



CODE A L'ECOLE

Séquence pour une première approche de la programmation

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

Programme Maths - 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

Compétence 3.4 : programmer

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

Niveau 2 : Réaliser un programme simple.



Déplacer un objet sur un quadrillage



Objectifs : Approcher la programmation de déplacements selon une logique allocentrée (ou déplacements absolus)

Étape 1



Individuel
Recherche

L'enseignant projette (ou reproduit au tableau) la **FICHE 2** 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.**

Étape 2



Collectif
Mise en commun

Après quelques minutes de recherche, 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 partir de la séance suivante.
Un exemple est fait au tableau avec les cartes de la **FICHE 2**.
D'autres situations peuvent être représentées en plaçant simplement le point de départ du robot et la base à différents endroits sur le quadrillage.

Étape 3



Trace écrite

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.



Programmer des déplacements absolus



Objectifs : Savoir programmer des déplacements selon une logique allocentrée.

Étape 1



Individuel
Recherche

L'enseignant rappelle ce qui a été appris lors de la séance précédente puis distribue la **FICHE 3**

Il demande aux élèves d'écrire individuellement le programme (ou algorithme) avec le langage choisi lors de la séance précédente.

Étape 2



Collectif
Mise en commun

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'invalidé collectivement.

Étape 3



Individuel
Entrainement

L'enseignant distribue la **FICHE 4**. Les étapes décrites pour la fiche 3 sont ensuite répétées.

Étape 4



Trace écrite

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: c'est l'optimisation.

Étape 5



Prolongement

Voir **ACTIVITE BRANCHEE 1**



i À partir du CEI, 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.

Découvrir les déplacements relatifs



Objectifs : Approcher la programmation de déplacements selon une logique autocentrée (ou déplacements relatifs).

Étape 1



Individuel
Recherche

L'enseignant projette au tableau la **FICHE 5** 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.

Étape 2



Collectif
Mise en commun

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'invalidier. La nécessité de « pivoter » avant de pouvoir se déplacer sur le côté ne sera probablement pas identifiée 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 exécutée comme pivoter). Cela permettra de faire remarquer qu'en déplacement absolu, 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**

Étape 3



Trace écrite

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 doit s'imaginer être **à la place du robot** (déplacements relatifs)



i 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.

Vivre les déplacements relatifs



Objectifs : Comprendre les déplacements relatifs pour programmer des déplacements

Étape 1



Par groupe
Activité décrochée
En extérieur

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.



Programmer des déplacements relatifs



Objectifs : Savoir programmer des déplacements selon une logique autocentrée.

Étape 1



Individuel
Recherche

L'enseignant rappelle ce qui a été appris lors des deux séances précédentes puis distribue la **FICHE 6**. Il demande aux élèves d'écrire individuellement le programme (ou algorithme) avec le langage choisi lors de la séance 3.

Étape 2



Individuel
Entrainement

L'enseignant distribue la **FICHE 7**.
Les étapes décrites pour la fiche 6 sont ensuite répétées.

Étape 3



Trace écrite

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 :

Programmer un robot demande plus d'instructions 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**. Une seule erreur peut entraîner un point d'arrivée très éloigné de celui attendu.

Étape 4



Prolongement

Voir **ACTIVITE BRANCHEE 2**



i 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.

Découvrir la notion de boucle



Objectifs : Comprendre la notion de boucle

Étape 1



Individuel
Recherche

L'enseignant projette au tableau la **FICHE 9** 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.

Étape 2



Collectif
Mise en commun

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 (voir ci-contre).

[↑ ↻ ↑ ↻] X6

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).

Étape 3



Trace écrite

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.



Programmer des boucles



Objectifs : Savoir programmer des déplacements avec des boucles / S'engager dans une démarche de recherche

Étape 1



Individuel
Recherche

L'enseignant rappelle ce qui a été appris lors de la séance précédente puis distribue la **FICHE 10**. 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).

