

# Coder des déplacements au Cycle 2



# Ce que disent les programmes

## Compétence(s):

« (Se) repérer et (se) déplacer en utilisant des repères et des représentations

- S'orienter et se déplacer en utilisant des repères
- Réaliser des déplacements dans l'espace et les coder pour qu'un autre élève puisse les reproduire.
- Programmer les déplacements d'un robot ou ceux d'un personnage sur un écran »

*Au cycle 2, les élèves apprennent à « coder et décoder pour prévoir, représenter et réaliser des déplacements dans des espaces familiers, sur un quadrillage, sur un écran ». Ces déplacements ont lieu dans des espaces réduits en début de cycle (classe ou école) pour s'étendre progressivement tout au long du cycle jusqu'au quartier ou village pour lesquels ils pourront utiliser des plans. À partir du CE1, les élèves sont invités à coder des déplacements à l'aide d'un logiciel de programmation adapté.*



**→ L'objectif est d'anticiper un déplacement complet, de prévoir à l'avance l'ensemble des instructions permettant d'obtenir un déplacement complexe souhaité (= développer l'abstraction)**



MINISTÈRE  
DE L'ÉDUCATION  
NATIONALE, DE  
L'ENSEIGNEMENT  
SUPÉRIEUR ET DE  
LA RECHERCHE

éduscol Informer et accompagner  
les professionnels de l'éducation

CYCLES 2 3 4

MATHÉMATIQUES

Espace et géométrie

## Initiation à la programmation aux cycles 2 et 3

### Introduction

L'initiation à la programmation constitue une nouveauté importante pour les cycles 2 et 3. Elle s'inscrit dans les objectifs du socle commun de connaissances, de compétences et de culture, où il est précisé, dans le domaine 1 (les langages pour penser et communiquer) : « L'élève sait que des langages informatiques sont utilisés pour programmer des outils numériques et réaliser des traitements automatiques de données. Il connaît les principes de base de l'algorithme et de la conception des programmes informatiques. Il les met en œuvre pour créer des applications simples. ». Il s'agit aux cycles 2 et 3 d'amorcer un travail qui sera poursuivi au cycle 4.

L'initiation à la programmation apparaît dans les programmes au sein du thème Espace et géométrie en lien avec l'objectif « (Se) repérer et (se) déplacer en utilisant des repères » au cycle 2 et « (Se) repérer et (se) déplacer dans l'espace en utilisant ou en élaborant des représentations » au cycle 3.

La diversité des équipements sur le territoire nécessite de s'appuyer sur des activités faisant appel des supports variés :

- sans matériel spécifique, « en débranché » ;
- des robots programmables ;
- des applications en ligne utilisables sur ordinateurs ou tablettes ;
- des logiciels pouvant être installés sur des ordinateurs ou des tablettes.

L'initiation à la programmation pourra être une opportunité pour des travaux interdisciplinaires :

- avec le champ questionner le monde au cycle 2, par exemple, autour de la question du repérage ;
- ou avec le français, dans le développement des usages du langage oral ou écrit, notamment en créant des histoires illustrées par de courtes animations créées par les élèves ;
- ou encore en langues vivantes en créant des animations où les personnages dialoguent en langue étrangère ;
- etc.

Les situations et les outils présentés dans ce document sont, en fonction de ce qui a été fait précédemment, aussi bien conçus pour des élèves de cycle 2 que de cycle 3, sauf si le contraire est précisé.

# Document d'accompagnement Eduscol, *Initiation à la programmation aux cycles 2 et 3*

<https://eduscol.education.fr/document/15409/download>



# Stratégies d'enseignement

## Démarche

Malgré des objectifs plutôt modestes, la programmation est un apprentissage à part entière. La phase de découverte ne doit pas être négligée et doit permettre aux élèves pour s'approprier les outils qui leurs sont proposés. La présentation des outils et des différentes étapes à réaliser doit permettre de mettre la classe dans une dynamique de projet, qui facilitera l'investissement des élèves dans leurs apprentissages.

Une fois la phase de découverte passée, l'enseignant joue sur les variables didactiques propres aux outils pour initier les élèves à la programmation, comme le nombre de pas pour le déplacement d'un robot ou d'un personnage dans une grille ou sur un écran, le nombre d'instructions nécessaires, l'environnement, les supports, les instructions disponibles, etc.

Il est important d'inviter les élèves à expliciter les programmes de déplacement qu'ils conçoivent pour gagner en abstraction et en autonomie.

Des synthèses régulières, accompagnées d'institutionnalisations écrites, sont nécessaires à la construction des connaissances.

## Précision du langage

Les activités proposées dans ce document reposent sur le fait de décrire des déplacements.

Le lexique est simple, proche du langage naturel de l'élève, puisqu'il s'agit de programmer une suite d'instructions élémentaires en utilisant les mots ou expressions usuels suivants : « s'orienter vers », « pivoter », « tourner », « avancer », « haut », « bas », « droite », « gauche », « nord », « ouest », etc.

Concrètement, l'enseignant organise des temps pour que les élèves explicitent leurs procédures ; ce qu'ils ont fait en premier, ce qu'ils ont fait ensuite, etc. Il met à disposition les supports qui facilitent ce temps de langage : étiquettes, dessins, photos, tablettes, etc.

L'enseignant privilégie le lexique déjà présent dans le support : le code des robots, les blocs déjà présents dans ScratchJr ou Scratch, etc. Des séances décrochées peuvent être mises en place avec des objectifs spécifiques pouvant relever d'autres domaines d'apprentissage et au service de l'activité de programmation. Par exemple, si le besoin s'en fait sentir, pour améliorer la façon de formuler des ordres, il peut être prévu en français des activités sur l'impératif présent avec les élèves de cycle 3.

