

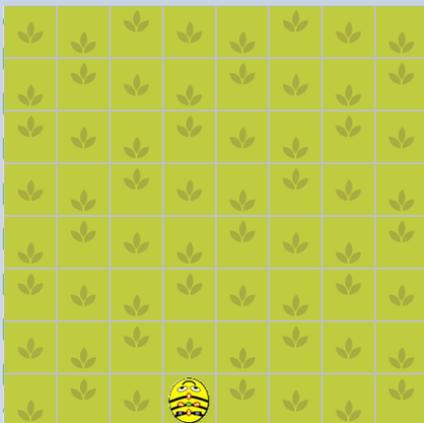
## BeeBot et Edison : deux robots au service de l'apprentissage du codage

- Nécessité de coder
- Manier les algorithmes depuis le cycle 1
- Progressivité des apprentissages

### Jouer et apprendre avec BeeBot



- Déplacement relatif



- Codage des instructions



- Programmation des déplacements

- Quels usages en classes ?

- Découverte des mouvements de la Beebot avec le corps , tout en travaillant en parallèle le lexique spécifique (Cf paragraphe sur le lexique). Les élèves "font le robot" ou commandent un "élève-robot". Cela permet ensuite un transfert lors du travail avec la Beebot.
- Découverte de la Beebot en tant qu'objet numérique  
La touche "Efface" : L'utilisation et l'importance de la touche efface dans la programmation de l'objet.

- Premiers déplacements : sur une ligne .  
Situation-problème : comment amener la Beebot sur la fleur ?  
Le défi est adapté en fonction des connaissances numériques des élèves (différenciation en fonction de la connaissance des nombres).  
Cette séance est répétée afin de permettre aux élèves de bien s'appropriier les différentes actions qu'il est possible de réaliser avec la Beebot.
- La touche "Pivote".  
La touche Pivote nécessite un travail spécifique et approfondi : la Beebot s'oriente vers la droite ou vers la gauche en restant sur place sans changer de case.  
Cela explique notre choix sur l'utilisation du terme PIVOTE au lieu de tourne qui lui induit un déplacement.  
Ce point a été travaillé sur un quadrillage et des cases de couleur.
- Déplacements sur un quadrillage .  
La séance se déroule en utilisant les cartes.  
L'élève montre le chemin que doit prendre Beebot avec son doigt, il pivote dans la case. Il s'agit ici d'anticiper les déplacements à coder.  
Les élèves de GS préparent ensuite avec les cartes les déplacements de la Beebot

## Le lexique

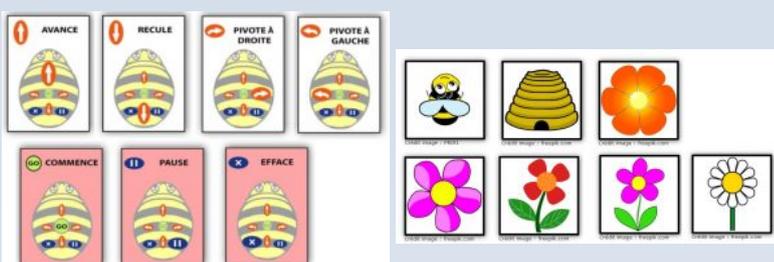
Il est important de veiller à utiliser un vocabulaire précis et commun tout au long des séances menées.

→ Les élèves doivent bien oraliser leurs actions pour renforcer l'acquisition du lexique. ("avancer", "reculer", " pivoter à droite ", "pivoter à gauche", "touche GO", "touche EFFACE")

Cela permet d'éviter l'utilisation de termes comme "monter", "tout droit"...

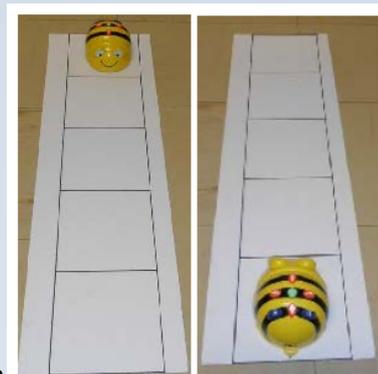
Pour se familiariser avec les cartes, présenter les cartes "Avance", "Efface" et "Go" en grand format dès qu'elles sont utilisées par les élèves.