Étape 2



Collectif
Mise en commun

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'invalidé 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).

[C ↑ ↻ ↑] X6 [↑↑↑↑↑↑C] X2

Étape 3



Individuel
Entraînement

L'enseignant distribue la **FICHE 11**. Les étapes décrites pour la fiche 10 sont ensuite répétées. Pour la , le programme le plus court est :

[↑↑C↑↻] X3

L'enseignant distribue la **FICHE 12**. 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 , le programme est :

Étape 4



Prolongement

Voir **ACTIVITE BRANCHEE 3**



CODE A L'ECOLE

i 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:) ou les vidéos et exercices d'entraînement du campus junior (connexion internet obligatoire)



ACTIVITES BRANCHEES

Activité 1

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 :

Pour s'entraîner 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 15, 16 et 17

Activité 2

Des séances de prolongement «branchées» peuvent être effectuées, avec le logiciel ou l'application tablette «Tuxbot».

Pour s'entraîner 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, 16 et 17

Activité 3

Des séances de réinvestissement «branchées» peuvent être effectuées, avec le logiciel ou l'application tablette «Tuxbot».

Pour s'entraîner 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 débiter l'activité «branchée». À partir du niveau 21, vous pouvez créer vos propres scènes de jeu.

Pour les classes non équipées, d'autres grilles peuvent être facilement créées à partir des fiches 15, 16 et 17