## Déplacement relatif — déplacement absolu

Dès les premières activités, les élèves sont confrontés aux différentes façons de préciser les déplacements.

- On parle de **déplacement absolu**, lorsque l'effet des instructions ne dépend pas de l'orientation initiale du « mobile » qui les reçoit. Par exemple : « tourne-toi vers le tableau », « avance de deux pas vers la fenêtre », « avance de trois pas vers l'est » ;
- On parle de **déplacement relatif**, lorsque l'effet des instructions dépend de l'orientation initiale du « mobile » qui les reçoit. Par exemple : « effectue un quart de tour vers la droite », « avance de trois pas ».

La notion de déplacement relatif peut poser problème, en particulier pour les élèves non latéralisés. Les difficultés rencontrées au quotidien pour lire des cartes papier et suivre les instructions d'un GPS l'illustrent assez bien.

« allocentré »

« autocentré »

**Document d'accompagnement Eduscol, *Initiation à la programmation aux cycles 2 et 3***

<https://eduscol.education.fr/document/15409/download>

→ **Le déplacement relatif implique que le programmeur se mette « à la place » du robot.**

# Quelle progressivité ?

## En principe:

1. Coder et décoder des déplacements absolus (exemple d'activité débranchée : la fusée)
2. Coder et décoder des déplacements relatifs (activités branchées avec Blue Bot)
3. Coder et décoder des déplacements relatifs **ET** absolus (robot + utilisation des points cardinaux pour programmer déplacements sur cartes au CE2)

## Mais....

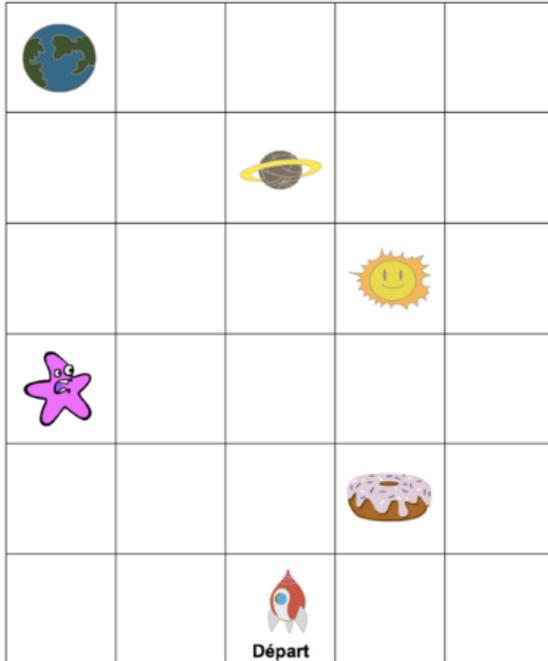
Risque de différer l'usages des robots (déplacements relatifs) alors qu'introduction possible dès la Maternelle !

## Donc...Un choix à opérer:

- Soit proposer des situations identifiées comme « absolues » ou « relatives » (= donner le langage à utiliser);
- Soit proposer une situation de découverte ouverte, et à partir des programmes produits par les élèves, les confronter aux différentes manières de coder (absolu et relative). Puis, réfléchir aux instructions les plus appropriées selon les cas.

# Deux situations de codage de déplacements absolus en débranché

Dès le début du cycle : La fusée



## Les déplacements de la fusée

Pour faciliter les choses, le déplacement s'effectue sur un quadrillage.

Le codage des déplacements de la fusée repose ici sur des flèches :

- ↑ : monte d'une case, c'est-à-dire avance d'une case vers le haut (de la feuille) ;
- ↓ : descend d'une case ;
- ← : avance d'une case vers la gauche ;
- : avance d'une case vers la droite.

## Activité 1

### OBJECTIF

Traduire le déplacement proposé à l'aide du codage.

### Consignes (présentées à l'oral ou à l'écrit)

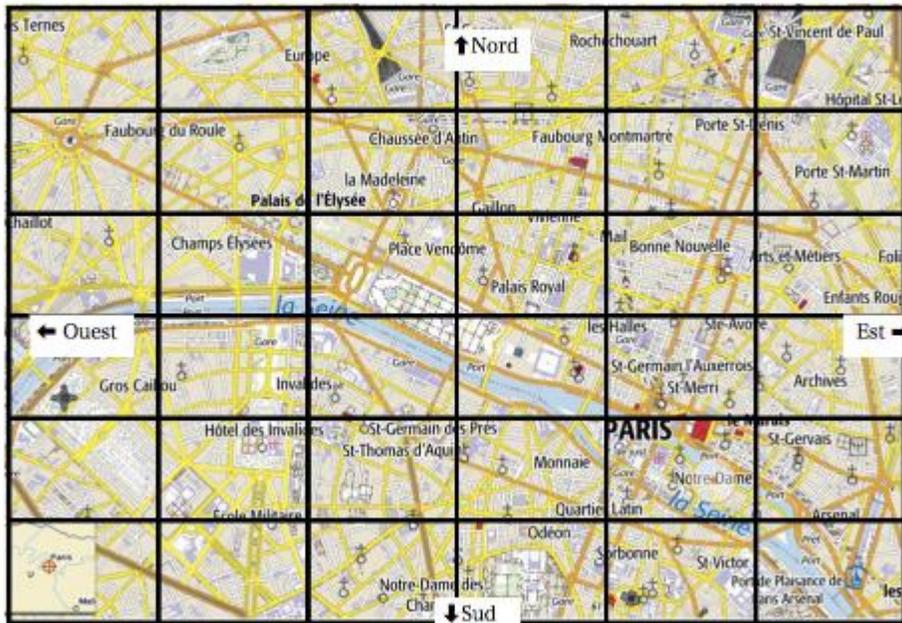
La fusée part de la case DÉPART. Le trajet de la fusée est le suivant :

- elle monte de quatre cases ;
- elle avance d'une case vers la droite ;
- elle descend de trois cases.

