Initiation à la programmation aux cycles 2 et 3

- **<u>Socle commun de connaissances, de compétences et de culture</u> - 2015 :**

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'algorithmique et de la conception des programmes informatiques.
Il les met en œuvre pour créer des applications simples.»

Domaine 2 : Les méthodes et outils pour apprendre

- **<u>Programmes de Mathématiques -2018</u> : Espace et géométrie :**

•**Au cycle 2** : **«(Se) repérer et (se) déplacer en utiliser des repères»**
«programmer les déplacements d'un robot ou ceux d'un personnage sur un écran :
-repères spatiaux;
-relations entre l'espace dans lequel on se déplace et ses représentations.»

- **<u>Cadre de Référence des Compétences Numériques (CRCN)</u> –2020 :**

•3. Création de contenu > 3.4 Programmer :
Niveau 1 : Lire et construire un algorithme qui comprend des instructions simples
Niveau 2 : Réaliser un programme simple

<u>Objectifs :</u>

L'initiation à la programmation permet notamment de travailler les compétences suivantes :

_Se repérer, s'orienter en utilisant des repères

_Adopter une démarche scientifique : utilisation d'un langage spécifique, contrôle, essais-erreurs

_Développer l'abstraction : apprendre à anticiper l'effet de telle ou telle séquence d'instructions avant même de la faire exécuter par une machine ou un programme.

<u>Séance 1</u> : activités débranchées

<u>1) Activité débranchée : « le robot idiot »</u>

Dispositif / Matériel nécessaire :
- 25 plots plats type coupelles [à disposer en formant un carré 5x5 à défaut 16 plots en 4x4] plusieurs grilles
- 2 plots type cône pour marquer le départ et l'arrivée
- 2 ardoises par groupe de 3

<u>Principe</u> :
Dans chaque groupe, 2 ou 3 élèves jouent les « programmeurs » et programment les actions permettant un déplacement pour aller du point de Départ au point d'Arrivée en utilisant un code établi ensemble. Ce programme sera exécuté par le dernier élève du groupe qui jouera le robot.

Phase 1 : Qu'est-ce qu'un robot ? Recueil des conceptions initiales

• Faire dessiner les élèves sur leur représentation de l'objet robot

• Questionner les élèves sur : comment sont les robots, à quoi servent-ils ?, comment et pourquoi ils se mettent en action - en mouvement…

• Établir une définition collégiale : un robot exécute des actions sur des ordres qu'on lui donne. Il fonctionne grâce à un ensemble d'instructions établies qu'on appelle programme.

Phase 2 :découverte

Consigne : *« Vous avez devant vous le terrain de jeu d'un robot. Le robot part de ce cône (montrer le départ) et doit arriver ici (montrer le cône d'arrivée). Décrivez les actions successives que devrait réaliser ce robot pour rejoindre son plot d'arrivée en sachant que le robot se déplace uniquement de plot en plot et ne peut pas se déplacer en diagonale. »*

_Un élève ou le maitre joue le rôle du robot afin de mettre en évidence l'importance de la précision et de la codification du langage utilisé.

Préciser que le robot ne peut avancer ET tourner en même temps. Lorsqu'il tourne, cela signifie qu'il pivot sur un quart de tour à gauche ou à droite.

Phase 3 :Consigne : *« Nous allons maintenant choisir ensemble un langage qui sera compris de tous ».*

_Se mettre d'accord sur les symboles codant les mouvements du robot (et éventuellement les actions)

Voici une proposition de langage qui peut être envisagé :

Phase 4 : appropriation du langage choisi

Les élèves se placent en groupe de 3.

Un élève jouera le rôle du robot, les autres sont les « programmeurs ». Les points D et A sont changés afin d'établir un nouveau parcours.

Consigne : *« Maintenant que nous nous sommes mis d'accord sur le langage à utiliser, proposez sur votre ardoise une programmation que votre robot va effectuer. Attention, votre programme doit être copié sur les 2 ardoises du groupe à l'identique car vous devrez remettre une de vos ardoises à votre robot afin qu'il effectue le parcours indiqué, l'autre vous servira à suivre et vérifier votre programme en temps réel. Vous avez 5 minutes de préparation.»*

Durant cette phase :

- Les élèves jouant le rôle du robot sortent ou sont isolés sans vue sur ce que préparent leurs camarades. La consigne leur est donnée de ne se déplacer que sur le code qui a été établi en commun auparavant et de refuser toute autre instruction. Ils devront suivre strictement ce qui leur a été demandé.

- Les élèves « programmeurs » peuvent investir l'espace. Certains resteront peut-être en retrait pendant que d'autres joueront le rôle du robot pour mieux appréhender le parcours. Il est important de laisser les groupes réfléchir et proposer leur solution.

Au bout des 5 minutes, retour des robots :
Les programmeurs donnent l'ardoise contenant le programme à réaliser. Ils gardent en copie sur la seconde ardoise le même programme. Chaque robot exécute le programme imaginé par les élèves de son groupe en oralisant les commandes du déplacement écrit. Les élèves qui ont réalisé la programmation entourent fur et à mesure les actions effectuées afin de repérer celles qui posent problème.

_Ce temps doit permettre de valider ou non le programme. Les élèves auront ensuite la possibilité de modifier leur programmation et de le retenter autant de fois que nécessaire avec leur robot.

Phase 5 : Mise en commun :
Consigne : *« Finalement, que faut-il faire pour permettre à notre robot d'exécuter la tâche que nous avions imaginée ? »*
Réponses attendues : « bien réfléchir au parcours, imaginer que l'on est à la place du robot, être précis, faire par étape…. »

Institutionnalisation :
« Pour réussir ma programmation, je dois respecter le langage du programme et anticiper toutes les étapes que devra réaliser mon robot. »

Prolongements possibles :
- On place des points de passages obligés ou des bonus (balles ou autres objets) à récupérer sur le chemin.

- Le robot suit un programme caché, les autres élèves doivent le retrouver en fonction des actions effectuées.

_Il ne faut pas hésiter à reconduire cette activité autant de fois que besoin sous sa forme initiale ou sous les prolongements proposés. Une expansion du
du langage peut se faire à chaque reprise de l'activité (premières boucles, « si …alors… » Etc.)

Séance 2 : notion d'algorithme/encodage/décodage

Phase 1 : aventure algorithmique débranchée : présentation – La main à la pâte
L'enseignant explique aux élèves qu'au cours de cette séance et des suivantes, ils vont suivre les aventures d'un héros ou d'une héroïne et qu'ils vont devoir l'aider à résoudre des énigmes pour lui permettre de rentrer chez lui ou chez elle

Situation déclenchante
À son réveil, un héros se retrouve au sommet d'une montagne. Il ne se souvient pas de l'avoir gravie, et la forêt au pied de la montagne ne lui est pas familière. Il ne reconnaît pas le chant des oiseaux… : il n'est pas chez lui. En contrebas, il lui semble apercevoir une clairière : il décide de la rejoindre.

Phase 2 : Exprimer des instructions à l'aide de conditions (par groupes)
L'enseignant distribue aux élèves la **Fiche 12**, et il la projette ensuite au tableau : il s'agit du parcours que va devoir effectuer le héros pour rejoindre la clairière au pied de la montagne. Pour l'aider, les élèves doivent décrire une succession d'instructions que le héros suivra à la lettre pour arriver sain et sauf.
La formulation de ces instructions doit prendre la forme SI… ALORS… Par exemple :
SI le héros rencontre une falaise, ALORS il doit escalader.

- **Par groupes, les élèves travaillent en autonomie**, l'enseignant les encourageant à

exprimer d'abord la liste d'obstacles puis les instructions à donner au héros.
Les élèves complètent la **Fiche 13.**
La mise en commun est l'occasion pour l'enseignant d'introduire un vocabulaire nouveau, utilisé en informatique :
_ Une méthode permettant de résoudre un problème s'appelle un « **algorithme** ». Dans le cas présent, l'algorithme s'exprime en utilisant des « tests » : une « condition » (« *SI le héros rencontre une falaise* ») suivie d'une ou plusieurs instructions à suivre si la condition est vérifiée (« *ALORS il doit escalader* »).
À chaque étape de son périple, le héros vérifie la totalité des conditions du programme (chaque condition est soit vraie, soit fausse) et obéit scrupuleusement à toutes les instructions applicables.
L'enseignant demande aux élèves de comparer cet algorithme avec une autre instruction que l'on aurait pu donner au héros : « Retourne chez toi. » Dans le second cas, on donne un problème complexe à résoudre, sans expliquer comment le faire. Si le héros ne sait pas comment le faire, notre instruction ne va pas l'aider. Un algorithme est construit à partir d'instructions « élémentaires » que le héros sait exécuter.

Phase 3 : Expérimentation : décoder un message encodé (par groupes)
Dans la phase précédente, le héros a pu descendre de la montagne en évitant ses dangers. Arrivé dans une clairière, il découvre une longue inscription, gravée sur un tronc d'arbre.
Expérimentation : décoder un message encodé (par groupes)

L'enseignant projette au tableau la première moitié de la **Fiche 14** : il s'agit de l'inscription gravée sur le tronc d'arbre. Il demande aux élèves ce qu'ils en pensent. Les enfants constatent que le message est une succession de chiffres placés dans des cases. Il y a des espaces entre des groupes de cases, un peu comme entre des mots. Il suffit peut-être, pour comprendre le message, de trouver une correspondance entre ces chiffres et les lettres de notre alphabet ?
L'enseignant introduit alors les termes « encoder » et « décoder ».

