Coder des déplacements

Cette séquence, qui s’adresse aux élèves de cycle 2 ou 3 n’ayant jamais abordé la programmation en classe, propose des activités de programmation dites « débranchées », au sens où elle ne nécessite pas de matériel informatique (ordinateurs/tablettes/robots…).

Pour les classes qui souhaiteraient aller plus loin, si elles sont équipées d’ordinateurs ou de tablettes, des activités de prolongement « branchées » sont proposées, si elles ne le sont pas, des fiches modifiables sont à votre disposition.

**Éléments de programme (Espace et géométrie) :**

Dès le CE1, les élèves peuvent coder des déplacements à l’aide d’un logiciel de programmation adapté, ce qui les amènera au CE2 à la compréhension, et la production d’algorithmes simples.

Les activités spatiales et géométriques [...] constituent des moments privilégiés pour une première initiation à la programmation notamment à travers la programmation de déplacements ou de construction de figures.

**Socle commun :**

« [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.»

**Cadre de référence des compétences numériques :**

Compétence 3.4 : programmer

Niveau 1 : Lire et construire un algorithme qui comprend des instructions simples.

Niveau 2 : Réaliser un programme simple.

|  |  |  |  |
| --- | --- | --- | --- |
| **Séance/**  **durée** | **Objectif(s)** | **Déroulement** | **Matériel** |
| 1  Déplacer un objet sur un quadrillage  40 min | Approcher la programmation de déplacements selon une logique allocentrée (ou déplacements absolus) | - L’enseignant projette (ou reproduit au tableau) la [fiche 2](#Fiche2) et demande aux élèves répartis en binômes de donner des ordres au robot pour qu’il aille jusqu’à sa base de rechargement  - Insister sur le fait que le robot ne comprend qu’un langage très simple. L’instruction ne peut pas être par exemple « Rentre à ta base ». Ne pas exiger de mots, textes ou symboles en particulier.  - Après quelques minutes de recherches, un binôme propose son travail.  - Le programme est effectué instruction par instruction au tableau par un autre élève afin de valider le programme.  - En fonction des travaux réalisés, montrer les travaux de quelques groupes. Faire remarquer qu’il y a plusieurs chemins possibles et que chaque groupe a probablement utilisé un codage différent.  - Trouver un consensus sur un code simple et efficace (privilégier les flèches haut, bas, droite, gauche) que tous les élèves utiliseront à partie de la séance suivante. Un exemple est fait au tableau avec les cartes de la [fiche 1](#Fiche1).  - D’autres situations peuvent être représentées en plaçant simplement le point de départ du robot et la base à différents endroits sur le quadrillage.  - La classe synthétise collectivement ce qui a été appris au cours de cette séance.  **Exemple de trace écrite à adapter en fonction de l’âge des élèves :**  Un robot ne comprend que des instructions simples. Elles peuvent être écrites ou codées avec des symboles. La liste de tous les symboles est appelée un langage de programmation. Il existe de nombreux langages de programmation. Une suite d’instructions est un programme. Dans un programme, les instructions s’effectuent les unes à la suite des autres. | 3 ou 4 copies des cartes de la [fiche 1](#Fiche1) (flèches simples uniquement pour les séances 1 et 2)  Robot et base découpés ([fiche 18](#Fiche18))  [Fiche 2](#Fiche2) (quadrillage) à projeter ou photocopier |
| 2  Programmer des déplacements  en logique allocentrée (ou déplacements absolus)  30 min | Savoir programmer des déplacements selon une logique allocentrée. | - L’enseignant rappelle ce qui a été appris lors de la séance précédente puis distribue la [fiche 3](#Fiche3). Il demande aux élèves d’écrire individuellement le programme (ou algorithme) avec le langage choisi lors de la séance précédente.  - Après quelques minutes de travail, un élève montre son programme. Il demande ensuite si quelqu’un a réussi à écrire un programme plus court. Si oui, alors on le valide ou l’invalide collectivement.  - L’enseignant distribue la [fiche 4](#Fiche4). Les étapes décrites pour la fiche 3 sont ensuite répétées.  - La classe synthétise collectivement ce qui a été appris au cours de cette séance.  **Exemple de trace écrite à adapter en fonction de l’âge des élèves :**  Un algorithme est une méthode permettant de résoudre un problème. Plusieurs algorithmes différents permettent d’arriver à un même résultat. Certains sont plus efficaces que d’autres. En programmation, on essaye toujours d’écrire le programme le plus court possible.  Des séances de prolongement « branchées » peuvent être effectuées avec le logiciel ou l’application tablette « Tuxbot », développée par les conseillers du département de la Mayenne, à télécharger sur :  <http://appli-etna.ac-nantes.fr:8080/ia53/tice/ressources/tuxbot/index.php>  Pour s’entrainer en logique allocentrée, sélectionner dans les paramètres le mode « Basic édition ». Si nécessaire, le logiciel propose également un mode entraînement dans lequel les déplacements sont effectués en direct plutôt que par anticipation.  Pour les classes non équipées, d’autres grilles peuvent être facilement créées à partir des fiches [13](#Fiche13) et [14](#Fiche14). | Fiches [3](#Fiche3) et [4](#Fiche4) |
| 3  Découvrir la logique autocentrée (ou déplacements relatifs)  40 min | Approcher la programmation de déplacements selon une logique autocentrée (ou déplacements relatifs) | À partir du CE1, il sera intéressant de passer rapidement à une logique autocentrée (ou relative). Elle s’avère plus compliquée que la logique allocentrée, mais est une étape indispensable vers la programmation de déplacements de robots qui utilisent tous cette logique.  - L’enseignant projette au tableau la [fiche 5](#Fiche5) et demande aux élèves répartis en binômes de donner des ordres au robot pour qu’il aille jusqu’à sa base de rechargement.  - Insister sur le fait que, contrairement au robot, elle ne peut pas se déplacer sur le côté sans avoir préalablement tourné.  - Après quelques minutes de recherches, un binôme propose son travail.  - Le programme est effectué instruction par instruction au tableau par un autre élève pour le valider ou l’invalider. La nécessité de « pivoter » avant de pouvoir se déplacer sur le côté ne sera probablement pas trouvée par un certain nombre d’élèves.  - En fonction des travaux réalisés, montrer les travaux de quelques groupes. Faire réaliser le programme comme le ferait le robot, c'est-à-dire en pivotant sur place lorsque l’instruction « pivoter » est donnée (l’instruction flèche à droite sera exécuter comme pivoter). Cela permettra de faire remarquer qu’en logique autocentrée, une seule erreur peut entraîner le robot très loin de ce qui était voulu.  - Trouver un consensus sur un code simple et efficace (privilégier les flèches avancer, reculer, pivoter à droite, pivoter à gauche) que tous les élèves utiliseront à partir de la séance suivante. Un exemple est fait au tableau avec les cartes de la [fiche 1](#Fiche1).  - D’autres situations peuvent être représentées en plaçant simplement le point de départ du robot et la base à différents endroits sur le quadrillage.  - La classe synthétise collectivement ce qui a été appris au cours de cette séance.  **Exemple de trace écrite à adapter en fonction de l’âge des élèves :**  Lorsque l’on donne des instructions de déplacement à un robot, on utilise généralement une logique appelée autocentrée (en opposition à la logique allocentrée). Cela signifie que l’on imagine être à la place du robot. | Ajouter les cartes « pivoter » de la [fiche 1](#Fiche1)  [Fiche 5](#Fiche5) |