# Deux situations de codage de déplacements absolus en débranché

En fin de cycle : Découvrir le monde

## Activité 2 : Au cœur de Paris



### Partie 1 : le déplacement est codé en français

Repère la case où se trouve la cathédrale Notre Dame de Paris. À partir de cette case, suis le déplacement suivant : « Avance d'une case vers le nord. Avance de 4 cases vers l'ouest.

À ton avis, quel monument est représenté par la croix grise qui se trouve dans la case où tu viens d'arriver ?

.....

Repère maintenant la case où se trouve l'Hôtel des Invalides.

À partir de ce point, suis le déplacement suivant : « Avance d'une case vers le sud. Avance de deux cases vers l'est. Avance de 5 cases vers le nord. Avance de deux cases vers l'est.

Quel(s) monument(s) ou construction(s) est (sont) indiqué(s) dans cette case ?

.....

[http://cache.media.education.gouv.fr/file/Initiation\\_a\\_la\\_programmation/88/8/R\\_A16\\_C2\\_C3\\_MATH\\_annexe\\_1\\_3\\_en\\_debranche\\_decouvrir\\_monde\\_624888.pdf](http://cache.media.education.gouv.fr/file/Initiation_a_la_programmation/88/8/R_A16_C2_C3_MATH_annexe_1_3_en_debranche_decouvrir_monde_624888.pdf)

**Coder des déplacements relatifs avec BeeBot /BlueBot**

## Exemple de séquence:

S1- Qu'est-ce qu'un robot ?

S2- Découverte de Blue Bot

S3- Je suis un robot ( activité EPS )

S4 et suivantes – Défis de plus en plus complexes

S? – Programmer à distance avec la barre bluetooth

→ *On peut proposer aux élèves de « jouer » au robot avant la découverte de Blue Bot. Dans ce cas, l'objectif n'est plus de réinvestir les instructions Blue Bot mais de confronter les élèves au problème de la meilleure manière de coder des déplacements (comment transmettre le message ? Quelles instructions utiliser ?).*





# Une activité de tri d'images

**ALTERNATIVE: Dessinez un robot.**

## Déroulé

Réaliser des groupes d'élèves et leur distribuer à chacun un jeu d'images 'robot ou pas'.

Vous avez devant vous des images qui représentent des objets qui existent autour de nous. En fonction de ce que nous savons nous allons essayer de deviner si ce sont des robots ou pas en réalisant 3 tris :

### 1- Vivant / non vivant

Extrait de...



Les élèves peuvent se trouver devant des objets qu'ils ne connaissent pas comme le rover martien.

Choisissez alors une zone pour les mettre de côté et en parler à la fin.

Quelques indices : ce qui est vivant a besoin de dormir, de manger, de boire...

Les robots ne sont pas vivants, ce sont des machines construites par l'homme. Elles n'ont pas besoin de dormir, manger, boire et sont faites de matériaux physiques comme le plastique ou le métal et non de tissus biologiques comme la peau, les feuilles, les poils, les muscles...

Tous ce qui est vivant est mis dans la colonne de gauche...

...et ce qui n'est pas vivant, dans la colonne de droite.

Mettez de côté les éléments de la colonne de gauche (vivants) et récupérez ceux de la colonne de droite (non-vivants).



## 2. Ont-ils besoin d'une source d'énergie ?

Comment savoir ? Vérifiez si vous les voyez connectés à un fil et une prise électrique ou bien s'ils ont une batterie. Echangez avec vos camarades pour vous mettre d'accord.

De la même manière que nous avons besoin de manger et boire pour vivre, les robots ont besoin d'énergie pour fonctionner.

Tout ce qui utilise une source d'énergie est mis dans la colonne de gauche ...

...et ce qui n'en utilise pas dans la colonne de droite.

Mettez de côté cette fois-ci, les éléments de la colonne de droite (non-électriques) et gardez les éléments de la colonne de gauche (électriques).



### 3- Peut-il réaliser un programme ?

Pas évident comme question... Lorsque vous appuyez sur l'un de ses boutons, est-ce que votre machine sait réaliser plusieurs actions en une fois (un programme) ou alors devez-vous intervenir pour lui faire réaliser une action à la fois ?

Contrairement aux autres machines, les robots savent effectuer des actions programmées, c'est ce qui fait leur particularité.

Tous les objets qui peuvent réaliser un programme sont mis dans la colonne de gauche ...

...et ceux qui ne peuvent pas le faire, dans la colonne de droite.

Mettez de côté les éléments de la colonne de droite (non-programmables) et gardez ceux de la colonne de gauche (programmables).

*Les objets qu'il nous reste sont donc des robots !*

### Conclusion Séance 1

Un robot est une machine physique et automatique qui réalise plusieurs actions programmées, les unes à la suite des autres lorsqu'on lui en donne l'ordre. De nos jours, les robots deviennent de plus en plus communs et nous les côtoyons avec les objets de tous les jours.

Il s'agit, dans un premier temps, de manipuler le robot, afin d'en connaître les premières fonctionnalités, d'amener les élèves à développer un vocabulaire spécifique et de coder des déplacements élémentaires.