_ Les élèves décodent le message :
SUIS LA RIVIERE JUSQU'A LA MER SOUS LES EAUX DORT UN BEAU TRESOR

Phase 4 : Conclusion et traces écrites
La classe synthétise collectivement ce qui a été appris au cours de cette séance :
• Un algorithme est une méthode permettant de résoudre un problème.
• Un test dit quelle action effectuer quand une condition est vérifiée.
• Une condition est une expression qui est soit vraie, soit fausse.
• On peut coder un texte en représentant ses lettres par des nombres choisis à l'avance (par exemple, 1 peut coder « A », 2 peut coder « B »…).

Séance 3 : Se repérer dans l'espace, coder un déplacement
L'élève doit se situer sur un quadrillage, coder un déplacement à l'aide de flèches.

Phase 1 : Expérimentation : inventer un langage pour guider le sous-marin
Situation déclenchante
Après avoir suivi la rivière, le héros arrive à la mer. Sur la plage, il repère un ponton et s'en approche. En regardant au fond de l'eau, il voit le trésor ! Mais celui-ci est hors de portée. En revanche, il voit un petit sous-marin qu'on peut commander à la voix. Il va falloir lui expliquer comment aller chercher le trésor.

L'enseignant projette au tableau la **Fiche 15** : on y voit le fond de l'eau avec un dédale de coraux que le sous-marin doit parcourir afin de parvenir jusqu'au trésor.

Par binômes, les élèves doivent proposer une série d'instructions qui pourront décrire le parcours à suivre. L'enseignant introduit alors le terme **« programme »** pour décrire

l'ensemble d'instructions simples qui peuvent être exécutées par une machine.

Les contraintes sont : le sous-marin ne peut se déplacer que d'une case à la fois ; il ne bouge pas en diagonale. Les binômes peuvent essayer de reproduire leur parcours en bougeant le pion qui leur est fourni, en lui faisant respecter scrupuleusement les instructions.

Phase 2 : Mise en commun

L'enseignant demande à un des groupes de présenter son programme à la classe. Pour vérifier ce que donne l'exécution du programme, les élèves déplacent au tableau la silhouette représentant le sousmarin, en suivant rigoureusement les instructions. Si la méthode est concluante, l'enseignant la reprend au tableau et demande si d'autres binômes ont eu d'autres idées.

On remarque qu'il existe (au moins) deux types de langages pour diriger le sous-marin. On peut lui donner des directions « absolues » (« va vers la surface, va à l'ouest/vers le ponton »…) ou, au contraire, des directions relatives, c'est-à-dire qui dépendent de l'orientation du sous-marin (« tourne vers la droite, avance, tourne vers la gauche, recule »…).

Note : il est préférable de découper l'instruction
« avance d'une case vers la droite » en 2 instructions bien distinctes :
1/ « tourne vers la droite » (sousentendu : en restant sur place),
puis 2/ « avance d'une case ».

Si cette tâche a été exécutée sans difficulté par les élèves, on peut leur demander de programmer le trajet retour du sous-marin, sans oublier l'instruction « attraper » qui signifie « attraper le trésor » afin qu'il ne rentre pas bredouille.

La classe remarque que le sous-marin n'a besoin que d'un langage très simple pour être commandé (en particulier, très peu de mots différents sont nécessaires). L'enseignant explique que les machines, comme les ordinateurs, les robots, etc., peuvent être programmés à l'aide de langages particuliers, appelés « **langages de programmation** », qui sont beaucoup plus simples que les langues naturelles comme le français, l'anglais, etc.
Cette mise en commun permet aussi d'appréhender la notion de « **bug** ». Au cours des présentations des différents programmes, il arrivera certainement une occasion où une instruction manquera ou sera erronée.

Phase 3 : Conclusion et traces écrites

La classe synthétise collectivement ce qui a été appris au cours de cette séance :
• Un programme est une suite d'instructions exprimées dans un langage particulier compréhensible par l'homme et la machine.
• Un bug est une erreur dans un programme. Un tout petit bug peut parfois avoir des conséquences énormes.

Phase 3 : entrainement

Exercices papier crayon

Séance 4 : coder un déplacement, activité débranché et activités logiciel

Lire et exécuter un programme 2 groupes

Trouver l'erreur/ coder un parcours – Jeu du lutin trésor

Etape 1 :

•Positionner le lutin et le coffre

• Donner la grille avec le code du parcours 1D au groupe « ordinateur » avec flèches repositionnables

• Répartition des élèves :

1 élève: lutin/robot

2 élèves: ordinateur/lecteur du code

1 élève : vérificateur pour les déplacements lus et effectués

• Lecture du code par l'ordinateur, déplacement du robot

Etape 2 :

•Positionner les obstacles Montagne, Mare

Expliquer que ces 2 cases sont des obstacles, le robot/lutin ne peut passer sur ces cases

• Donner la grille avec le code du parcours 3D au groupe « ordinateur » avec flèches repositionnables

• Répartition des élèves :

1 élève: lutin/robot

2 élèves: ordinateur/lecteur du code

1 élève : vérificateur pour les déplacements lus et effectués

• Lecture du code par l'ordinateur, déplacement du robot

Etape 3 :

Donner la grille avec le code du parcours 2D au groupe « ordinateur » avec flèches repositionnables

Programme avec erreur

• L'enseignant peut poser les questions suivantes :

Le robot a-t-il bien suivi les consignes données par l'ordinateur ?

L'ordinateur a-t-il bien lu le code ?

• L'enseignant peut aider à l'induction de l'origine de l'erreur :

o D'où vient l'erreur ?

o Que pouvez-vous faire ?

• Les élèves doivent analyser l'erreur, prendre conscience que l'erreur provient du programmeur

• Les élèves peuvent éprouver le besoin d'effectuer à nouveau le parcours pour s'assurer que le programme a bien été lu et que le déplacement effectué est le bon

• Les élèves déduisent que l'erreur provient du programme. Qu'il faut le **corriger**.

• L'enseignant met à disposition des élèves la grille de programme avec le code du parcours 2D + des flèches (moyen format) repositionnables, pour permettre la correction

• Les élèves s'approprient la grille.

• Le programme n'est pas à refaire entièrement : Trouver la partie du programme à corriger

• Corriger le programme en modifiant le code

• Vérifier et valider ce nouveau programme: un élève/robot refaisant le parcours

Etape 4 :

idem avec le poisson qui doit arriver à la rivière

Etape 5 : Elaboration de programmes – Feuille A4

• A partir de la même grille d'évolution au sol (emplacements du lutin, de la montagne, de la mare et du coffre inchangés) : construire un parcours par binôme et le faire valider par un pair. Trouver plusieurs possibilités

• Distribuer le matériel de codage (bande ou grille) individuellement ou par binôme

• Les élèves ont pour consigne de programmer un parcours, en évitant les obstacles, afin que le lutin puisse atteindre le coffre.

Ils codent ce parcours sur la bande ou sur la grille de programmation

• Les élèves testent leur code

• Auto-validation

Lire et exécuter un programme 2 groupes

Tuxbot – déplacement relatif CE2 – absolu CE1

Objectifs :

•Se repérer sur un plan
•Programmer le déplacement d'un personnage sur un écran
•Deux types de déplacements sont possibles : absolus ou relatifs
•Créer un algorithme afin d'aider le pingouin à manger des poissons
•Possibilité d'insérer des boucles

Lire et exécuter un programme 1 groupe

Lightbot- Clic ma classe

Objectifs :

•Se repérer sur un plan
•Programmer le déplacement d'un personnage sur un écran
•Deux types de déplacements sont possibles : absolus ou relatifs
•Créer un algorithme afin d'aider le pingouin à manger des poissons
•Possibilité d'insérer des boucles
Découvrir les bases de la programmation

Séance 5 à 8 : coder un déplacement, activités logiciel et robot

Ozobot 4/5Elèves

Compétences travaillées Domaines du socle

Pratiquer des démarches scientifiques
Se situer dans l'espace
- construire des repères spatiaux
Comprendre et s'exprimer à l'oral
- écouter pour comprendre des messages oraux ou des textes lus par un adulte
- participer à des échanges dans des situations diversifiées

Attendus de fin de cycle 2 :

les objets techniques :
- comprendre la fonction et le fonctionnement d'objets fabriqués

se situer dans l'espace :
- lire des plans, se repérer sur une carte

comprendre et s'exprimer à l'oral :
- conserver une attention soutenue lors de situations d'écoute ou d'interactions et manifester, si besoin et à bon escient, son incompréhension
- participer avec pertinence à un échange (questionner, répondre à une interpellation, exprimer un accord ou un désaccord, apporter un complément…)

Souris 4 Elèves

Objectifs :

Découverte de la souris, de sa programmation et de ses déplacements.

Objectifs pour l'enseignant :
• Faire découvrir les fonctions de la souris et les boutons de commande du robot.
• Réinvestir le code universel (flèches) découvert dans les défis débranchés.
• Introduire le terme « algorithme ».
• Faire transposer un tracé sur feuille en un programme sur souris
• Faire transposer les commandes saisies sur le robot en un code écrit.
• Expliquer qu'il peut y avoir plusieurs façons de programmer pour répondre à une même problématique

Objectifs des élèves :
• Travailler l'essai/erreur afin de maîtriser les commandes de la souris
•Travailler l'essai/erreur afin de réaliser les différentes missions proposées.
• Créer un programme sur le robot pour répondre à la consigne.
• Résoudre des problèmes mathématiques

Botley 4 Elèves

Objectifs :

Découverte de Botley, de sa programmation et de ses déplacements.