| 4  Vivre la logique autocentrée  30 min | Comprendre la logique autocentrée pour programmer des déplacements | À l’issue de la séance 3, certains élèves éprouveront probablement encore des difficultés à donner des instructions en logique autocentrée. Cette séance consiste à leur faire vivre la situation concrètement de manière à ce qu’ils l’intègrent plus facilement.  Par groupe de 4, chaque élève va, à tour de rôle, être le « robot », pendant que ses camarades lui donneront des instructions pour l’aider à se rendre à sa base.  - Chaque groupe d’élèves travaille sur un damier.  - L’enseignant demande à un élève de se placer aléatoirement sur le damier. Il sera le « robot ».  - Les autres élèves du groupe définissent une case sur le damier qui sera la base de rechargement mais n’en informent pas le « robot ». Ils se mettent ensuite d’accord sur les instructions à donner au « robot » pour qu’il retourne à sa base.  - L’enseignant demande à ce que les élèves écrivent l’intégralité du programme avant de commencer à donner les instructions au « robot » et rappelle que ce dernier ne peut effectuer qu’une seule action à la fois : avancer, reculer, pivoter à gauche (sans changer de case) ou pivoter à droite (sans changer de case).  - Une fois le programme écrit, les instructions sont données à « l’élève robot » oralement, une à une. Celui-ci les exécute au fur et à mesure qu’on les lui donne. À la fin de l’exécution du programme, l’élève robot doit être arrivé sur la case définie par ses camarades comme étant la base de rechargement.  - L’exercice est recommencé jusqu’à ce que chaque élève ait eu au moins une fois le rôle de robot.  - Si certains groupes sont très à l’aise, l’enseignant peut leur proposer d’ajouter des obstacles et/ou d’agrandir le damier. | Un espace assez grand (cour, gymnase, salle de motricité…)  Autant de damier (4x4 ou 5x5) que de groupes de 4 élèves seront préalablement tracés ou matérialisés au sol par l’enseignant  1 feuille et 1 crayon par groupe  Éventuellement des objets pour matérialiser des obstacles |
| 5  Programmer des déplacements en logique autocentrée  30 min | Savoir programmer des déplacements selon une logique autocentrée. | - L’enseignant rappelle ce qui a été appris lors des deux séances précédentes puis distribue la [fiche 6](#Fiche6). Il demande aux élèves d’écrire individuellement le programme (ou algorithme) avec le langage choisi lors de la séance 3.  - Après quelques minutes de travail, un élève montre son programme. Il demande ensuite si quelqu’un a réussi à écrire un programme plus court. Si oui, alors on le valide ou l’invalide collectivement.  - L’enseignant distribue la [fiche 7](#Fiche7). Les étapes décrites pour la fiche 6 sont ensuite répétées.  - La classe synthétise collectivement ce qui a été appris au cours de cette séance.  **Exemple de trace écrite à adapter en fonction de l’âge des élèves :**  La logique autocentrée demande plus d’instructions que la logique allocentrée pour effectuer un parcours identique car pour que le robot se déplace sur le côté, on doit lui donner deux instructions : d’abord pivoter, puis avancer. Avec cette logique, une seule erreur peut entraîner un point d’arrivée très éloigné de celui attendu.  Des séances de prolongement « branchées » peuvent être effectuées, avec le logiciel ou l’application tablette « Tuxbot ».  Pour s’entrainer en logique autocentrée, désactiver dans les paramètres le mode « Basic édition ». Si nécessaire, le logiciel propose également un mode entraînement dans lequel les déplacements sont effectués en direct plutôt que par anticipation.  Pour les classes non équipées, d’autres grilles peuvent être facilement créées à partir des fiches [15](#Fiche15), [16](#Fiche16) et [17](#Fiche17). | Fiches [6](#Fiche6) et [7](#Fiche7) |
| 6  Découvrir la notion de boucle  30 min | Comprendre la notion de boucle | Une fois le principe du déplacement acquis, il sera intéressant d’apprendre à utiliser des boucles (répétition d’une séquence d’instructions) afin d’économiser des lignes de code.  - L’enseignant projette au tableau la [fiche 9](#Fiche9) et demande aux élèves répartis en binômes d’écrire le programme (ou l’algorithme) permettant au robot d’aller jusqu’à sa base de rechargement.  - Après quelques minutes de recherches, un binôme propose son travail.  - Le programme est effectué instruction par instruction au tableau.  - Si la remarque ne vient pas naturellement, demander aux élèves quelles particularités contient ce programme. Cette fiche ne permet volontairement qu’un seul parcours afin de permettre aux élèves de remarquer que la même série d’instructions est répétée plusieurs fois. Leur demander alors s’ils ont une idée qui permettrait de simplifier notre programme. Les amener jusqu’à la boucle qui consiste à répéter un certain nombre de fois la même série d’instructions.  - L’enseignant montre aux élèves une manière de coder une boucle :  Pour l’exemple de la [fiche 9](#Fiche9), le programme sera :  [ ] x 6  (Le fait que la dernière instruction pivoter à gauche soit inutile lors de la dernière boucle n’est pas un problème, l’important étant que le robot, à l’issue de l’exécution du programme, soit sur la bonne case, peu importe l’orientation.)  - D’autres situations peuvent être représentées en plaçant le point de départ du robot et la base à différents endroits sur le quadrillage (ne pas les placer complètement aléatoirement si l’on veut pouvoir utiliser des boucles).  - La classe synthétise collectivement ce qui a été appris au cours de cette séance.  **Exemple de trace écrite à adapter en fonction de l’âge des élèves :**  Parfois, lorsque l’on donne des instructions de déplacement à un robot, la même série d’instructions peut se répéter plusieurs fois. On peut alors utiliser des boucles pour que notre programme soit plus court. | Cartes de la [fiche 1](#Fiche1)  [Fiche 8](#Fiche8) (Cartes à découper pour les boucles)  [Fiche 9](#Fiche9) |
| 7  Programmer des déplacements intégrant des boucles  40 min | Savoir programmer des déplacements avec des boucles  S’engager dans une démarche de recherche | - L’enseignant rappelle ce qui a été appris lors de la séance précédente puis distribue la [fiche 10](#Fiche10). Il demande aux élèves d’écrire individuellement le programme (ou algorithme) avec le langage choisi lors de la séance précédente (ajout de la notion de boucle).  - Après quelques minutes de travail, un élève montre son programme. Il demande ensuite si quelqu’un a réussi à écrire un programme plus court. Si oui, alors on le valide ou l’invalide collectivement.  Il y a plusieurs possibilités pour ce programme (même si la première est plus courte, il sera intéressant de le faire remarquer aux élèves s’ils ne les ont pas trouvées d’eux-mêmes).  [ ] x 6  ou  [ ] x 2  ou encore  [ ] x 6 [ ] x 6  - L’enseignant distribue la [fiche 11](#Fiche11). Les étapes décrites pour la fiche 10 sont ensuite répétées. Pour la [fiche 11](#Fiche11), le programme le plus court est :  [ ] x 3  - L’enseignant distribue la [fiche 12](#Fiche12). Les étapes décrites pour la fiche 10 sont ensuite répétées. L’intérêt de cette fiche est de montrer aux élèves qu’une boucle ne constitue pas toujours un programme complet, elle doit souvent être associée à d’autres instructions hors boucle. Ici, nous limitons volontairement le nombre d’instructions pour inciter les élèves à rechercher la meilleure solution. Les autres ne sont pas incorrectes pour autant si l’on reste sur de la programmation pure.  Pour la [fiche 12](#Fiche12), le programme est :  [ ] x 2  - La classe synthétise collectivement ce qui a été appris au cours de cette séance.  **Exemple de trace écrite à adapter en fonction de l’âge des élèves :**  Dans un programme, on peut mélanger des boucles avec des instructions simples. On essaye d’écrire l’algorithme le plus court possible. Parfois, on ne le trouve pas, notre algorithme n’est pas faux pour autant, le plus important étant qu’il résolve notre problème.  Des séances de réinvestissement « branchées » peuvent être effectuées, avec le logiciel ou l’application tablette « Tuxbot ».  Pour s’entrainer avec des boucles, sélectionner dans les paramètres l’« interface avancée ». Attention, le mode de fonctionnement est légèrement différent car la boucle est créée dans un programme d’instructions à part (appelé registre mémoire) et est ensuite appelée dans le programme principal. Cette différence sera vite comprise par les élèves s’ils ont déjà compris les boucles mais nécessitera d’être explicitée par l’enseignant avant de débuter l’activité « branchée ». À partir du niveau 21, vous pouvez créer vos propres scènes de jeu. Pour l’instant le logiciel ne permet pas de dupliquer automatiquement les parcours créés. En attendant que la fonctionnalité soit développée, vous pouvez copier le fichier "C:\Utilisateurs\<NomUtilisateur>\ApplicationData\Roaming\tuxbot\Local/Store\#SharedObjects\tuxbot.swf\tuxbot.sol" d'un ordinateur à l'autre. Ce chemin vaut pour windows 7 et 10.  Pour les classes non équipées, d’autres grilles peuvent être facilement créées à partir des fiches [15](#Fiche15), [16](#Fiche16) et [17](#Fiche17). | Fiches [10](#Fiche10), [11](#Fiche11) et [12](#Fiche12) |