Extrait de ...



The image shows the cover of a document from the French Ministry of National Education. It features the French flag logo and the text 'MINISTÈRE DE L'ÉDUCATION NATIONALE, DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE'. The main title is 'MATHÉMATIQUES' with a sub-title 'Espace et géométrie'. It is specifically for 'CYCLES 2 3 4' and 'Annexe 2.2 : Premières séances avec Bee-Bot, Blue-Bot et Pro-Bot'.

[http://cache.media.education.gouv.fr/file/Initiation\\_a\\_la\\_programmation/89/7/RA16\\_C2\\_C3\\_MATH\\_annexe\\_2\\_2\\_robots\\_premiere\\_seance\\_624897.pdf](http://cache.media.education.gouv.fr/file/Initiation_a_la_programmation/89/7/RA16_C2_C3_MATH_annexe_2_2_robots_premiere_seance_624897.pdf)

## Présentation générale

Avec les robots, les déplacements sont **relatifs**. Le robot ne peut pas savoir où est le nord, ni où est la porte par exemple, il se déplace par rapport à sa position : il avance, recule, pivote à gauche ou à droite.

Cette ressource porte sur les premières séances avec les robots.

1. **La première étape consiste à découvrir les différentes fonctionnalités de déplacement des robots** : avancer, pivoter à droite, pivoter à gauche, reculer. L'objectif est d'amener les élèves à acquérir un vocabulaire spécifique en prenant conscience que, pour chaque robot, à chaque bouton correspond une action unique.
2. **La deuxième étape consiste à programmer le robot** pour le faire aller d'un point à un autre de l'espace de travail avec un petit nombre d'instructions successives.
3. **Dans un troisième temps**, les élèves seront amenés à produire des programmes comportant un plus grand nombre d'instructions.

Pour chacun des robots, on peut privilégier des univers spécifiques à celui-ci. Par exemple, pour donner du sens aux déplacements de Bee-Bot et Blue-Bot, on peut utiliser des tapis avec des fleurs à butiner ou des ruches à visiter.

**Exemple** : un tapis à fleurs réalisés pour la Bee-Bot

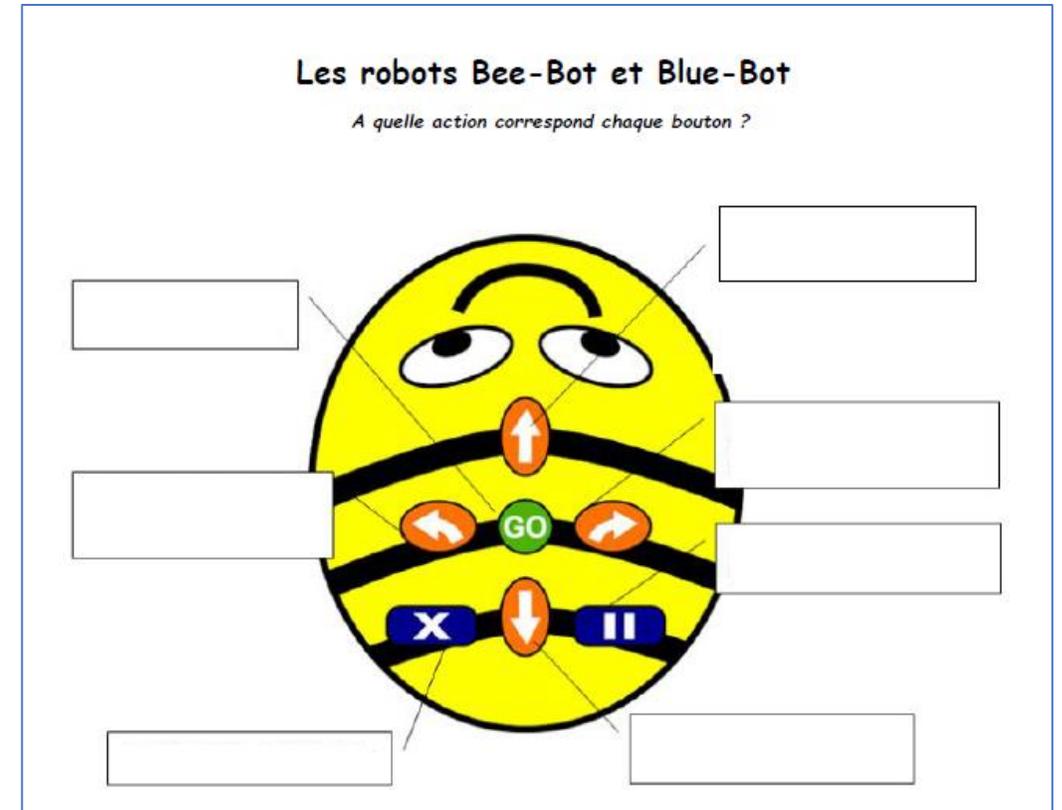


# Séance de découverte de Bee-Bot / Blue-Bot

Objectifs: - Découvrir le fonctionnement du robot

- Se mettre d'accord sur le décodage oral des instructions (précision du lexique)

1. Observer le robot, faire des hypothèses sur son fonctionnement (recherche individuelle)
2. Manipuler le robot pour tester les hypothèses. Associer chaque bouton à l'action réalisée (en petit groupe).
3. Mise en commun. Réalisation d'une affiche. Validation du décodage oral des instructions:



- [Fiche élève](#)
- [Trace écrite](#)

# « Pivoter » Vs « Tourner »

Extrait de...