Fiche 1



AVANCER



AVANCER



AVANCER



AVANCER



AVANCER



AVANCER



PIVOTER À DROITE



PIVOTER À DROITE



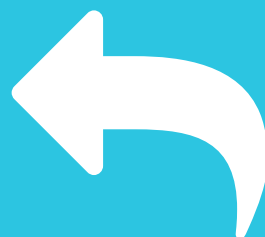
PIVOTER À DROITE



PIVOTER À GAUCHE



PIVOTER À GAUCHE

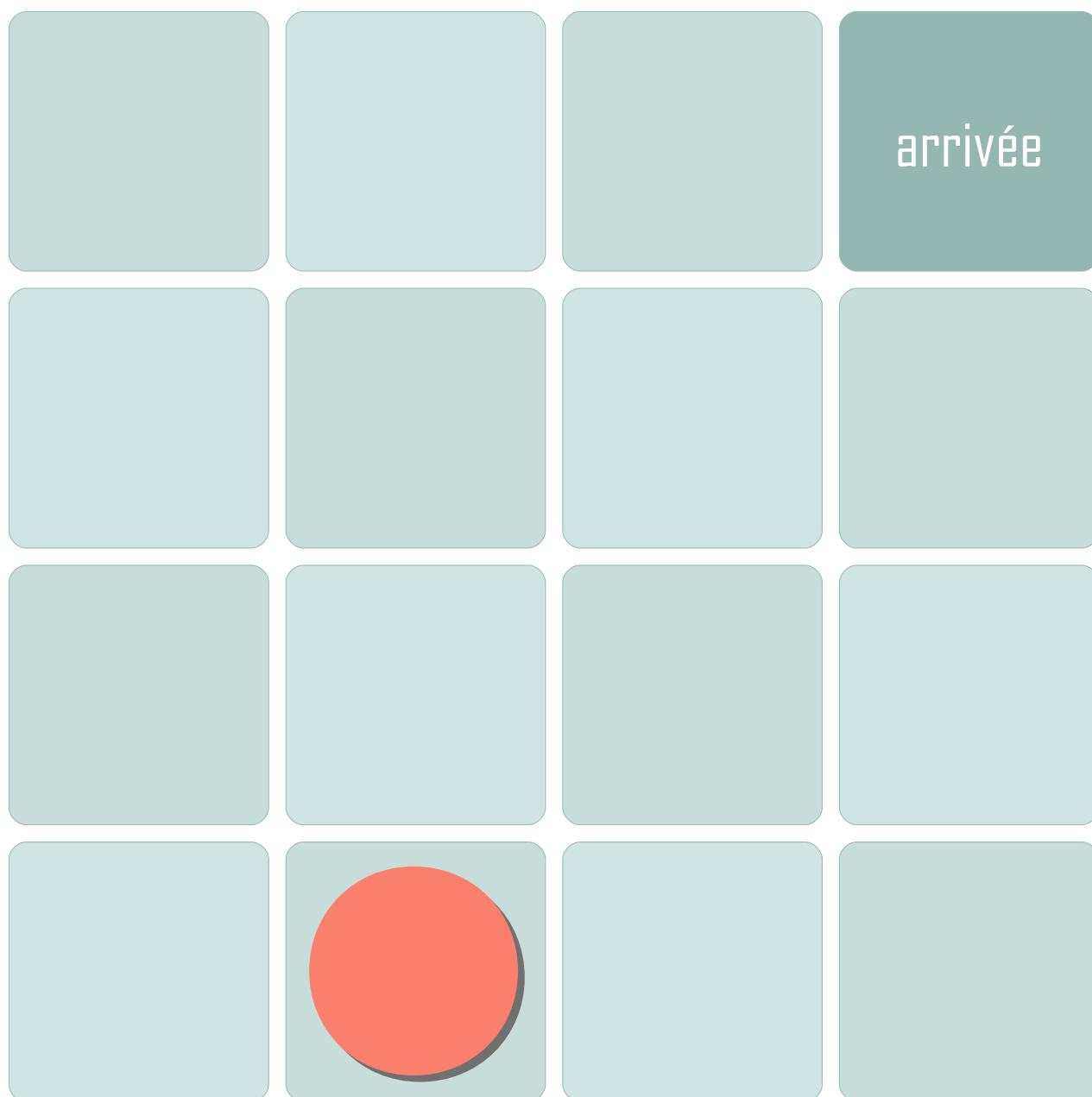


PIVOTER À GAUCHE



Fiche 2

Donne des instructions pour déplacer le pion sur la case foncée, Il ne peut se déplacer que verticalement ou horizontalement.