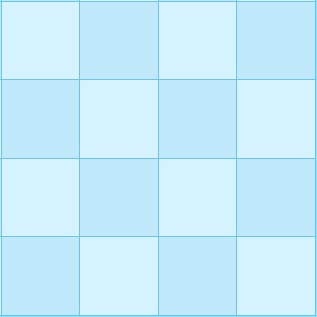
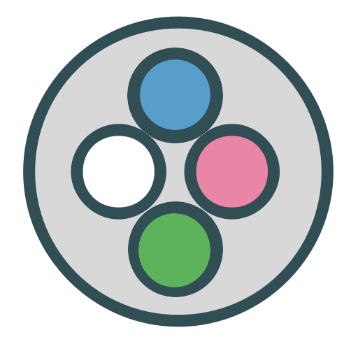
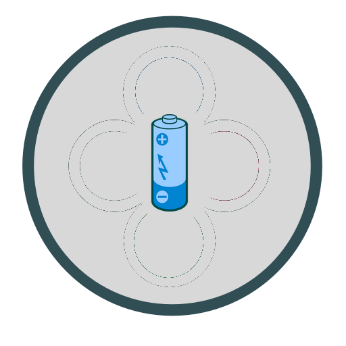
Au cycle 3, il sera intéressant de poursuivre avec une séquence utilisant un langage de programmation plus complexe (voir par exemple la séquence proposée par le réseau CANOPE du département des Yvelines à partir du logiciel scratch : <https://www.reseau-canope.fr/atelier-yvelines/spip.php?article1158>) ou les vidéos et exercices d’entrainement du campus junior (connexion internet obligatoire) : <https://www.lecampusjunior.fr/#!/)>