Comment le faire se déplacer sur une case vers la gauche / la droite ?

Il faut appuyer sur une des touches directionnelles gauche ou droite, appuyer sur *avancer* puis GO.

Le robot ne tourne pas, il pivote. Quelle est la différence entre les deux termes ?

Tourner (comme une voiture), c'est changer de direction tout en avançant.

Pivoter c'est changer de direction en restant sur place. Pour un humain, cela correspond à tourner sur soi-même (regarder son épaule gauche ou droite puis aligner son corps avec sa tête pour regarder en face). Pour changer de case, il faut ensuite rajouter l'ordre d'avancer de nouveau.

**Pour les plus jeunes, il est possible de mettre le bracelet jaune sur le poignet gauche lorsque l'on parle des pivots droite et gauche.**

**Il sera ainsi possible d'associer la couleur jaune à la main gauche, nous retrouverons cette couleur sur la touche gauche et les cartes de commandes 'pivot gauche' du robot.**

<http://ienlille1lambertart.etab.ac-lille.fr/files/2023/01/LIVRET-DES-SEANCES-TOTAL.pdf>

**Dans un deuxième temps, l'enseignant peut proposer des activités sous la forme de défis qui conduisent les élèves à programmer les déplacements des robots avec l'ajout de contraintes.**

Extrait de ...

The image shows the cover of a document from the 'éduSCOL' collection. At the top left, it says 'éduSCOL Informer et accompagner les professionnels de l'éducation'. To the right, there are three circular icons labeled 'CYCLES 2', '3', and '4'. Below this, a dark blue banner contains the word 'MATHÉMATIQUES' with a white arrow pointing right. Underneath the banner, a light blue box contains the text 'Espace et géométrie'. The main title 'Initiation à la programmation' is centered in a large, bold, teal font. Below the title, the subtitle 'Annexe 2.3 : Premiers défis avec Bee-Bot et Blue-Bot' is written in a smaller teal font.

Les élèves devront alors coder les déplacements d'un robot en utilisant les symboles adaptés, anticiper son déplacement, procéder par essais-erreurs pour parvenir au déplacement de ce dernier.

Par exemple, sur un tapis avec une fleur, l'enseignant demande à ce que l'abeille butine la fleur :

- en utilisant toutes les commandes ;
- en interdisant certaines commandes (comme « avancer ». L'enfant doit comprendre que pour avancer le robot d'un pas, il doit d'abord lui demander de pivoter deux fois dans le même sens puis lui demander de reculer d'un pas) ;
- en passant par des endroits précis, en évitant des obstacles.

## **Proposition d'activités d'approfondissement avec la Bee-Bot**

Si plusieurs fleurs sont disposées sur le tapis, l'enseignant peut demander aux élèves de butiner toutes les fleurs :

- sans contraintes dans un premier temps (cycle 2) ;
- en optimisant le programme : les élèves doivent alors produire le programme le plus court (cycle 3) ;
- avec un nombre d'essais erreurs réduit (cycles 2 et 3).

# Des missions à la difficulté progressive

REPARTITION SUR LE TAPIS

	Maison		Maison
Garage			
	Maison		Boulangerie
Maison		Poste	Maison
	Maison bleue		

<https://ienlille1marcqenbaroeul.etab.ac-lille.fr/files/2023/01/missions-BB.pdf>

## MISSION N°1

### La réparation



Bee Bot est dans sa maison bleue.  
Il a rendez-vous au garage pour une réparation.

<u>Départ</u> : Maison bleue	<u>Arrivée</u> : Garage
------------------------------	-------------------------

## MISSION N°9

### La réparation (5)



Bee Bot est dans sa maison bleue.  
Il a rendez-vous au garage pour une réparation car la touche « avancer » ne fonctionne plus.  
Une fois la réparation terminée, Bee Bot rentre chez lui.

<u>Départ</u> : Maison bleue	<u>Arrivée</u> : Maison bleue
------------------------------	-------------------------------

#### Contraintes :

- . Ne pas utiliser la touche « avancer » à l'aller.
- . Toutes les touches peuvent être utilisées au retour.

Annexe 13 : un support pédagogique pour Blue-bot (exemple réalisé en lien avec l'éducation artistique suite à un travail autour de Van Gogh)

La visite au musée :

Le parcours est inspiré d'un travail réalisé par les élèves de Cours Moyen de l'école de Julienrupt. Suite à la découverte de l'œuvre de Van Gogh, Nuit étoilée (cf. annexe I - expérimentations en classe)

Ici, les élèves ont proposé deux consignes :

**Niveau facile** : Blue-bot doit passer devant quatre œuvres de Van Gogh.

**Niveau difficile** : Blue doit passer devant toutes les œuvres et terminer par la nuit étoilée.

Les variables didactiques :

Le support et le parcours sont évolutifs. On peut simplifier ou complexifier la consigne. On peut modifier le support en ajoutant ou en enlevant des photos d'autres artistes par exemple (cf. annexe II - les supports).

