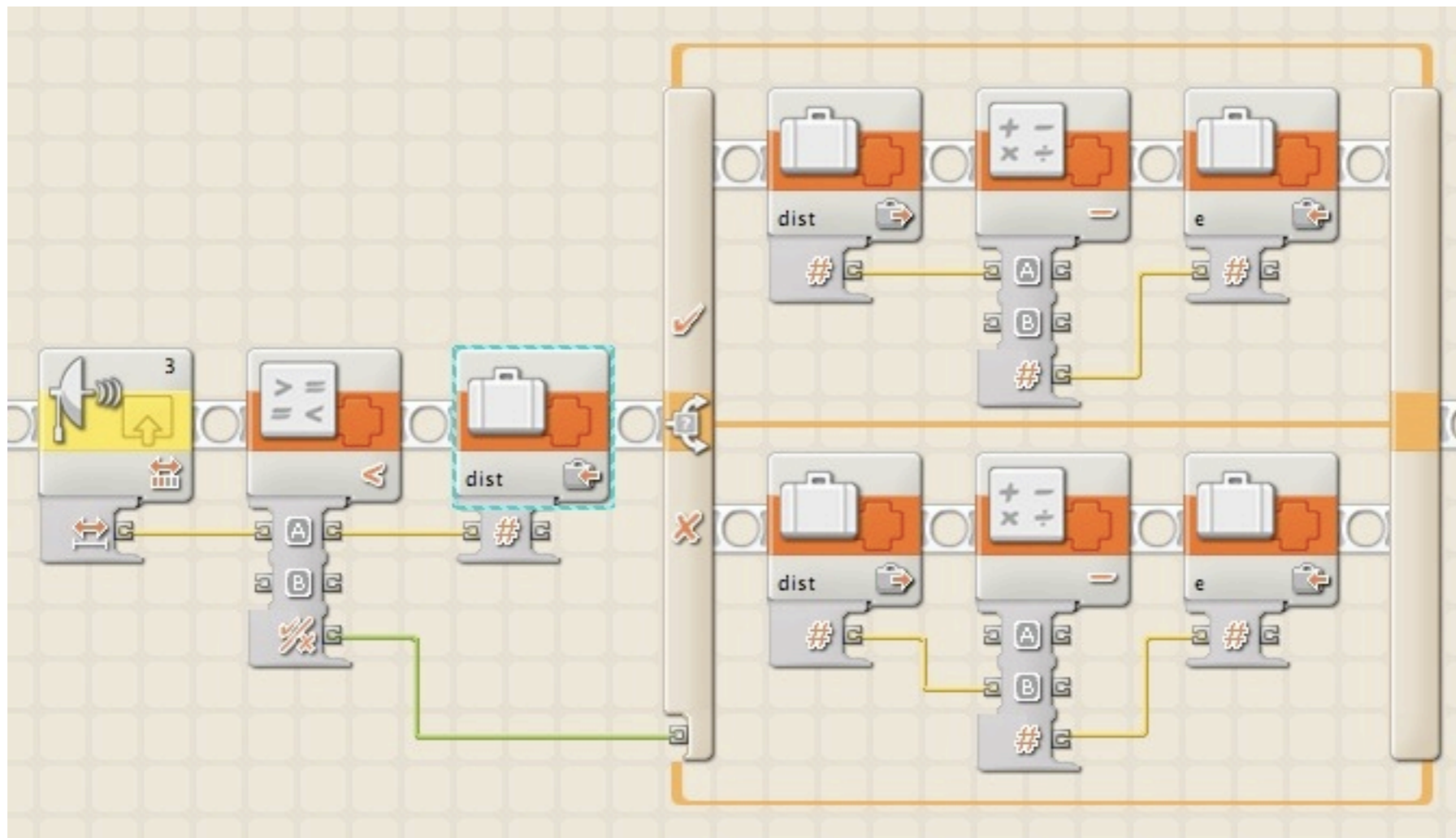
$$y_p(t) = k_p e(t)$$

$$k_p = 0,2$$

$$y_i(t) = y_i(t-1) + k_i e(t)$$

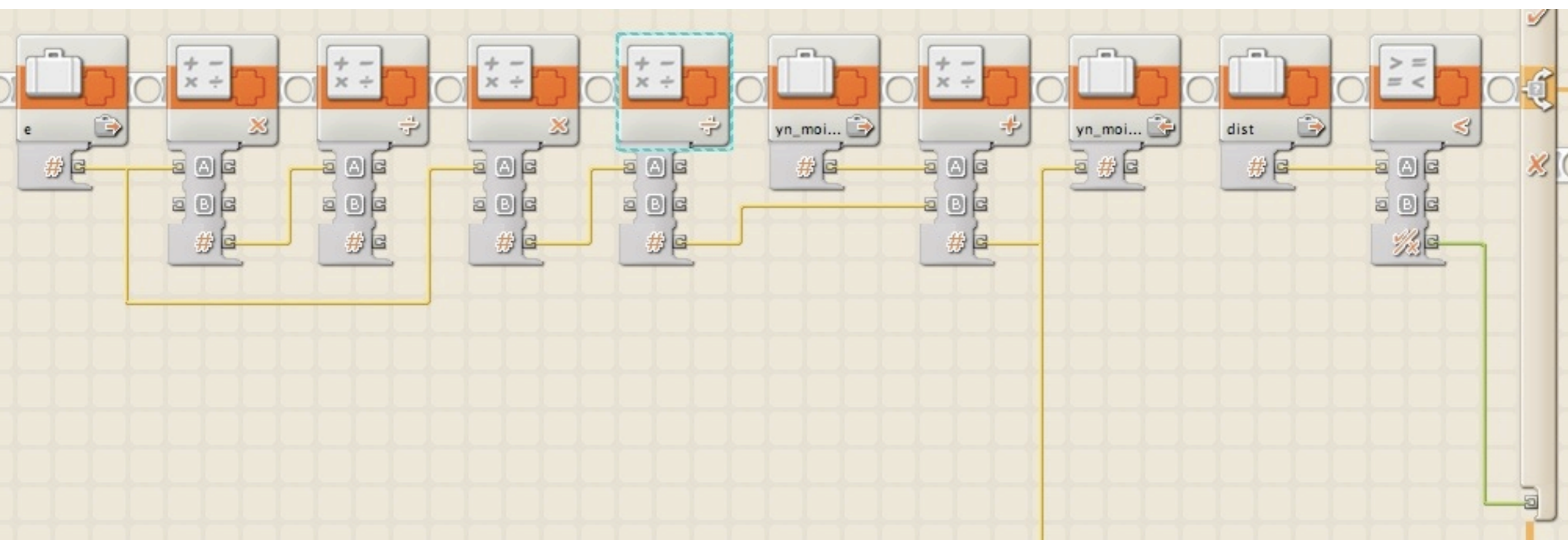
$$k_i = 0,3$$

# Acquisition des données

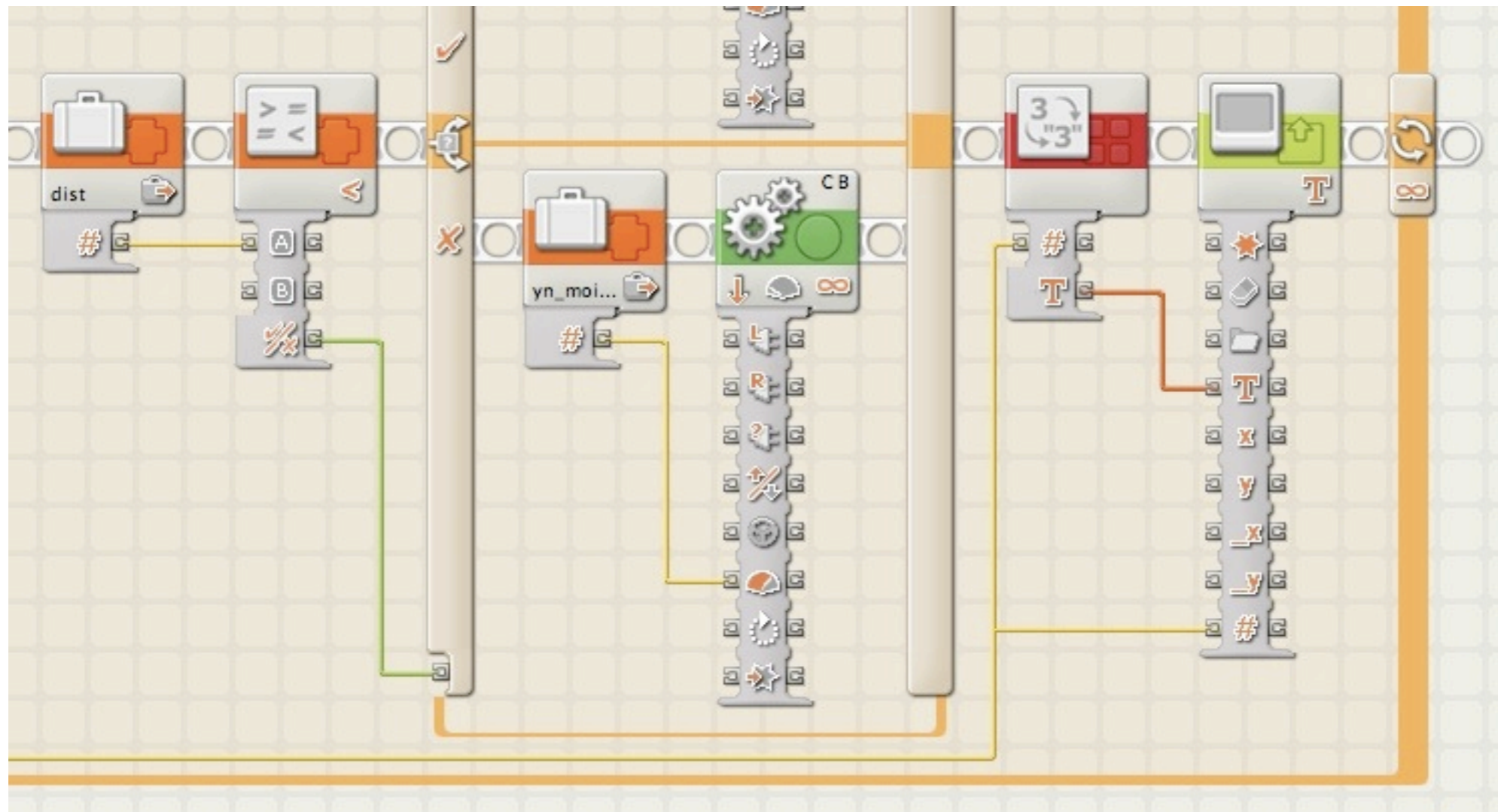




# Calcul de la réponse



# Activation des moteurs



# 4. Conclusion

---



# Exemple de programme Java

```
public class Cl8 {
    static final int SEUIL_LUM = 430;
    static final int SEUIL_DIST = 35;
    static final float WHEEL_DIAM = 5.6F;
    static final float TRACK_W = 13F;

    static UltrasonicSensor us = new UltrasonicSensor(SensorPort.S1);
    static LightSensor ls = new LightSensor(SensorPort.S4, true);
    static TouchSensor ts = new TouchSensor(SensorPort.S2);
    static Pilot cl8 = new Pilot(WHEEL_DIAM, TRACK_W, Motor.C, Motor.B, true);

    public static void main(String[] args) throws Exception {
        init(); // initialise le robot
        while(Button.ESCAPE.isPressed()==false) {
            guidageLumiere(); // execute guidageLumiere()
        }
    }

    public static void guidageLumiere() throws Exception {
        int lum;
        int k=0;
        while(k<2) {
            affiche();
            suivreRail();
            lum =ls.readNormalizedValue(); // memoriser l'intensite lumineuse
            if (lum<SEUIL_LUM) {
```



# Conclusion



Programmation  
en Ada, C ...  
(L1,L2,L3)

Intégration &  
validation du logiciel  
(M1)

Projets de  
programmation (L3)

Licence  
info



Intelligence  
Artificielle (L2)

Qualité numérique  
(L3)



Masters pro ou recherche,  
écoles d'ingénieurs, etc.