Fiche 1

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Fiche 2

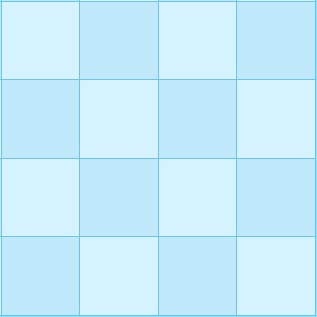
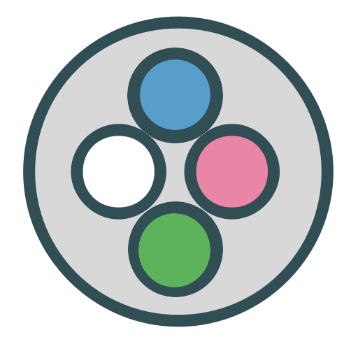
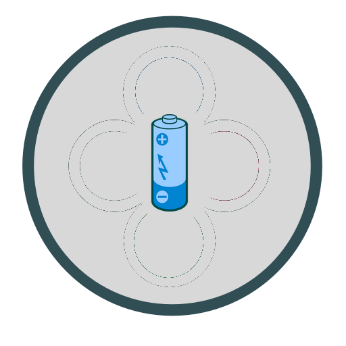
La batterie du robot est faible, écris des instructions pour qu’il aille sur sa base de rechargement le plus vite possible. Il ne peut se déplacer que verticalement ou horizontalement.



Fiche 3

La batterie du robot est faible, écris dans le tableau ci-dessous des instructions pour qu’il aille sur sa base de rechargement le plus vite possible. Il ne peut se déplacer que verticalement ou horizontalement.

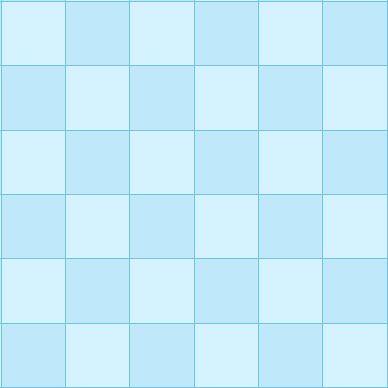
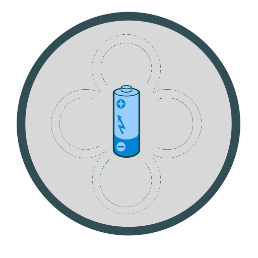
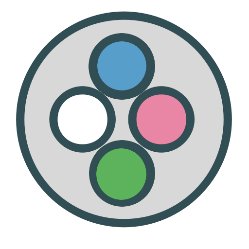
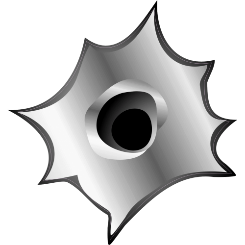
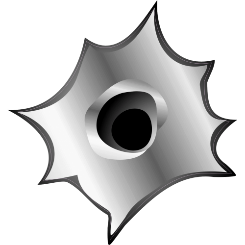
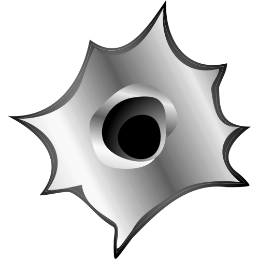
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |



Fiche 4

La batterie du robot est faible, écris dans le tableau ci-dessous des instructions pour qu’il aille sur sa base de rechargement le plus vite possible. Il ne peut se déplacer que verticalement ou horizontalement. Attention, il doit éviter les obstacles !

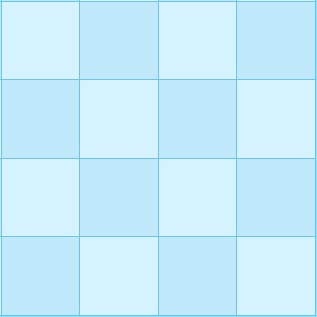
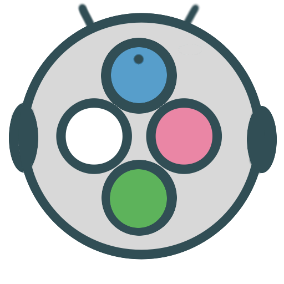
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |



Fiche 5

La batterie du robot est faible, écris dans le tableau ci-dessous des instructions pour qu’il aille sur sa base de rechargement le plus vite possible. Attention, ce robot a un sens, tu dois en tenir compte dans tes instructions. Il peut seulement se déplacer en avant, en arrière ou tourner sur lui-même.

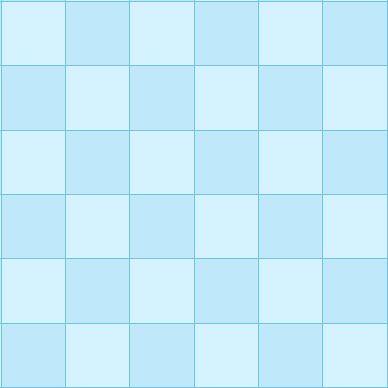
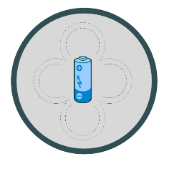
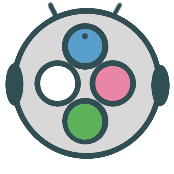
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |



Fiche 6

La batterie du robot est faible, écris dans le tableau ci-dessous des instructions pour qu’il aille sur sa base de rechargement le plus vite possible. Attention, ce robot a un sens, tu dois en tenir compte dans tes instructions. Il peut seulement se déplacer en avant, en arrière ou tourner sur lui-même.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |

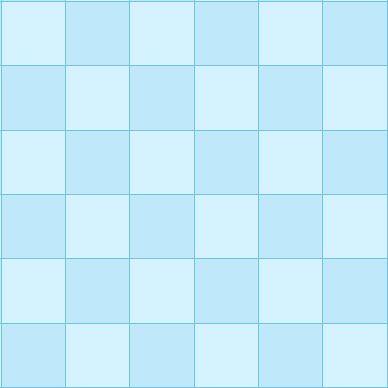
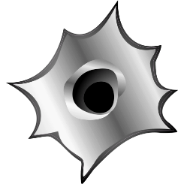
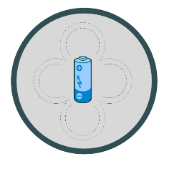
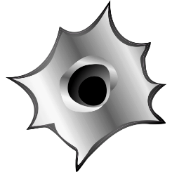
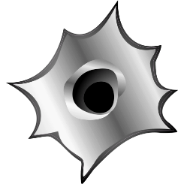
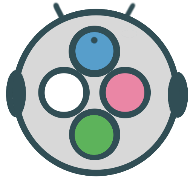


La batterie du robot est faible, écris dans le tableau ci-dessous des instructions pour qu’il aille sur sa base de rechargement le plus vite possible. Attention, ce robot a un sens, tu dois en tenir compte dans tes instructions. Il

Fiche 7

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

peut seulement se déplacer en avant, en arrière ou tourner sur lui-même.



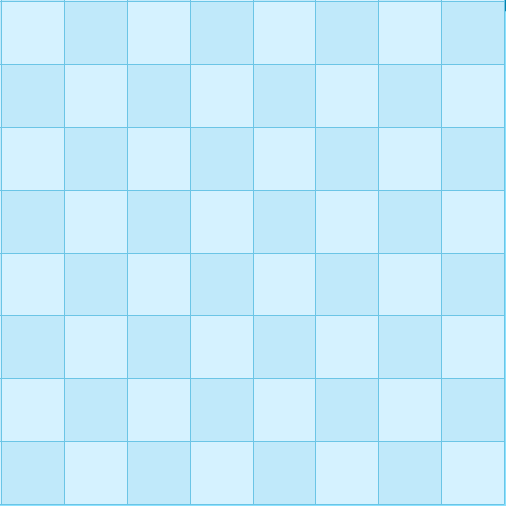
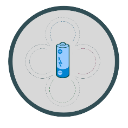
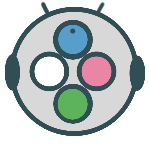
Fiche 8

|  |  |  |
| --- | --- | --- |
| [ | ] | [ |
| ] | X2 | X3 |
| X4 | X5 | X6 |
| X7 | X8 | X9 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

La batterie du robot est faible, écris dans le tableau ci-dessous des instructions pour qu’il aille sur sa base de rechargement le plus vite possible.