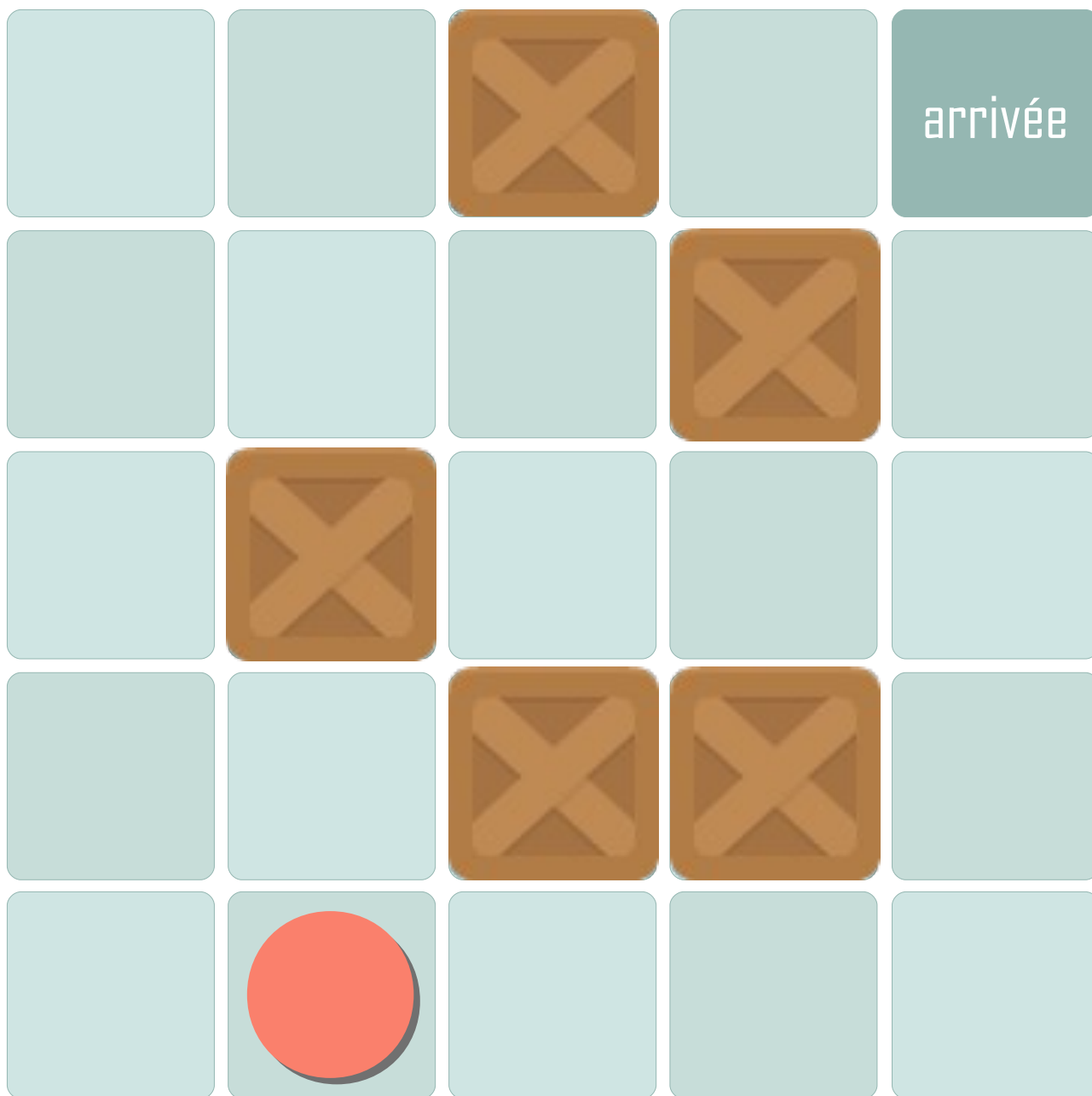




Fiche 4

Ecris dans les cases ci-dessous les instructions pour déplacer le pion sur la case foncée le plus vite possible, Il ne peut se déplacer que verticalement ou horizontalement. Attention, il faut éviter les obstacles.