Objectifs pour l'enseignant :
• Faire découvrir les fonctions de Botley et les boutons de commande du robot.
• Réinvestir le code universel (flèches) découvert dans les défis débranchés.
• Introduire le terme « algorithme ».
• Faire transposer un tracé sur feuille en un programme sur Botley.
• Faire transposer les commandes saisies sur le robot en un code écrit.
• Expliquer qu'il peut y avoir plusieurs façons de programmer pour répondre à une même problématique

Objectifs des élèves :
• Travailler l'essai/erreur afin de maîtriser les commandes de Botley
•Travailler l'essai/erreur afin de réaliser les différentes missions proposées.
• Créer un programme sur le robot pour répondre à la consigne.
• Transposer les déplacements de Botley en une suite d'actions écrites

BlueBot 4 Elèves x 2

Objectifs :

Découverte de Bluebot, de sa programmation et de ses déplacements.
Course aux dés et récolte de pièces

Objectifs pour l'enseignant :

• Faire découvrir les fonctions de Bluebot et les boutons de commande du robot.
• Réinvestir le code universel (flèches) découvert dans les défis débranchés.
• Introduire le terme « algorithme ».
• Faire transposer un tracé sur feuille en un programme sur Bluebot.
• Faire transposer les commandes saisies sur le robot en un code écrit.
• Expliquer qu'il peut y avoir plusieurs façons de programmer pour répondre à une même problématique

Objectifs des élèves :

• Travailler l'essai/erreur afin de maîtriser les commandes de Bluebot
• Créer un programme directement sur le robot pour répondre à la consigne.
•Travailler l'essai/erreur afin de réaliser les différentes missions proposées.
• Créer un programme sur le robot pour répondre à la consigne.
• Transposer les déplacements de Bluebot en une suite d'actions écrites (programme

Découverte d'Ozobot

1. Voici un robot, selon vous, que peut-il faire ?

…….

…….

2. **Manipuler Ozobot en utilisant les parcours 1 et 1 bis.**

Que fait Ozobot ? …….

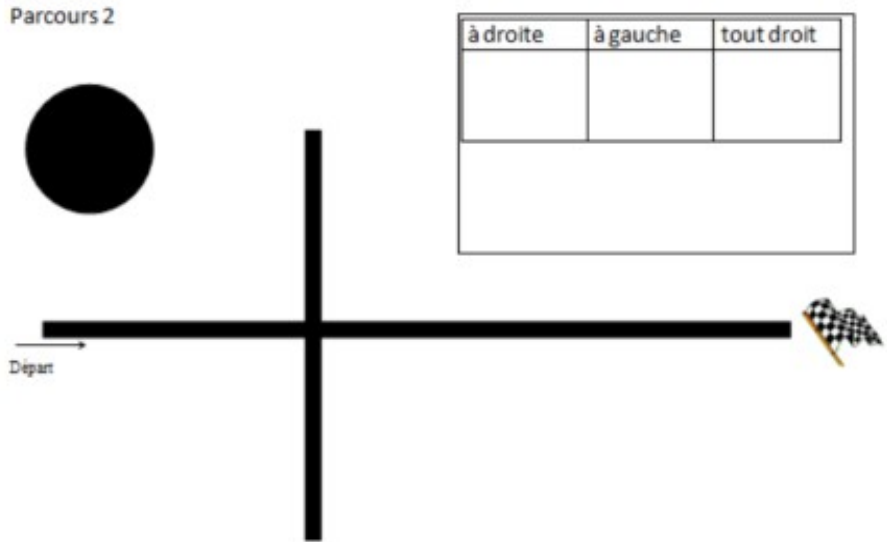
Qu'est ce qui permet à Ozobot de suivre les lignes et de changer de couleur ?

…….

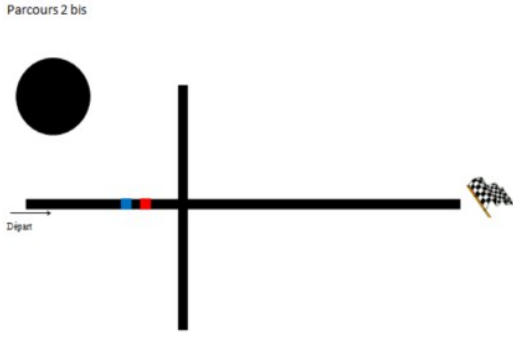
…….

3. **Objectif :** faire aller Ozobot du point de départ au point d'arrivée.

Note dans le tableau le nombre de fois où le robot prend une direction « tout droit », « à droite » ou « à gauche ».



Puis test le parcours 2 bis



Comment peut-on contraindre le robot à réaliser les instructions que l'on souhaite ?

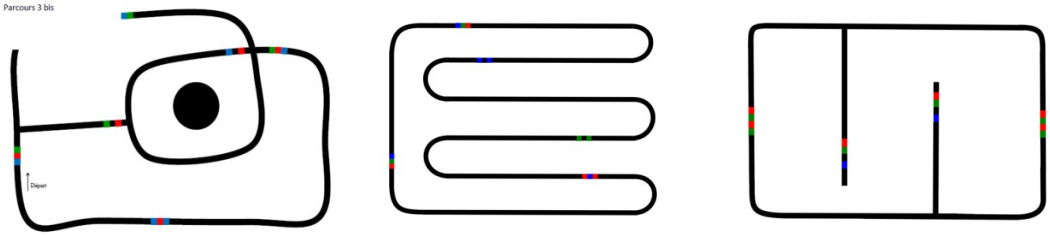
…….

…….

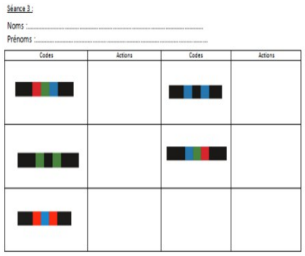
DECOUVERTE DES POSSIBILITES DES CODES COULEURS

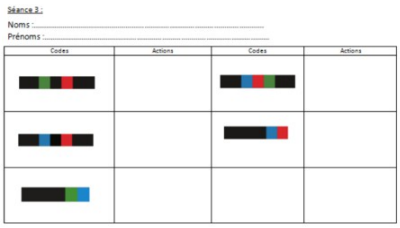
<u>Objectif :</u> Découvrir le langage qu'utilise ce robot.

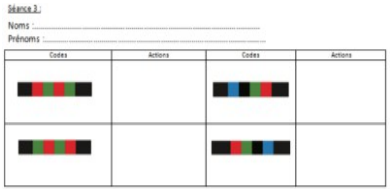
1. **Teste, grâce au robot, les parcours qui te sont confié.**



Complète les tableaux :



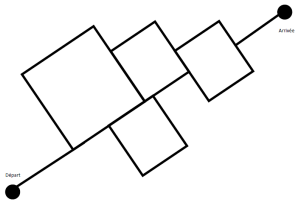




ENTRAINEMENT A L'UTILISATION DES CODES COULEURS | **3**

<u>Objectif :</u> S'entraîner à coder et décoder un parcours d'ozobot à l'aide des cartes déplacement.

Munis-toi du Labyrinthe :



1. A partir du parcours proposé, tu dois déterminer et lister en français les instructions qu'Ozobot va devoir suivre tout au long du parcours. Puis code chaque instruction à l'aide des cartes de déplacement et disposent ces cartes sur le parcours.

Ecris les instructions :

…….

…….

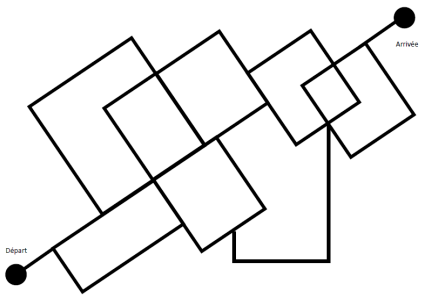
…….

…….

…….

…….

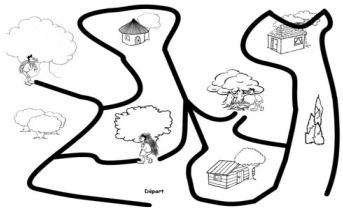
2. **<u>Trouve l'algorithme qui permet au robot de rejoindre l'arrivée le plus rapidement/lentement possible. Utilise le</u> code couleurs**



ENTRAINEMENT A L'UTILISATION DES CODES COULEURS

<u>Objectif :</u> Coder les déplacements de l'Ozoloup sur un parcours de plus en plus complexe en développant les stratégies les plus efficaces pour éviter les obstacles.

<u>Etape 1</u> : À partir du parcours proposé (annexe-9-parcours-ozoloup), tu dois définir :



- Le point d'arrivée,
- Le sens du parcours
- L'ordre de passage devant chacune des maisons,
- Les obstacles à éviter (les êtres humains)

en utilisant les codes :

- tourne à droite , 
- tourne à gauche, 
- va tout droit. 

Puis vérifie tes hypothèses en testant le parcours avec Ozoloup.