Fiche 9

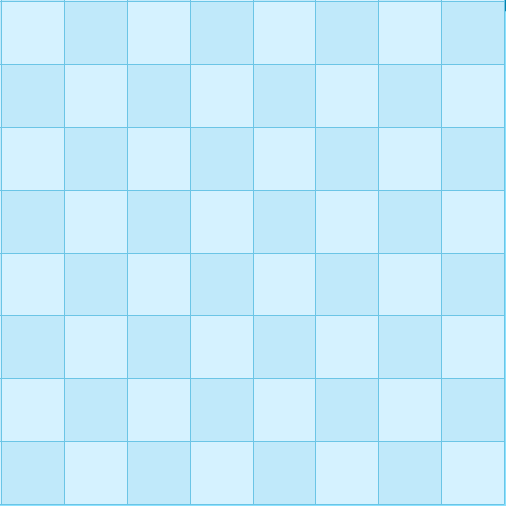
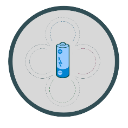
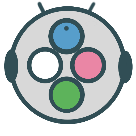


La batterie du robot est faible, écris dans le tableau ci-dessous le moins d’instructions possible pour qu’il aille sur sa base de rechargement. Attention, tu ne peux pas donner plus d’instructions qu’il n’y a de cases dans le tableau !

Fiche 10

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |

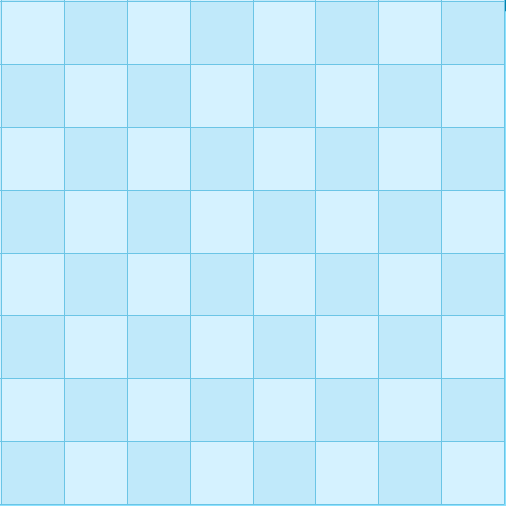
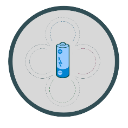
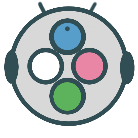




La batterie du robot est faible, écris dans le tableau ci-dessous le moins d’instructions possible pour qu’il aille sur sa base de rechargement. Attention, tu ne peux pas donner plus d’instructions qu’il n’y a de cases dans le tableau !

Fiche 11

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |

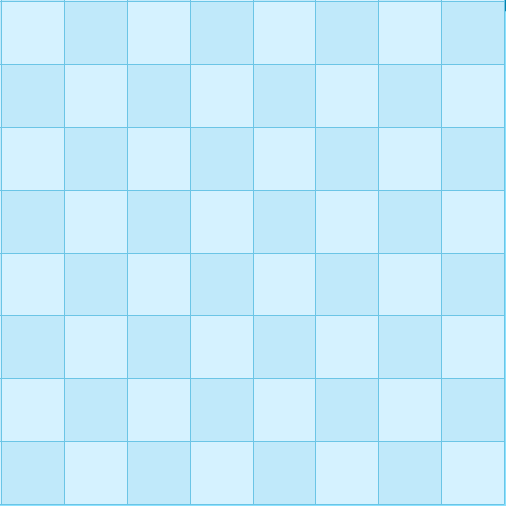
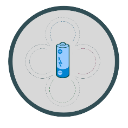
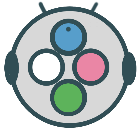


La batterie du robot est faible, écris dans le tableau ci-dessous le moins d’instructions possible pour qu’il aille sur sa base de rechargement. Attention, tu ne peux pas donner plus d’instructions qu’il n’y a de cases dans le tableau !