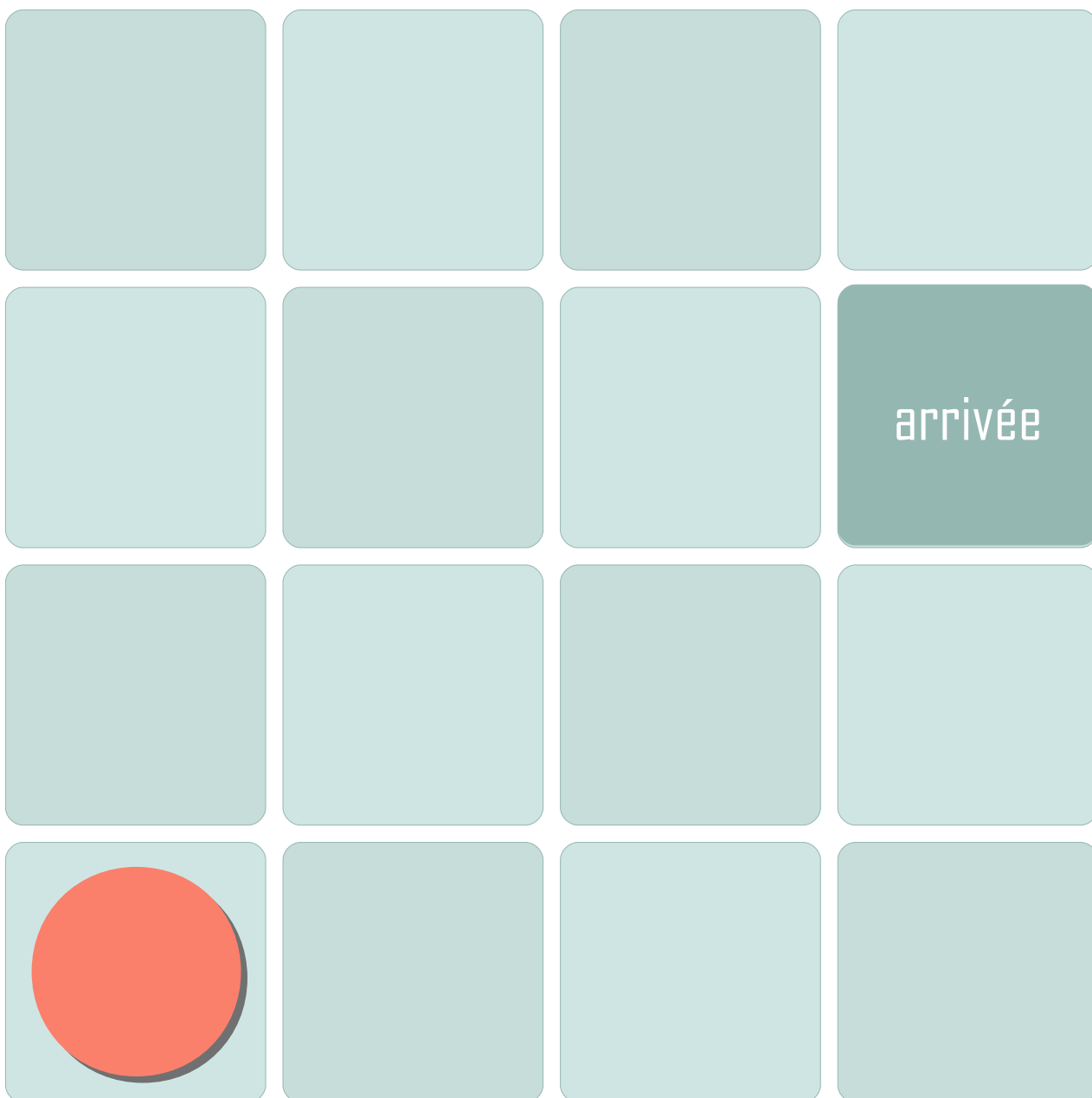
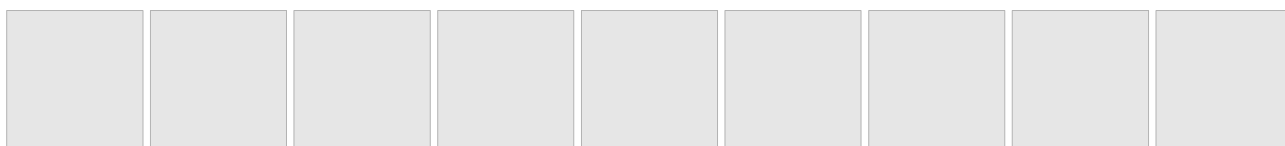
--	--	--	--	--	--	--	--	--	--	--





Fiche 3

Donne des instructions pour déplacer le pion sur la case foncée le plus vite possible, Il ne peut se déplacer que verticalement ou horizontalement.