Quelques exemples de consignes :

*Blue-bot doit voir des tournesols, un lit et un semeur.*

*Blue-bot doit voir uniquement deux portraits.*

*Blue-bot doit voir une scène de genre et un portrait en moins de huit déplacements...*

# Inventer des défis, créer de nouvelles cartes missions

<p>MISSION N°</p> <p>Titre</p>  <p>Indique le point de départ du robot et ce qu'il doit faire.</p> <table border="1"><tr><td>Départ :</td><td>Arrivée :</td></tr></table>	Départ :	Arrivée :	<p>REPARTITION SUR LE TAPIS</p> <table border="1"><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																								
Départ :	Arrivée :																										
<p>MISSION N°</p> <p>Titre</p>  <p>Indique le point de départ du robot et ce qu'il doit faire.</p> <table border="1"><tr><td>Départ :</td><td>Arrivée :</td></tr><tr><td colspan="2">Contraintes :</td></tr></table>	Départ :	Arrivée :	Contraintes :																								
Départ :	Arrivée :																										
Contraintes :																											

*On peut imaginer que les CE1 créent des défis pour les CP, et les CE2 des défis pour les CE1...*

[Cartes missions vierges](#)

## Séance 6 : créer des parcours, des programmes et des défis pour programmer Blue-bot...

Obj : Se repérer et déplacer dans un espace un objet programmable en utilisant ou en élaborant des représentations ; fabriquer un support de déplacement

Phases	Déroulement/ consignes	Temps	Matériel	Groupement
Rappel	<ul style="list-style-type: none"><li>- Rappel du support pédagogique proposé aux élèves dans le cadre d'une séance d'arts visuels.</li><li>- Mise en exergue du rôle de la consigne pour définir les critères de réussite.</li></ul>	5'	Annexe 13 + Blue-bot	Collectif
Définition de l'objectif	Demander aux élèves de choisir un sujet d'apprentissage et de définir la « mission » de Blue-bot pour mener à bien le parcours. La rédaction d'un objectif et d'une consigne vont permettre aux élèves de mieux définir la pertinence de leur support. L'enseignant observe et écoute les propositions, il sera sans doute nécessaire de bien remettre les élèves en perspective sur la tâche qu'ils mènent et l'objectif visé.	10' à 15'		Binôme
Fabrication du support	Une fois l'objectif et la consigne rédigée, les élèves peuvent procéder à la fabrication d'un support au brouillon pour identifier le nombre de cases nécessaires et leurs besoins en matériel (photos, dessins, calculs...). Selon les besoins, un temps de collecte d'informations peut être nécessaire. Une fois collectée, les élèves passent à la création du support. Selon le temps, les élèves peuvent réaliser le tracé du quadrillage en veillant au bon respect des contraintes dimensionnelles (15 x 15 ; nombre de cases). Sinon, on utilise le support transparent comme trame de quadrillage.	45' à 1h30	Papier format affiche, crayons, règle (de tableau), photos...	Binôme
Mise en commun et validation	Présentation des supports et des consignes au groupe classe. Les supports peuvent ainsi faire l'objet d'un atelier en autonomie au sein de la classe pour le réinvestissement d'un apprentissage.  Les élèves rentrent les instructions pour que Blue-bot réalise un parcours qui répond à la consigne. Une trace écrite	10'	Blue-bot + quadrillage + objets ou cartes à atteindre	Collectif  Individuel

Tiré de L.Bouhours, *Débuter en programmation avec Blue Bot*:

[https://www.generationrobots.com/media/sequence\\_debuter\\_en\\_programmation\\_avec\\_blue-bot.pdf](https://www.generationrobots.com/media/sequence_debuter_en_programmation_avec_blue-bot.pdf)

# Générateur de grilles pour s'exercer en débranché

<https://classedeflorent.fr/outils/index.php#beebot>

Prénom : \_\_\_\_\_ Date : \_\_\_\_\_

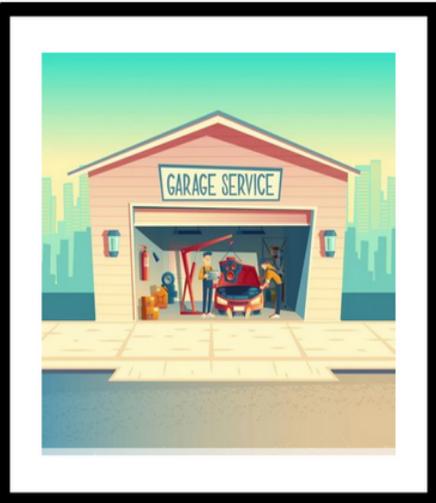
				
				
				

Mes déplacements

# Générateur d'images 15x15cm pour tapis quadrillé

<https://micetf.fr/cases4bb/>



Afficher un texte

Texte à afficher

Taille des caractères Position du texte

Insérer une image

Uploader une image

Valider la carte

# Programmer à distance: la barre de programmation Blue Bot



Photo TTS Group Ltd®



La barre de programmation possède dix **emplacements** pour insérer des cartes séquentielles. Les cartes sont recto-verso et permettent de coder en ligne ou verticalement comme l'application.

Les cartes incluses avec la barre de programmation permettent les déplacements de base. Des **kits d'extension** avec de nouvelles cartes sont disponibles pour permettre des fonctions plus avancées comme les **boucles** ou les **répétitions**.

La connexion entre la barre programmable et le Blue-Bot se fait en appuyant sur la touche « Bluetooth »

de la barre lorsque Blue-bot est allumé. Une fois connecté, les diodes bleues du Blue-bot s'allument.

L'enregistrement d'un déplacement se fait à l'aide des cartes séquentielles. Une fois que les cartes sont placées sur la barre, il faut appuyer sur Go pour envoyer le programme au Blue-bot qui réalise le déplacement.