Fiche 12

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |

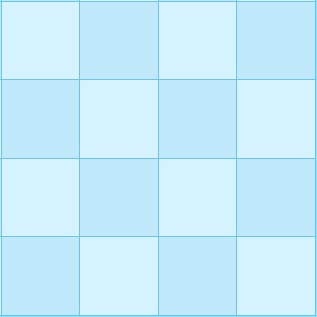


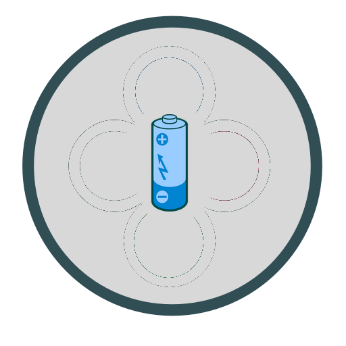
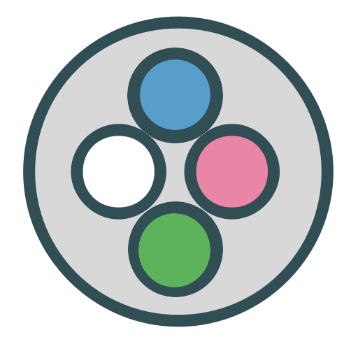


Fiche 13 (pour la création de nouvelles fiches)

La batterie du robot est faible, écris dans le tableau ci-dessous des instructions pour qu’il aille sur sa base de rechargement le plus vite possible. Il ne peut se déplacer que verticalement ou horizontalement.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |

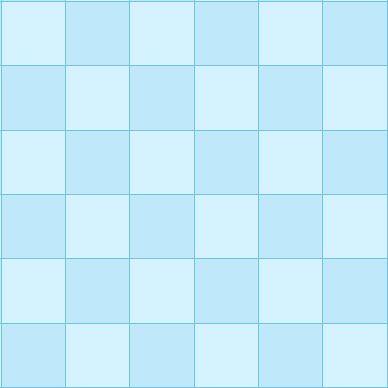
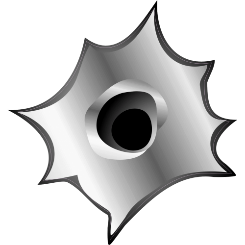
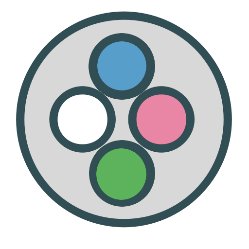
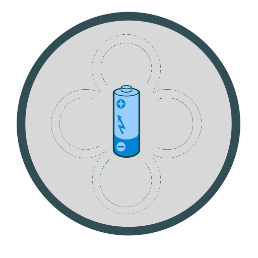




Fiche 14 (pour la création de nouvelles fiches)

La batterie du robot est faible, écris dans le tableau ci-dessous des instructions pour qu’il aille sur sa base de rechargement le plus vite possible. Il ne peut se déplacer que verticalement ou horizontalement. Attention, il doit éviter les obstacles !

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |

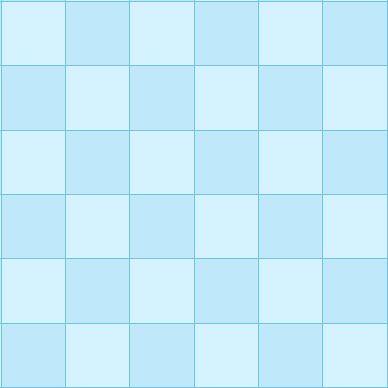
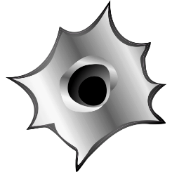
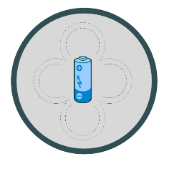
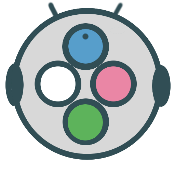


La batterie du robot est faible, écris dans le tableau ci-dessous des instructions pour qu’il aille sur sa base de rechargement le plus vite possible. Attention, ce robot a un sens, tu dois en tenir compte dans tes instructions. Il

Fiche 15 (pour la création de nouvelles fiches)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

peut seulement se déplacer en avant, en arrière ou tourner sur lui-même.

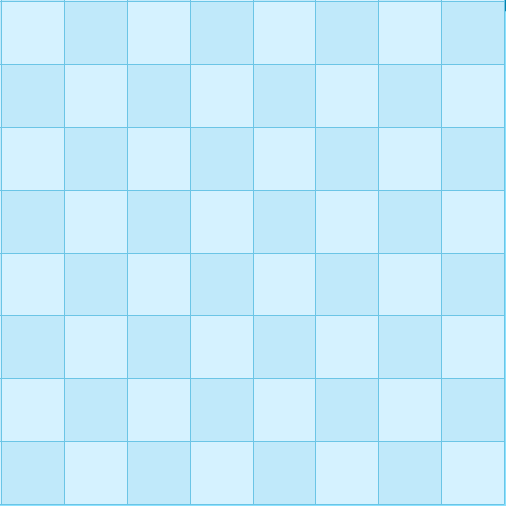
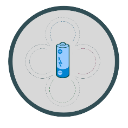
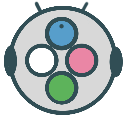


La batterie du robot est faible, écris dans le tableau ci-dessous des instructions pour qu’il aille sur sa base de rechargement le plus vite possible. Attention, ce robot a un sens, tu dois en tenir compte dans tes instructions. Il

Fiche 16 (pour la création de nouvelles fiches)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

peut seulement se déplacer en avant, en arrière ou tourner sur lui-même.