<u>Etape 2 :</u>

<u>Les déplacements d'Ozoloup :</u>

- la pause de 3 sec devant les maisons (pour que le loup souffle dessus), 
- l'accélération en sortie de maison pour courir après le cochon, 
- la tornade sur la 3eme maison, car il n'arrive pas à la détruire 
- Il passe par la cheminée, se brûle les fesses et part au triple galop 

Découverte de BlueBot

1

Expérimentation :

Tu vas découvrir ou redécouvrir Bluebot , manipule-le (allumage, son, les boutons de déplacement : tourne à droite, tourne à gauche, avancer, reculer, pause, le bouton d'effacement.).
_ Complète le tableau suivant

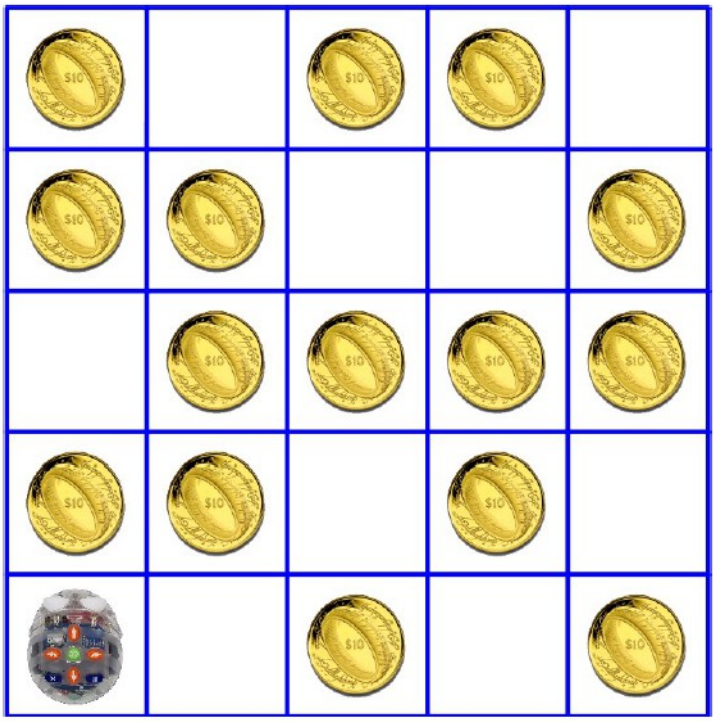
Quand j'appuie sur 	Blue Bot tourne sur lui-même à droite. Blue Bot se déplace à droite. Blue Bot tourne sur lui même à gauche.
Quand j'appuie sur 	Blue Bot se déplace à gauche. Blue Bot se déplace à droite. Blue Bot tourne sur lui-même à gauche.
Quand j'appuie sur 	Blue Bot tourne. Blue Bot recule. Blue Bot avance.
Quand j'appuie sur 	Blue Bot s'arrête. Blue Bot recule. Blue Bot se lève.
Quand j'appuie sur 	Blue Bot fait ce qu'il veut. Blue Bot réalise un programme. Blue Bot fait une pause.
Quand j'appuie sur 	Blue Bot avance. Blue Bot recule. Blue Bot fait une pause.
Quand j'appuie sur 	Blue Bot tourne sur lui-même. Blue Bot joue de la musique. Blue Bot efface son programme.

Programmer BlueBot

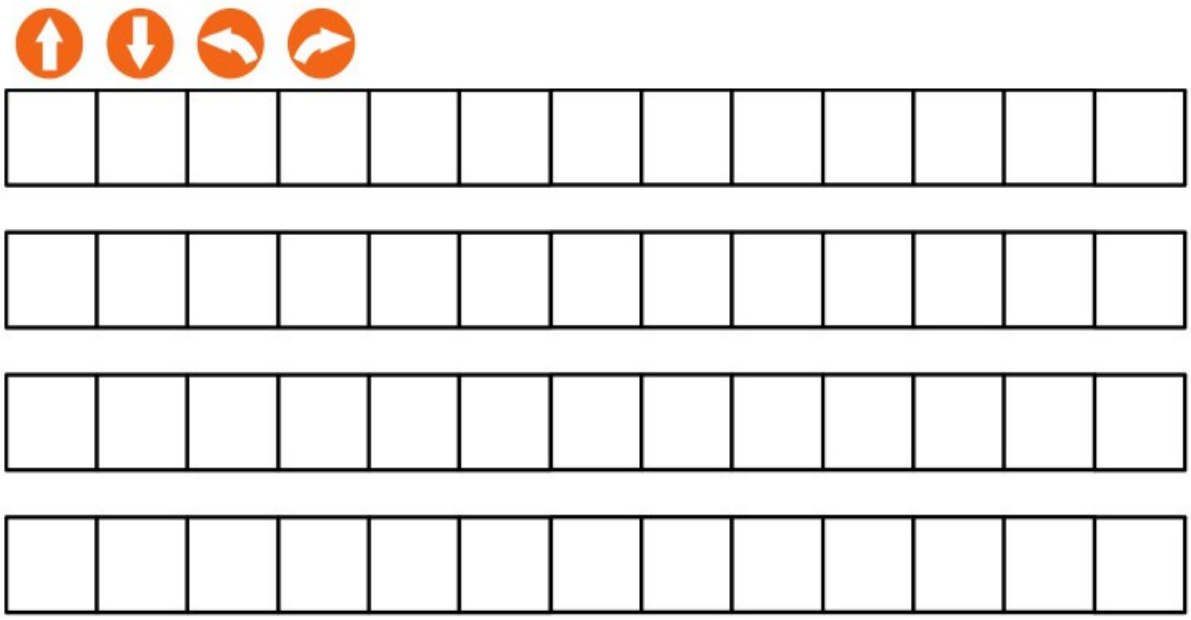
Mission 1 :

Mission 1 : Bluebot doit ramasser 10 pièces, trace le parcours qu'il devra faire puis saisis le programme dans Bluebot. Il s'arrêtera dans un angle.

_ Placer les pièces sur la grille ainsi que la Bluebot.



Complète la grille de programmation :

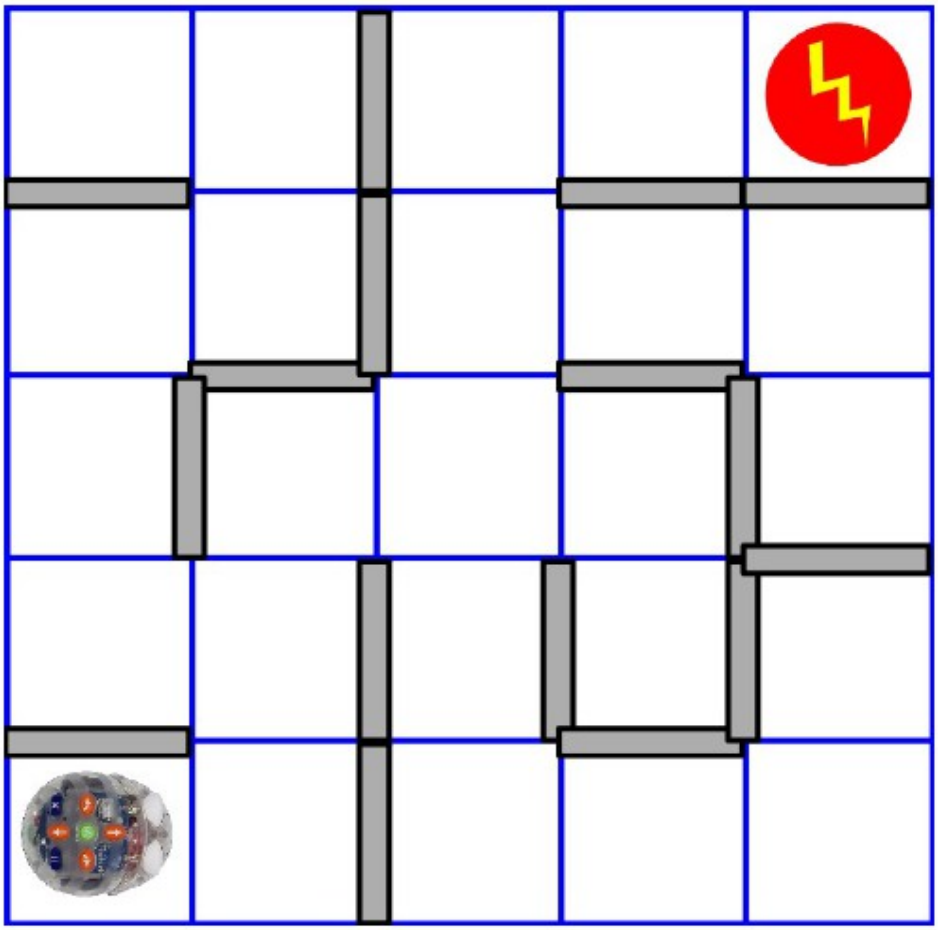


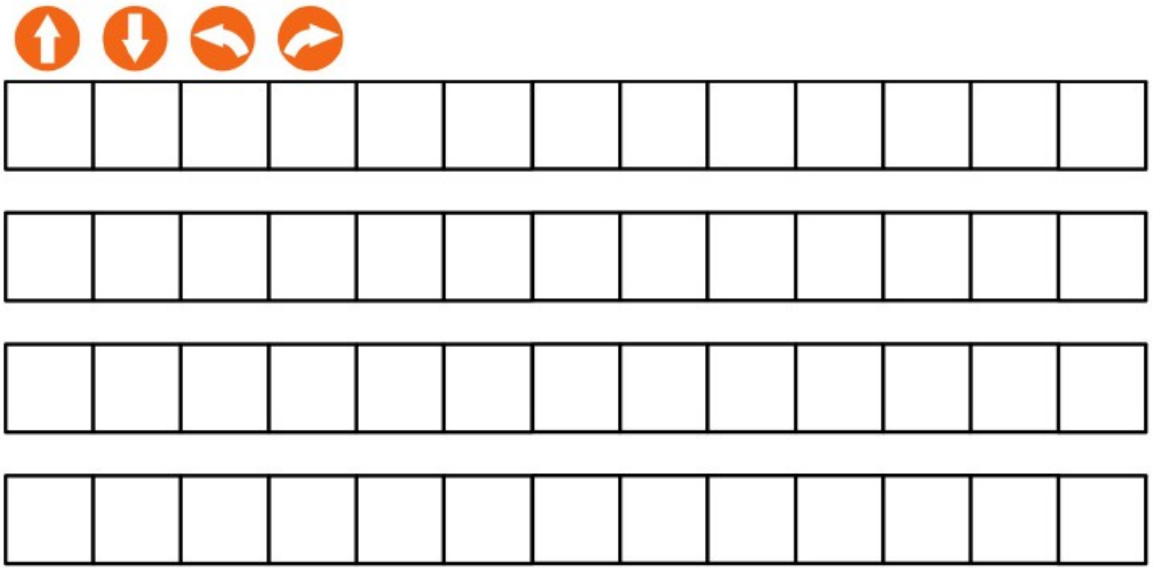
Programmer BlueBot

Mission 2 :

<u>Mission 2</u> : Bluebot doit aller sur sa base de rechargement, écris le programme pour qu'il puisse trouver son chemin dans le labyrinthe.