**Intérêts: visualisation du programme / visualisation de l'instruction en cours d'exécution (diode) / repérage des erreurs éventuelles (facilitation du débogage)**



Photo TTS Group Ltd®



Photo TTS Group Ltd®

# Codage de déplacements: variables didactiques

- Support: papier / sur écran / avec robot / vécu (jeu du robot)
- Instructions orales / instructions écrites (langage naturel ou codage fléché)
- Longueur et complexité du parcours
- Introductions de contraintes liées au quadrillage : cases obligatoires, cases à éviter
- Introduction de contraintes liés aux déplacements autorisés : coder sans utiliser « tout droit » ou « pivoter à droite »
- Avec le robot, médiation de l'écrit : étiquette fléchées / bande programme / barre de programmation
- Différenciation des rôles des élèves (programmeur, ingénieur, débogueur...)
- En débranché :cases du quadrillage /nœuds du quadrillage
- Types d'activités:
  - Codage (case départ et d'arrivée connues → écrire le programme);
  - Décodage (case départ et programme donnés → déterminer case d'arrivée);
  - Débogage (programme erroné donné → repérer l'erreur, corriger)
- Coordination avec les déplacements d'un autre robot (utilisation de la pause)
- ...

# **Prolongements**

# Le jeu du robot



<https://www.youtube.com/watch?v=9AtmJ9mTaB0&list=PLWvGMqXvyJAPSMFgCiy6qVHW9bAPu93X5&index=2>



# Le jeu du robot

## Marcel le Robot

Le document que vous êtes en train de consulter n'est pas une référence très finalisée, ni un guide strict à suivre. Il regroupe le plan du tapis, des idées de programmes ainsi que quelques idées d'étapes pour réaliser le jeu du robot, appelé ici Marcel, en référence à la première école où cette activité a été testée : l'école primaire spécialisée Marcel Leroy à Nancy. La diffusion de ce document est libre, vous pouvez suggérer des améliorations/enrichissements à [marie.duflot-kremer@loria.fr](mailto:marie.duflot-kremer@loria.fr). Si ce document vous a été utile, vous pouvez également me le signaler car j'envisage de mentionner sur ma page médiation les écoles/associations/etc. qui ont testé et approuvé l'activité. La page <https://members.loria.fr/MDuflot/> permet de trouver (section médiation/activités) une liste d'autres activités pour faire découvrir différents aspects de l'informatique, réalisables en grande majorité sans ordinateur.

### Étape 1 Découverte du langage

Cette étape peut être réduite ou supprimée en fonction de l'âge des participants. D'expérience elle est appréciée et utile pour les petits (testée principalement en Grande section et CP). Tous les enfants se mettent sur le tapis, chacun sur un point, et suivent les indications données par l'animateur, à choisir entre un pas à droite, à gauche, en avant ou en arrière. Dès qu'un enfant se retrouve dans le décor (rivière, lac, forêt ou volcan) ou bien sort du tapis, il se met sur le côté. Quand tout le monde est sur le côté on peut recommencer.

Bien préciser :

- qu'un pas c'est se déplacer d'un point, pas deux
- rappeler gauche et droite pour ceux qui pourraient se tromper
- pas de déplacement en diagonale
- on reste toujours dans la même orientation. On fait des pas de côté mais on ne tourne pas sur soi-même
- on est un robot, donc on exécute les instructions sans les changer, même si elles nous envoient dans le décor

Pour la suite de l'activité, on peut aider les enfants ayant des problèmes de latéralisation en surlignant sur les programmes de deux couleurs différentes les flèches correspondant aux pas ou aux quarts de tours à droite et à gauche, puis en mettant des bracelets (un morceau de tissu noué fait l'affaire) de couleur correspondante sur les poignets de l'enfant. La lecture du programme ressemble alors à : "un pas en avant, un pas du côté vert, un pas en arrière, un pas du côté rose, etc."

### Étape 2 Exécution de programmes

Des exemples de programmes sont donnés en fin de document, à découper (et plastifier si on veut qu'ils durent plus longtemps).

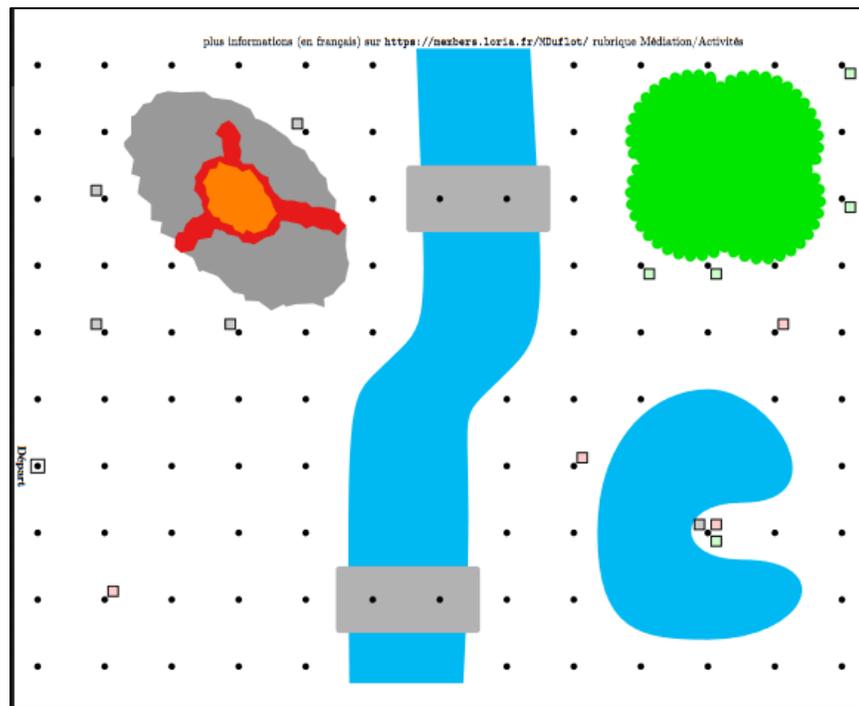
On met le point de départ comme montré sur le plan (tout en bas, 7ème point à partir de la gauche). Tous les programmes vont démarrer de là.