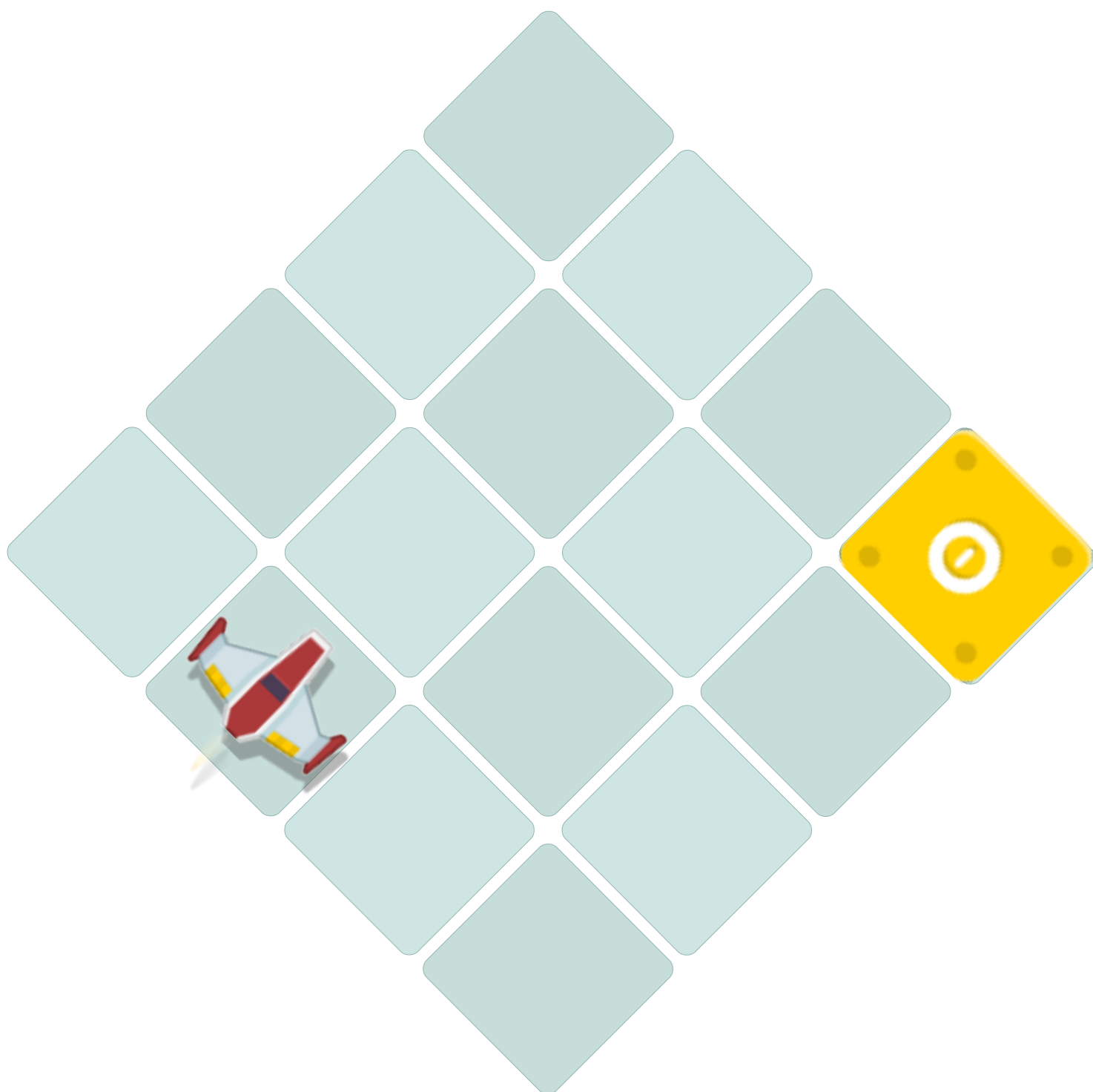




Fiche 5

La batterie de ce robot volant est faible, écris dans le tableau ci-dessous des instructions pour qu'il aille sur sa base de d'atterrissage 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.