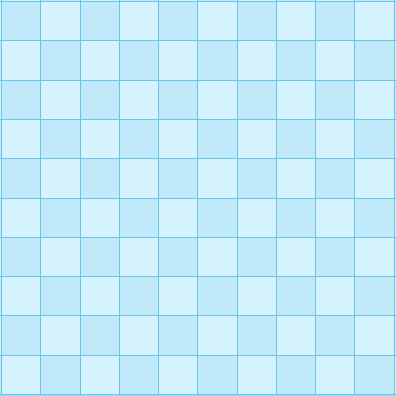
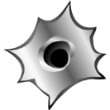
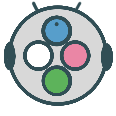


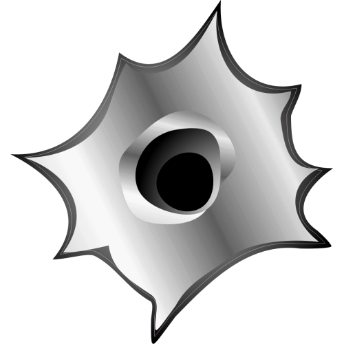
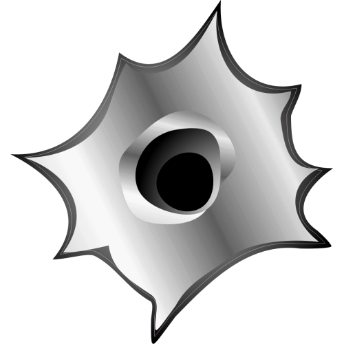
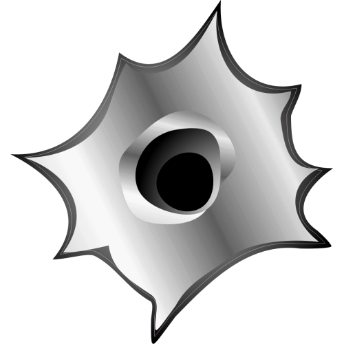
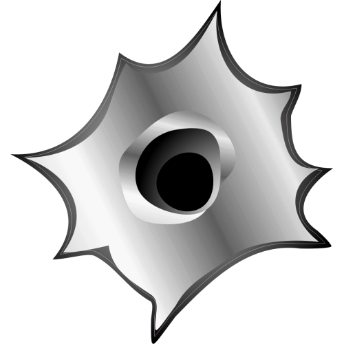
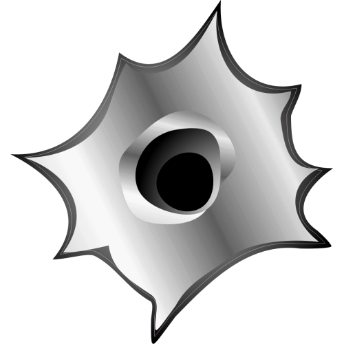
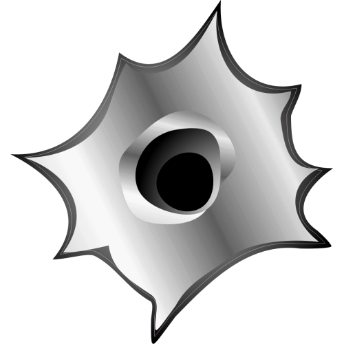
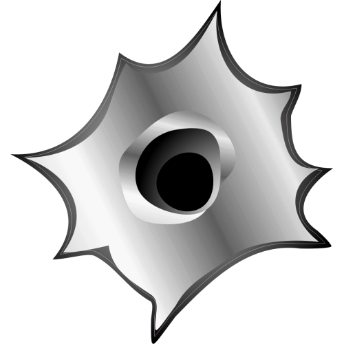
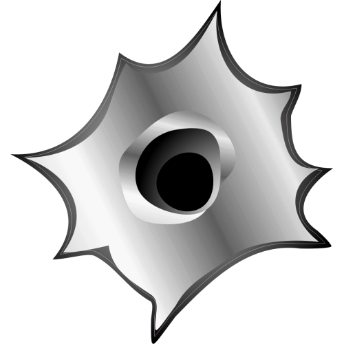
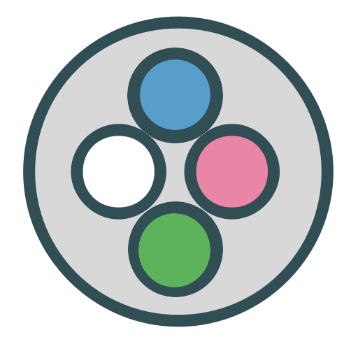
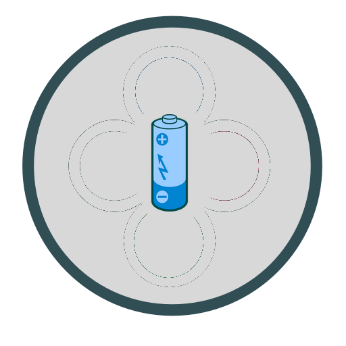
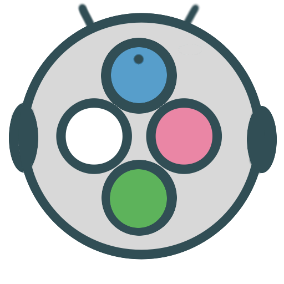
La batterie du robot est faible, écris dans le tableau ci-dessous des instructions pour qu’il aille sur sa base de rechargement le plus vite possible. Attention, ce robot a un sens, tu dois en tenir compte dans tes instructions. Il

Fiche 17 (pour la création de nouvelles fiches)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

peut seulement se déplacer en avant, en arrière ou tourner sur lui-même.





Fiche 18 (pour les manipulations)