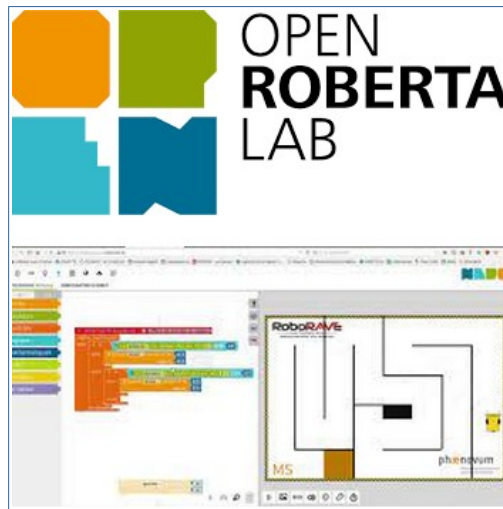


Présentation Open Roberta : programmation de robots en ligne

Index des pages

1 Présentation	1
2 Présentation de l'Interface	2
Principe.....	2
Interface.....	2
11 Défis pour progresser	3
Activité 1 : Faire avancer le robot jusqu'au mur (séquences).....	3
Activité 2 : Afficher sur l'écran du robot NTX la valeur du capteur dans l'afficheur du robot NXT (boucles conditionnelles et variables).....	3
Activité 3 : Suivi de ligne et détection de ligne.....	3
Activité 4 : Simulation d'un robot aspirateur.....	3



1 Présentation

OpenRoberta est un environnement de programmation de robots en ligne avec une interface de simulation du programme.

Ce projet, d'initiative allemande, "Roberta - Learning with robots" a été développé par Fraunhofer IAIS. Diffusé en open source, Il cherche à encourager les enfants à coder. Gratuit, il ne nécessite aucune installation.

Plusieurs robots et cartes programmables sont accessibles à la programmation (robots LegoMinstorms,, mBot, micro:bit, ...), qu'il faut définir au lancement de l'application web.

Des capteurs et actionneurs configurables sont mis à disposition pour chaque robot avec des blocs de programmation type scratch pour les programmer. Une fois le robot programmé, une interface de simulation permet de visualiser le résultat.

[Vidéo de présentation :](#)



Principe

1. Lancer OpenRoberta : <https://lab.open-roberta.org/>
2. Choisir un robot (NXT par exemple)
3. Ouvrir l'interface de simulation SIM
4. Si besoin, Configurer capteurs et actionneurs : CONFIGURATION DU ROBOT NXTbase
5. Choisir et configurer la scène : PROGRAMME NEPOprog puis SIM puis Changer la scène
6. Programmer le robot avec les blocs de programmation à disposition
7. Si besoin, Afficher la façade du robot pour accéder à l'afficheur et aux boutons
8. Lancer la simulation
9. Si besoin, Débugger
10. Enregistrer son programme : Exporter le programme

Interface

The screenshot shows the OpenRoberta web interface. On the left is a vertical menu with categories: action (Mouvement, Rouler, Affichage, Fils, Lumière), Capteurs, Contrôle (Décision, Boucles, Attendre), logique, Mathématiques, Texte, Listes, Couleurs, variables, Fonctions, messages. The main workspace is divided into several sections:

- Configuration du robot:** Includes a 'Configurer capteurs actionneurs' button and a 'Facéade robot NXT' window showing a robot's front panel with a screen and buttons. Labels include 'ENREGISTRER, charger un programme', 'Blocs de PROGRAMMATION', and 'Espace pour glisser-déposer ses blocs'.
- Simulation:** A central area with a 'SIM' button and a 'Débugage' window. Labels include 'Choix de la scène', 'Ajouter une scène', 'Ajouter Obstacle', 'Changer couleur', and 'Ajouter Marque de couleur'.
- Scène:** A 3D simulation environment showing a robot on a track with colored markers (blue, yellow, green, red) and a black obstacle. A 'Scène' label points to the environment. A 'Marque de couleur' label points to a red marker, and an 'Obstacle' label points to the black barrier.
- Statistiques:** A panel on the right shows sensor data: 'Light Sensor 3 33%', 'Ultra Sensor 1 52cm', 'Ultra Sensor 4 52cm', 'Sound Sensor 0%', 'Color Sensor 3'. A label 'Affichage valeurs des capteur entre autres' points to this panel.
- Bas de page:** Includes 'Réinitialiser la position du robot', 'Afficher valeurs capteurs', 'Afficher Façade robot NXT', and 'Lancer la simulation' buttons.

11 Défis pour progresser

Dans les défis, nous vous invitons à cliquer sur les liens : ViDÉO pour visualiser le résultat attendu, BLOCS à PROGRAMMER pour lancer l'interface Lab OpenRoberta avec à disposition les blocs à agencer et à paramétrer.

Activité 1 : Faire avancer le robot jusqu'au mur (séquences)

- **Défi 1** : Quelle distance avant de toucher le mur ?
[ViDÉO1](#) – [BLOCS à PROGRAMMER1](#) - [CoRRECTION1](#)
- **Défi 2** : Passer par les 4 portes pour réussir le parcours
[ViDÉO2](#) – [BLOCS A PROGRAMMER2](#) – [CORRECTION2](#) – [Image du plateau](#)
- **Défi 3** : Tracer un carré de 50 cm de côté avec 3 blocs seulement
[ViDÉO3](#) – [BLOCS A PROGRAMMER3](#) – [CORRECTION3](#)

Activité 2 : Afficher sur l'écran du robot NTX la valeur du capteur dans l'afficheur du robot NXT (boucles conditionnelles et variables)

- Défi 4 : Le robot doit s'arrêter à 10 cm d'un obstacle
[ViDÉO4](#) – [BLOCS à PROGRAMMER4](#) - [CoRRECTION4](#)
- Défi 5 : Afficher sur l'écran la valeur d'une variable "distance", arrondie à l'unité, de la distance du robot à l'obstacle.
[ViDÉO5](#) – [BLOCS à PROGRAMMER5](#) - [CoRRECTION5](#)
- Défi 6 : Afficher sur l'écran le nom de la couleur détectée et s'arrêter à la marque noire
[ViDÉO](#) – [BLOCS à PROGRAMMER](#) - [CoRRECTION](#)

Activité 3 : Suivi de ligne et détection de ligne

- Défi 7 : Après appui sur le bouton, le robot s'arrête sur la ligne noire
[ViDÉO7](#) – [BLOCS à PROGRAMMER7](#) - [CoRRECTION7](#)
- Défi 8 : Après appui sur le bouton, le robot suit la ligne noire
[ViDÉO8](#) – [BLOCS à PROGRAMMER8](#) - [CoRRECTION8](#)
- Défi 9 : Après appui sur le bouton, le robot suit la ligne noire et s'arrête si obstacle détecté
[ViDÉO](#) – [BLOCS à PROGRAMMER](#) - [CoRRECTION](#)

Activité 4 : Simulation d'un robot aspirateur

- Défi 10 : Simuler le fonctionnement d'un robot aspirateur. Le robot parcourt toute la surface délimitée par un trait noir.
[ViDÉO10](#) – [BLOCS à PROGRAMMER10](#) - [CoRRECTION10](#)
- Défi 11 : Ajouter une détection d'obstacles à la simulation du robot aspirateur
[ViDÉO11](#) – [BLOCS à PROGRAMMER11](#) - [CoRRECTION11](#)