Placer les obstacles sur la grille, la Bluebot et la zone de rechargement
Préparer le programme sur feuille avant de l'entrer dans la Bluebot.





Quelle distance a-t-il parcourue ? .....

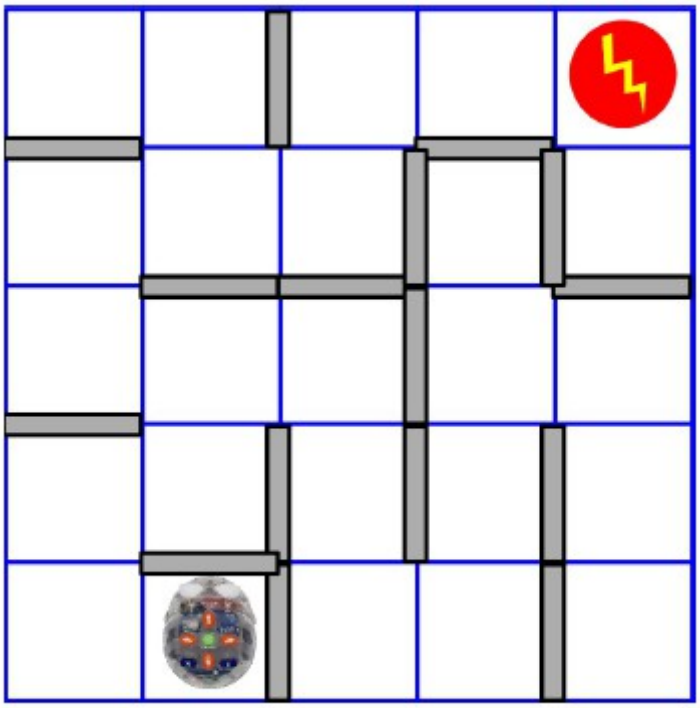
Pour t'aider, utiliser le gabari ainsi que la grande règle de la classe.

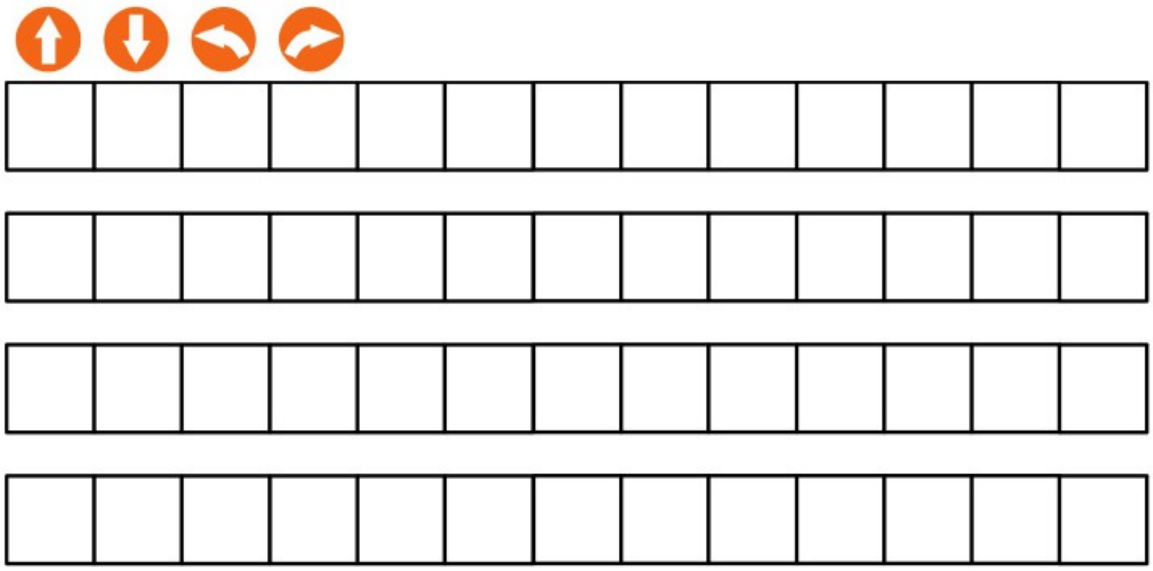
Programmer BlueBot

Mission 3 :

<u>Mission 2</u> : Bluebot doit aller sur sa base de rechargement, écris le programme pour qu'il puisse trouver son chemin dans le labyrinthe.

Placer les obstacles sur la grille, la Bluebot et la zone de rechargement.
Préparer le programme sur feuille avant de l'entrer dans la Bluebot.





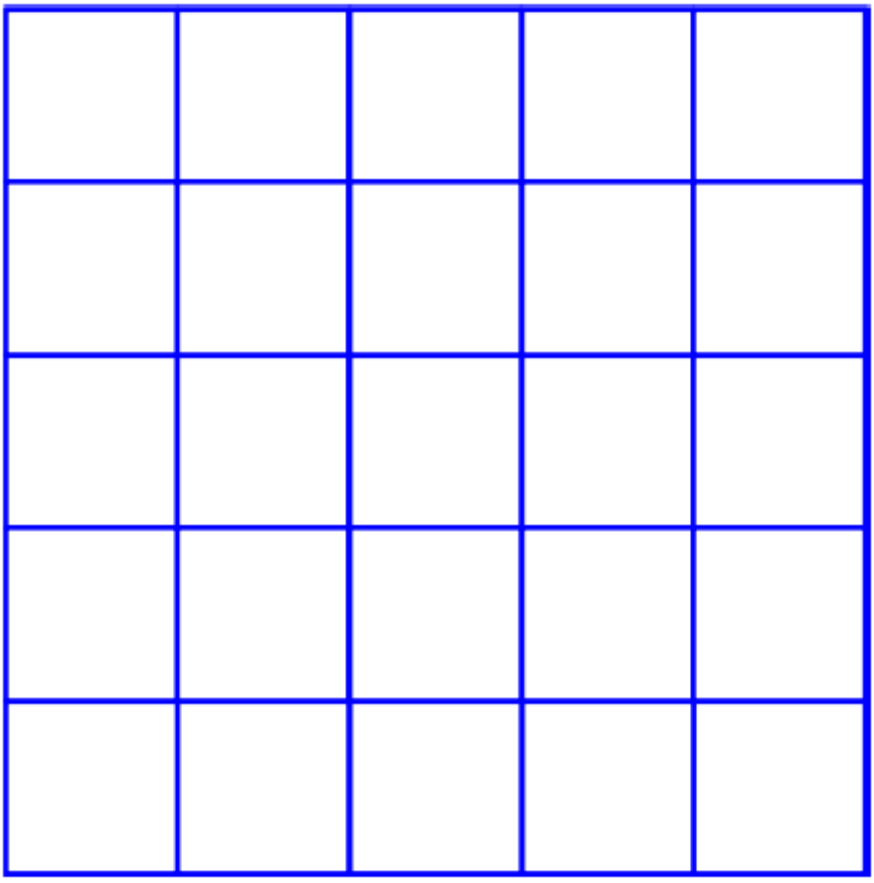
Quelle distance a-t-il parcourue ? .....

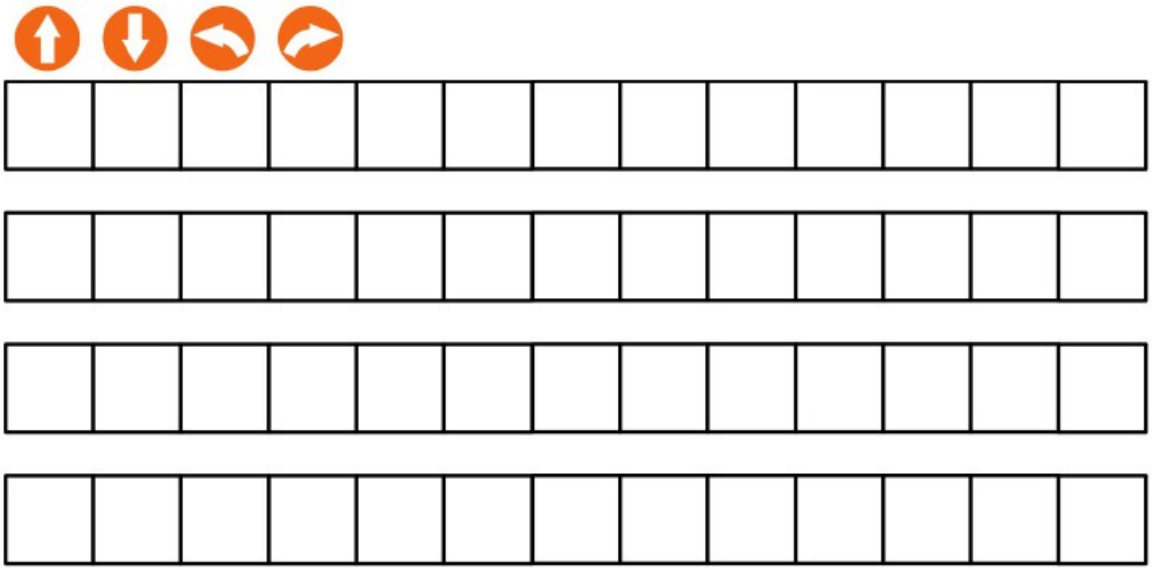
Pour t'aider, utiliser le gabari ainsi que la grande règle de la classe.

Mission 4:

• Vous devez écrire le programme qui permettra de rejoindre la base de rechargement. Avec un autre groupe, échangez vos places et retrouvez le programme pour sortir du labyrinthe, l'autre groupe vérifiera votre programme et vous vérifierez le leur.

Votre groupe construit son propre labyrinthe, place le point de départ et la base de rechargement pour l'arrivée. Ecrivez le programme, puis vérifier le programme de l'autre groupe.