Pour chaque programme on va répartir la tâche entre un enfant (ou plusieurs) qui va jouer le rôle de la mémoire de l'ordinateur : lire les instructions une par une, et un autre qui va jouer le processeur de l'ordinateur (ou le robot) et donc exécuter le programme. Il faut bien préciser que les deux jouent le rôle d'un ordinateur, et que donc ils doivent effectuer très précisément les tâches qui leur ont été attribuées.

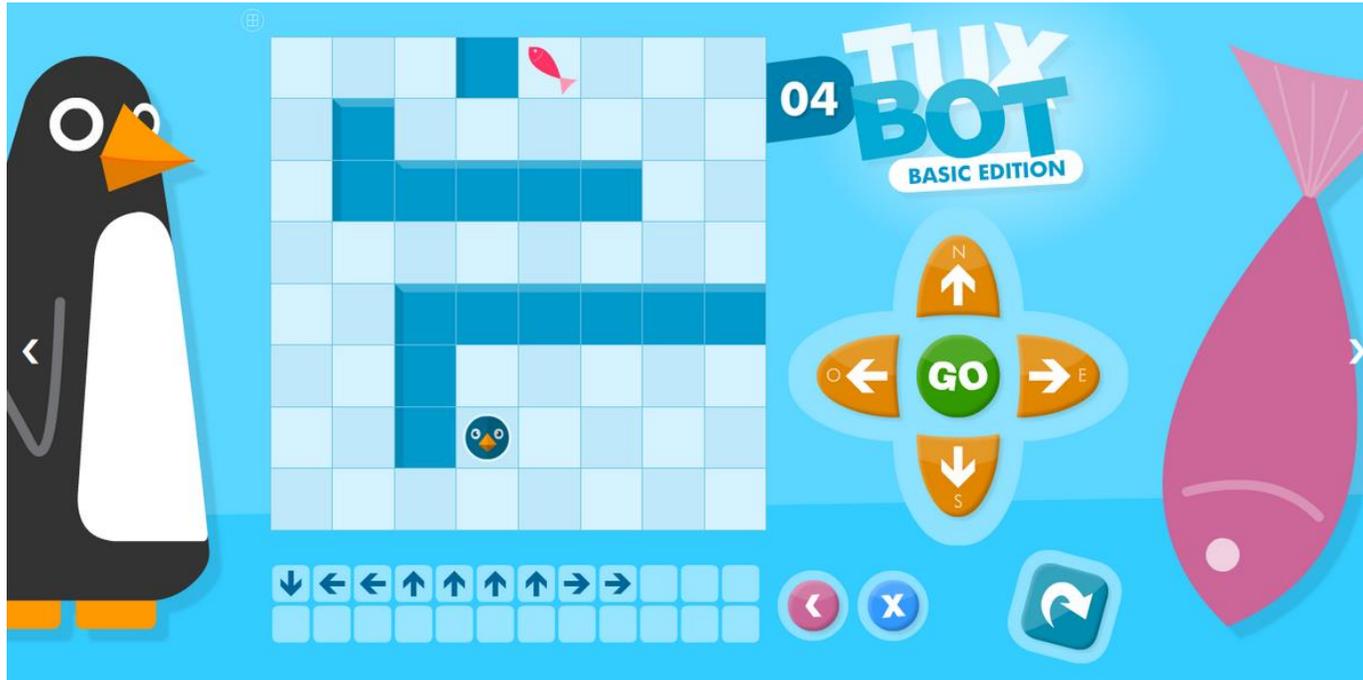
*Marcel le robot*, une ressource clé en mains pour jouer au jeu du robot dans sa classe. Un document qui regroupe le plan du tapis, des idées de programmes et quelques idées d'étape .

#### Programme 1 :

```
↑ ← ↑ ← ↑ ↑ ↑ ←
← ← ↓ ← ↓ ↓ ↓ →
→ → ↑
```



# Tuxbot



Tuxbot est une **application** développée par l'académie de Nantes afin d'initier les élèves de manière ludique à l'algorithmique. Elle est disponible pour **Windows, Android et Linux**.

L'application est complétée par des **cahiers de programmation imprimables**.

Le but du jeu consiste à programmer le parcours d'un manchot afin qu'il ramasse tous les poissons présents sur la grille de jeu.

Le programme ne peut comporter qu'un maximum de 24 instructions et le manchot ne doit pas tomber dans l'eau ni sortir de la grille.

📁 Des cahiers de programmation pour les élèves



<http://numerique53.ac-nantes.fr/ressources/tuxbot/index.php>



# STAMP IT!

Le PIXEL ART à portée de blocs ...

STAMP IT!

NOUVEAU | ENREGISTRER | IMPORTER |

Nombre de blocs utilisés : 0

BIBLIOTHÈQUE

Tampon  
Boucles

début du programme

Le tampon traceur  
dans son quadrillage

Le bouton de  
démarrage  
du programme

Les instructions  
de codage  
disponibles

L'espace où  
les instructions  
choisies  
sont assemblées

▶ Exécuter

Apprendre à programmer en réalisant du pixel art.

L'application en ligne STAMP IT ! est une simulation de machine à tamponner consistant en un automate orienté se déplaçant sur une grille de forme carrée.

L'application propose un tutoriel qui permet une prise en main progressive, un module **d'activité libre** ainsi que des idées de **motifs à programmer**.