## Les supports



## Les prolongements possibles

- Apprendre à décoder un déplacement écrit (programme)
- Garder une trace écrite qui code un déplacement (bande avec des cases).



L'appli

## Programmer avec Edison

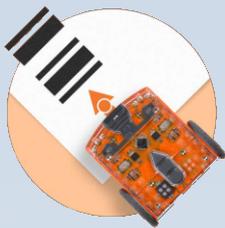
### Description du robot



### Quatre modes de programmation



- Par codes-barres



Ce type de programmation s'adresse à des élèves à partir de 4 ans. Les codes-barres correspondent à des comportements différents pour le robot. Pour attribuer un comportement au robot, il suffit de le faire passer sur le code-barre pour qu'il le scanne.



Avance au bruit



Evite les obstacles

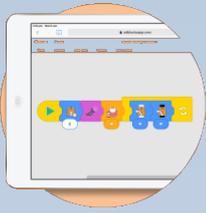


Suit une ligne



Reste dans un périmètre

## Par assemblage de blocs figuratifs :



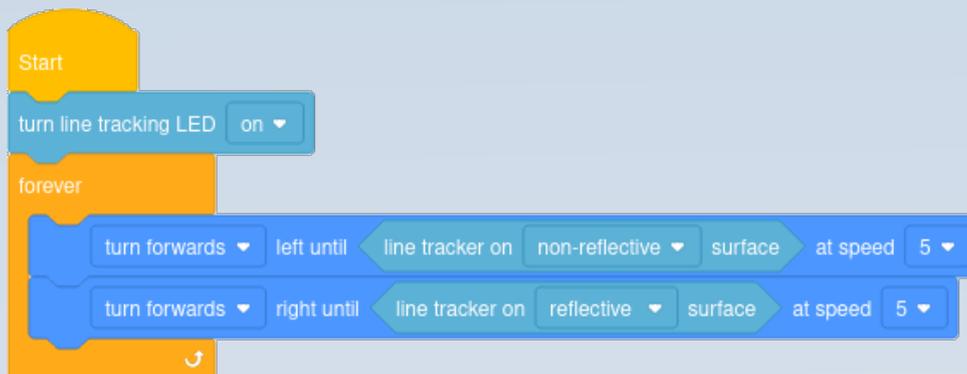
Ce type de programmation est destiné à des enfants de 8 à 12 ans. Il consiste en l'élaboration d'algorithmes permettant de faire fonctionner le robot avec des comportements personnalisés. Les algorithmes sont composés d'une série de blocs figuratifs comme dans Scratch Junior. La programmation se fait via l'application en ligne [EdBlocks](#).



## Par assemblage de blocs explicites:



Cette programmation s'adresse à des élèves à partir de 10 ans. Les algorithmes sont ici lus de haut en bas et sont constitués de blocs explicites (en anglais) et propose une interface plus avancée et plus évoluée, comme dans Scratch. La programmation se fait via l'application en ligne [EdScratch](#).



## Par programmation en langage Python :



Ce langage de programmation est accessible à des élèves à partir de 13 ans. La programmation du robot se fait en élaborant un programme en ligne de commande à l'aide de fonction prédéfinies en langage Python. La programmation se fait via l'application en ligne [Ed.Py](#).

```

1
2 #-----Setup-----
3
4 import Ed
5
6 Ed.EdisonVersion = Ed.V2
7
8 Ed.DistanceUnits = Ed.TIME
9 Ed.Tempo = Ed.TEMPO_MEDIUM
10
11 #-----Your code below-----
12
13 while True:
14     Ed.PlayBeep()
15     Ed.LeftLed(Ed.OFF)
16     Ed.RightLed(Ed.ON)
17     Ed.Drive(Ed.SPIN_RIGHT, 5, 350)
18     Ed.TimeWait(20, Ed.TIME_MILLISECONDS)
19     Ed.PlayBeep()
20     Ed.LeftLed(Ed.ON)
21     Ed.RightLed(Ed.OFF)
22     Ed.Drive(Ed.SPIN_LEFT, 5, 350)
23     Ed.TimeWait(20, Ed.TIME_MILLISECONDS)
24

```