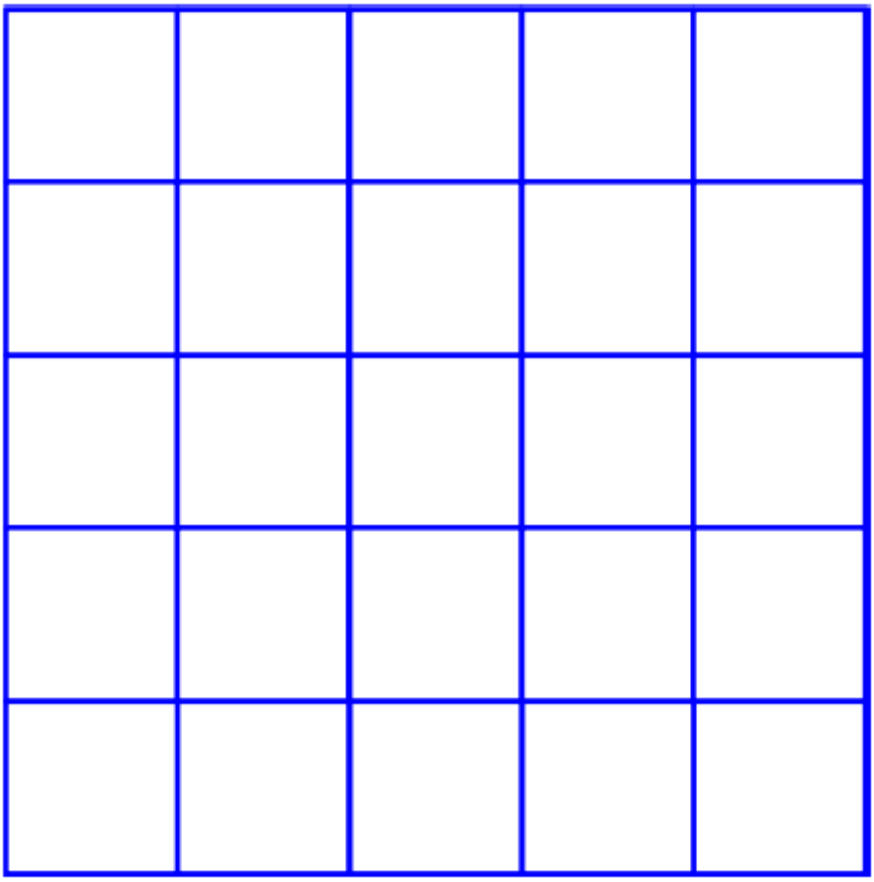


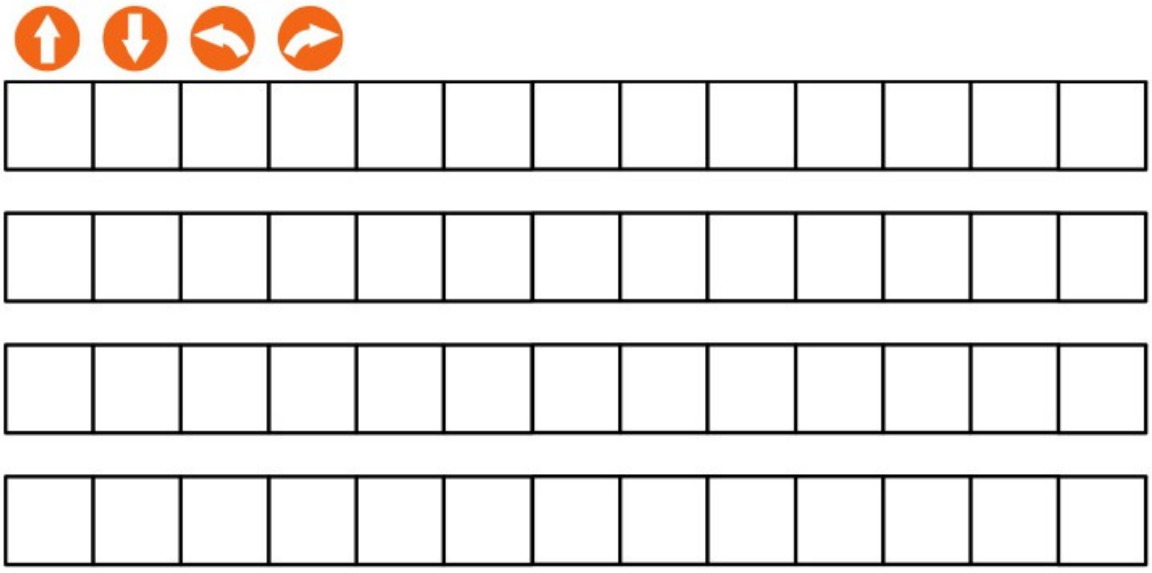
Programmer BlueBot **Mission 5:**

Mission 5:

• Vous devez écrire le programme qui permettra de rejoindre la base de rechargement. Attention BluBot n'a presque plus de batterie, il ne peut donc parcourir que 75 cm.
Vous n'avez pas le droit de mesurer, vous devez utiliser uniquement le gabarit.

Votre groupe construit son propre labyrinthe, place le point de départ et la base de rechargement pour l'arrivée. Attention la base de rechargement doit se trouver à 75 cm .
Ecrivez le programme, un autre groupe le vérifiera.





Programmer BlueBot **Mission 6:**

<u>Mission 6</u>:

Bluebot part en vacances en Australie. Il a préparé son itinéraire mais il n'est pas sûr de lui, peux tu vérifier avant son départ.

Arrivera-t-il au bon endroit ? .....

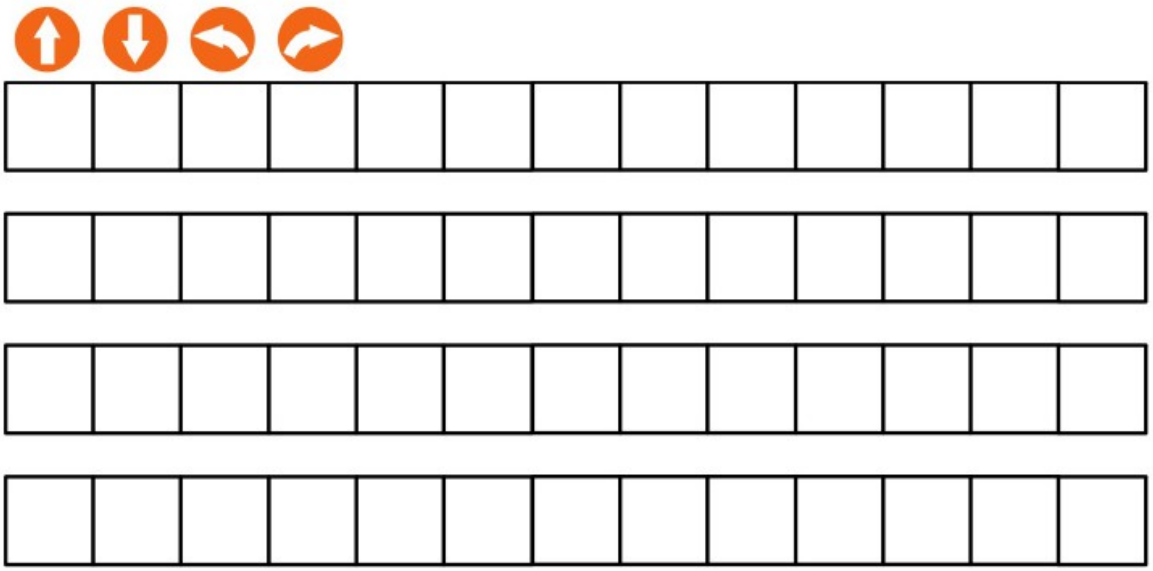
Où est-il arrivé ? .....

Corrige son itinéraire :

Programmer BlueBot **Mission 7**

Mission 7:

Bluebot est en vacances en Alaska, il doit rentrer chez lui en Nouvelle-Zélande. Ses batteries sont déchargées, il peut parcourir au maximum que 120 cm. Aide-le à rentrer chez lui.

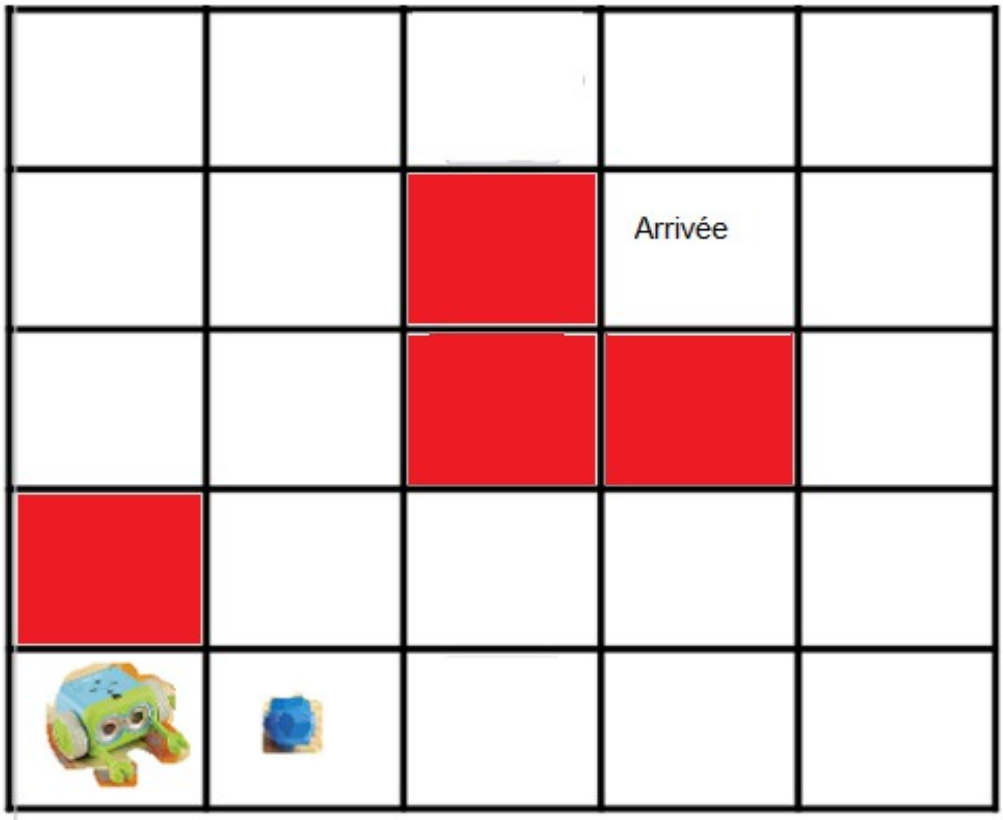


120 cm = ... cases

Programmer Botley

Mission 1

<u>Mission 1</u> :
Botley doit emmener sa balle sur la ligne d'arrivée sans passer par les cases rouges.
Attention, à l'arrivée, il doit se retrouver dans sa position de départ !

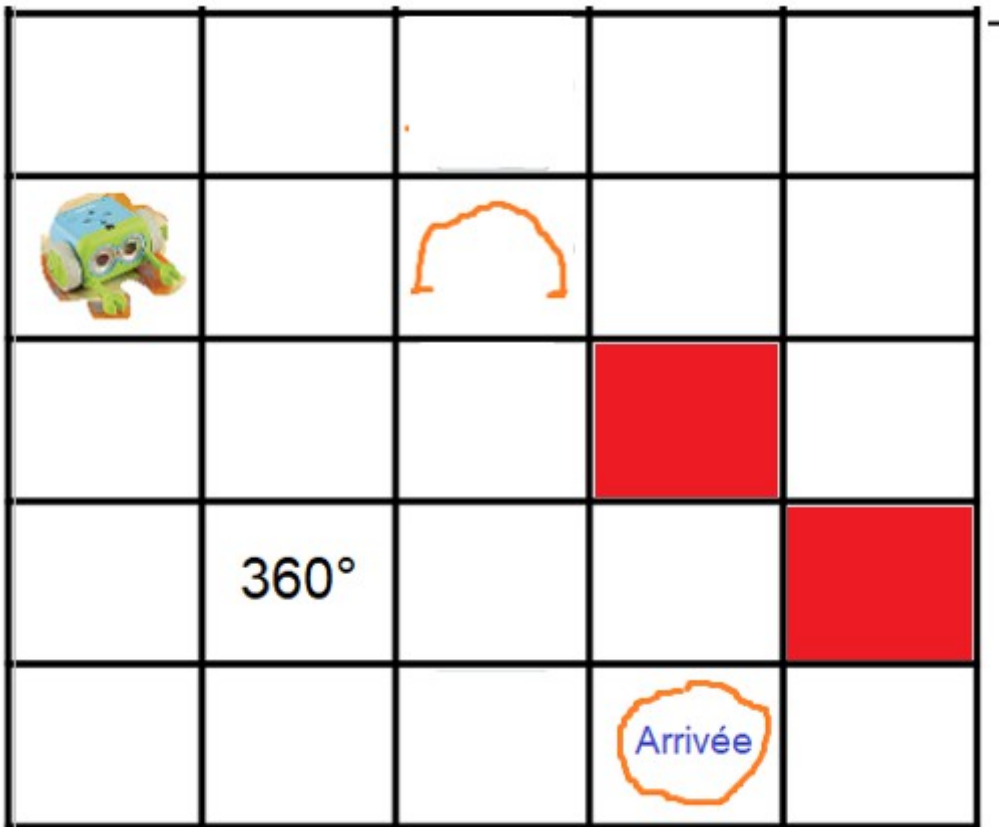


Programmer Botley **Mission 2**

Mission 2 :

Botley doit passer sous un pont, effectuer un 360° puis aller chercher son trophé à l'arrivée.

Réalise le programme, il doit avoir le moins d'actions possibles.



Programmer Botley

Mission 3

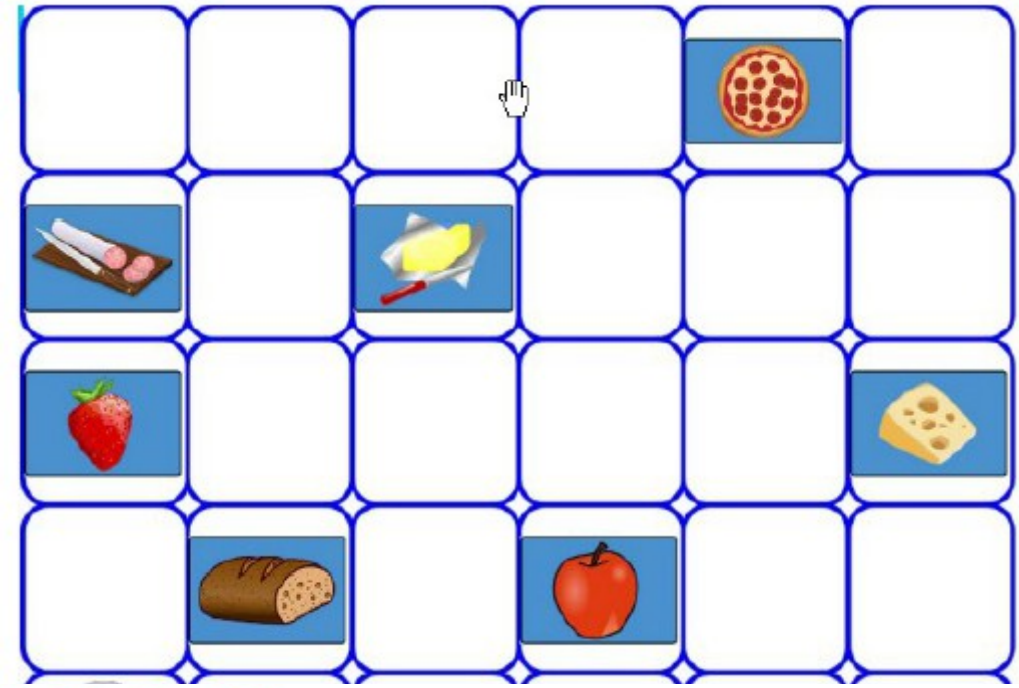
Mission 3 :

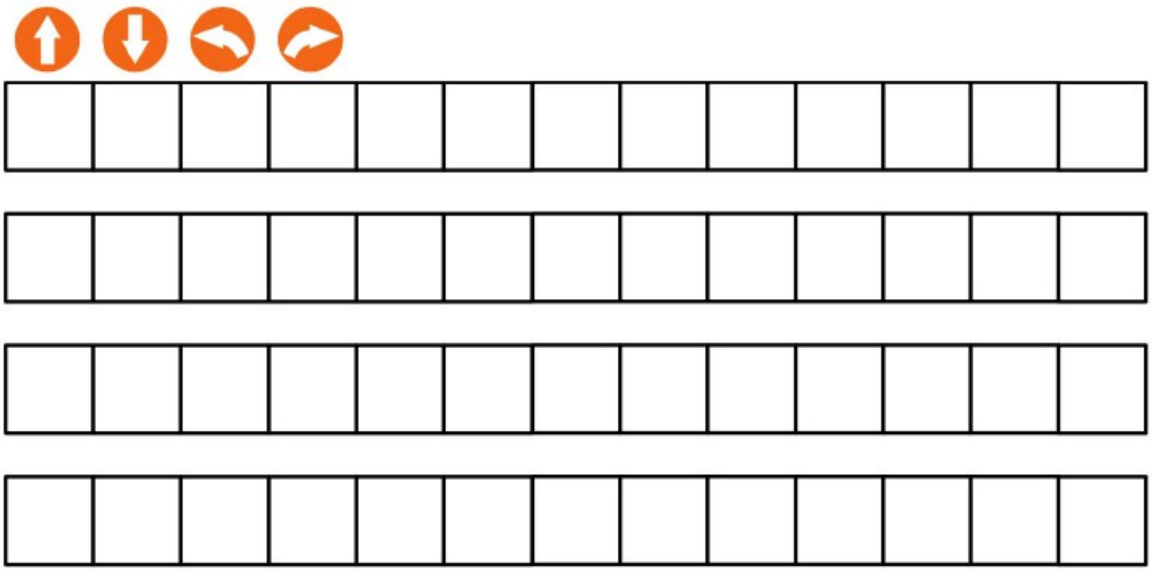
Botley veut se faire un sandwich. Il doit uniquement passer sur les cases où se trouvent les ingrédients dont il a besoin. Il faut faire le moins de déplacements possibles.

Pour son sandwich, Botley a besoin :

- de pain
- de saucisson
- de beurre
- de fromage

Programme Botley pour qu'il récupère tous ces ingrédients.





Programmer Botley

Mission 4

Mission 4 :

Invente une mission pour Botley :

…….

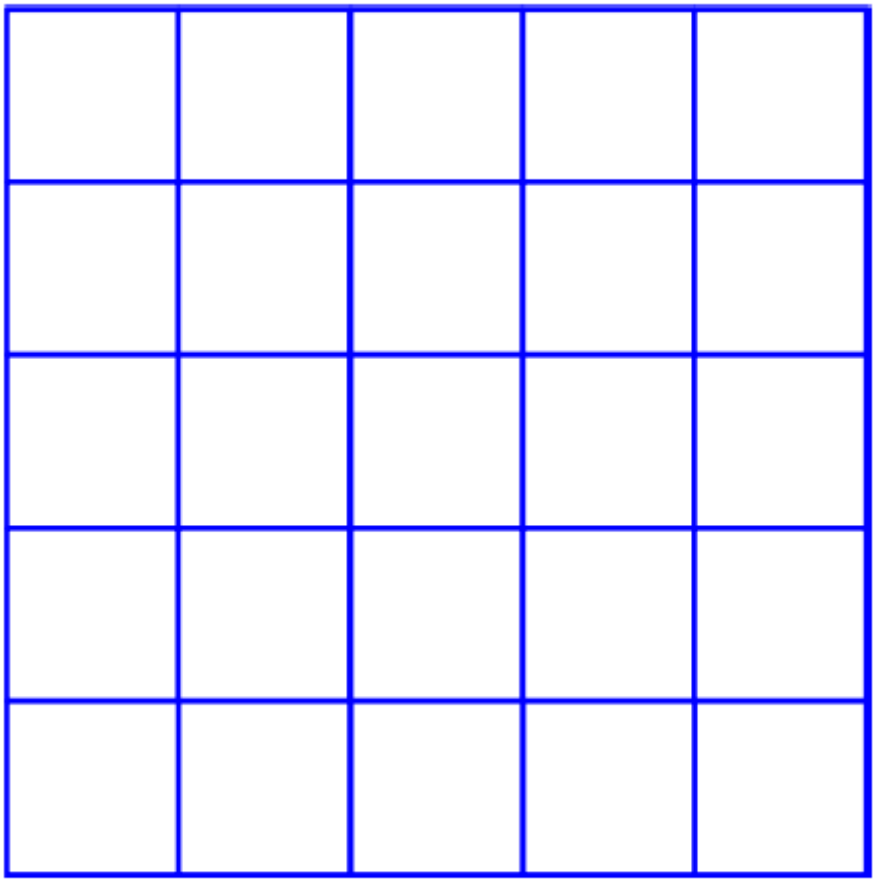
…….

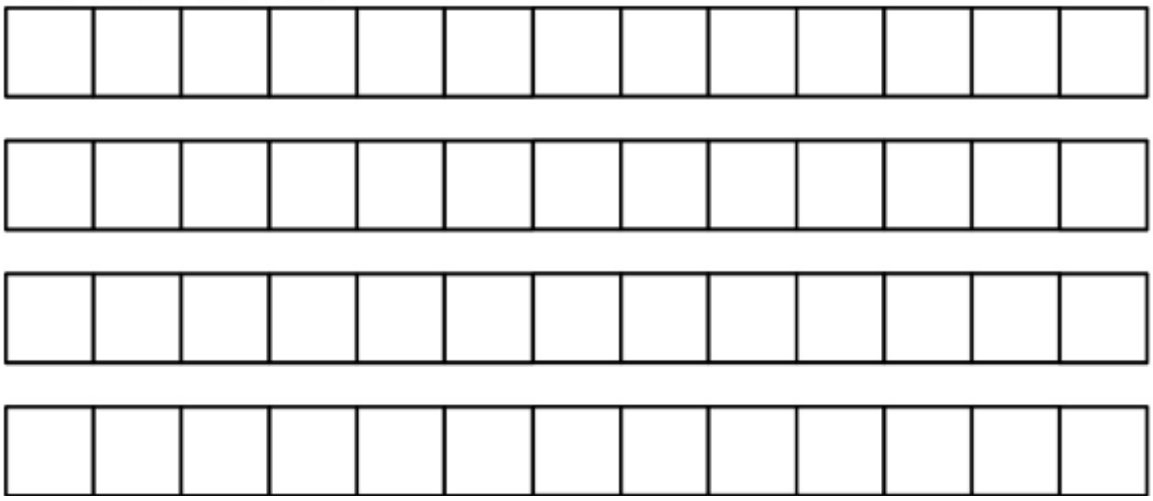
…….

…….

…….

…….





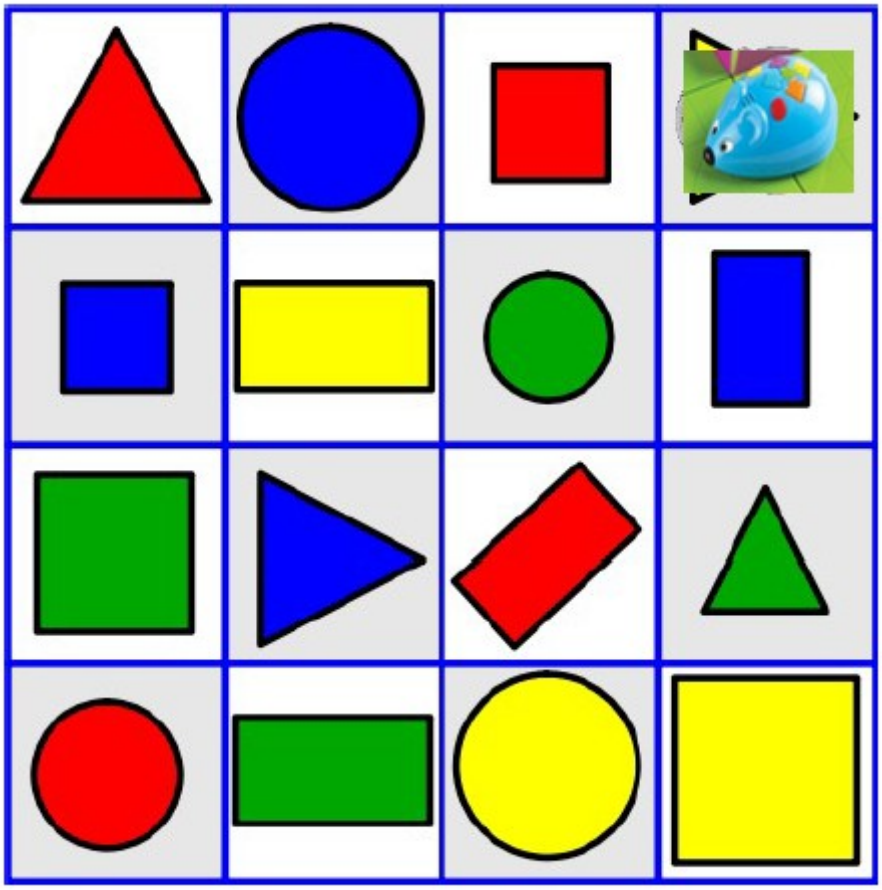
Programmer la souris

Mission 1

Mission 1 :

La souris doit partir du triangle jaune et doit passer sur tous les triangles de la grille.

Placer les formes géométriques sur la grille ainsi que la souris.

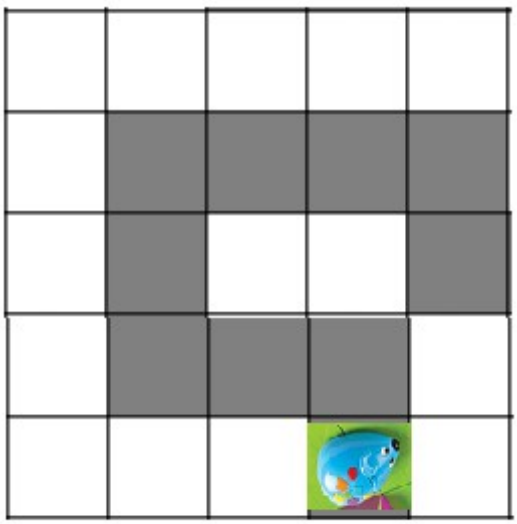


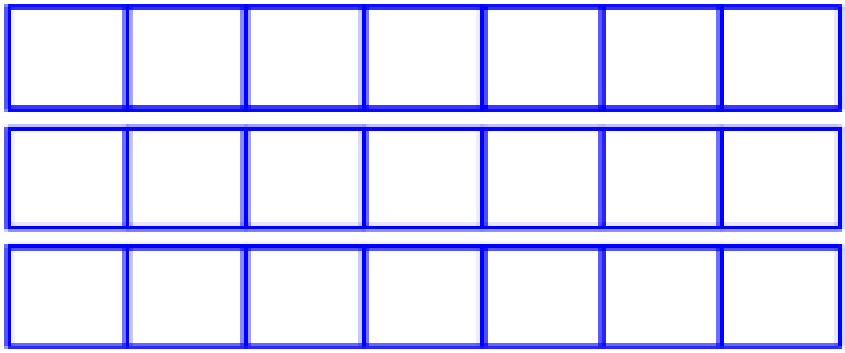
Programmer la souris

Mission 2

Mission 2 :

Code le chemin que doit suivre la souris :





Attention !

Quand elle avance sur la case 1, elle dépose 1 morceau de fromage ; quand elle avance sur la case 2, elle pose 2 morceaux de fromage ...etc...

Combien a-t-elle déposé de morceaux de fromage ?

…….

…….

…….

…….

Programmer la souris

Mission 3

Mission 3 :

La souris se rend en ville en prenant le train. Elle doit :

• déposer un courrier à la Poste,

• récupérer ses médicaments,

• acheter ses livres à la librairie,

Elle en profitera aussi pour manger ses épices préférées en visitant le musée.

Programme la souris pour qu'elle puisse faire tout ce qu'elle a prévu en utilisant le moins de déplacements possibles.

Programmer la souris

Mission 4

Mission 4 :

La petite souris n'a pas mangé depuis deux jours, elle n'a plus beaucoup d'énergie. Elle doit absolument manger : elle a besoin de 4 morceaux de fromage. Aide-la !

Comme elle a peu d'énergie, elle peut effectuer uniquement 12 déplacements. Les morceaux de fromage sont cachés sous les cases égales à 13.

7+4	9+4	9+8	7+3	4+9
3+6	4+8		6+6	3+9
8+5	5+5	9+3	5+7	5+8
4+3	8+7	6+7	9+6	3+6

Programmer la souris

Mission 5

Mission 6 :

La petite souris doit dévorer 4 morceaux de fromage. Attention, tu n'as pas le droit d'utiliser la fléche de droite (la rose), ni de passer sur les cases rouges.