--	--	--	--	--	--	--	--	--

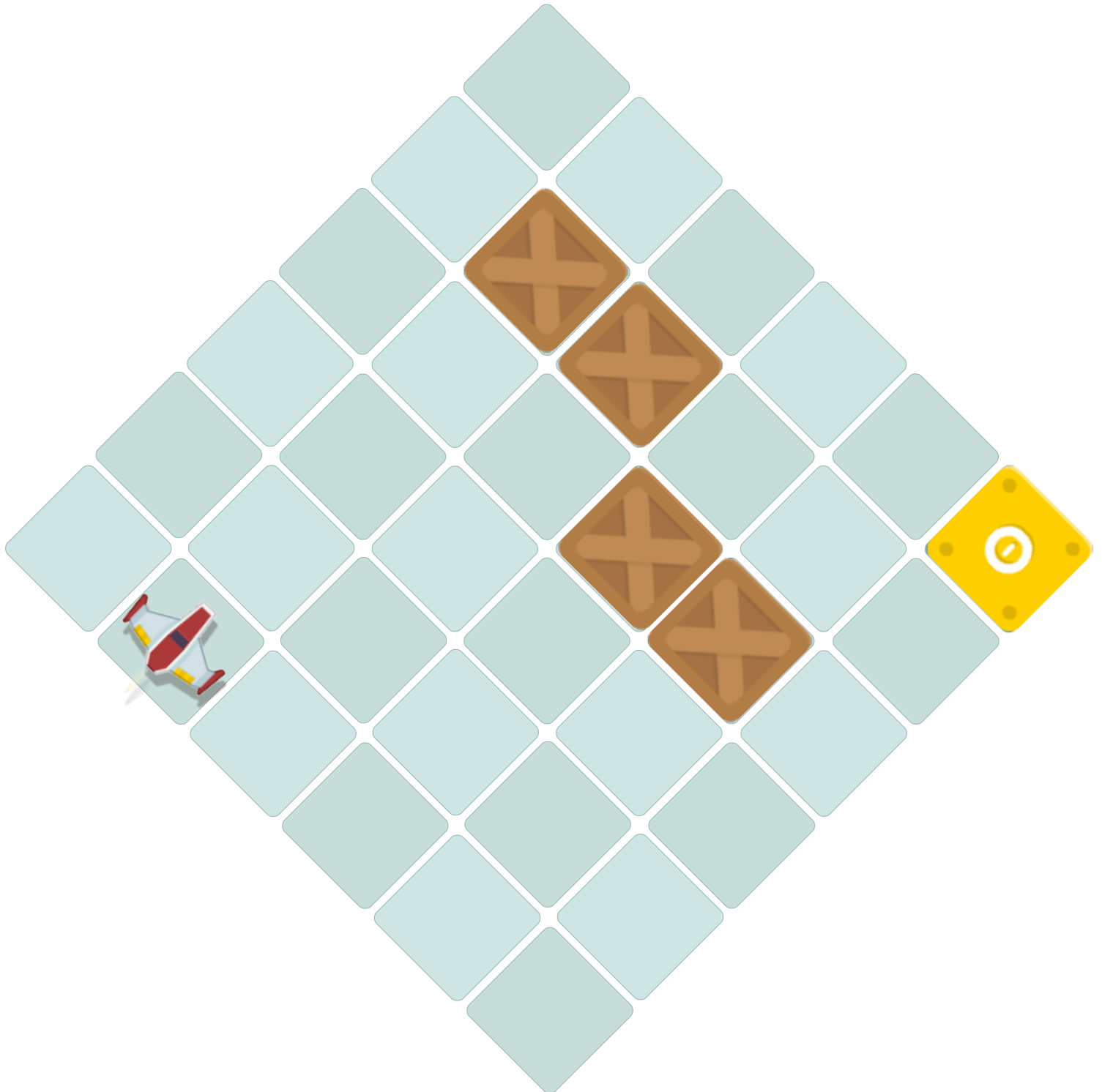




Fiche 6

La batterie de ce robot volant est faible, écris dans le tableau ci-dessous des instructions pour qu'il aille sur sa base de d'atterrissage 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.

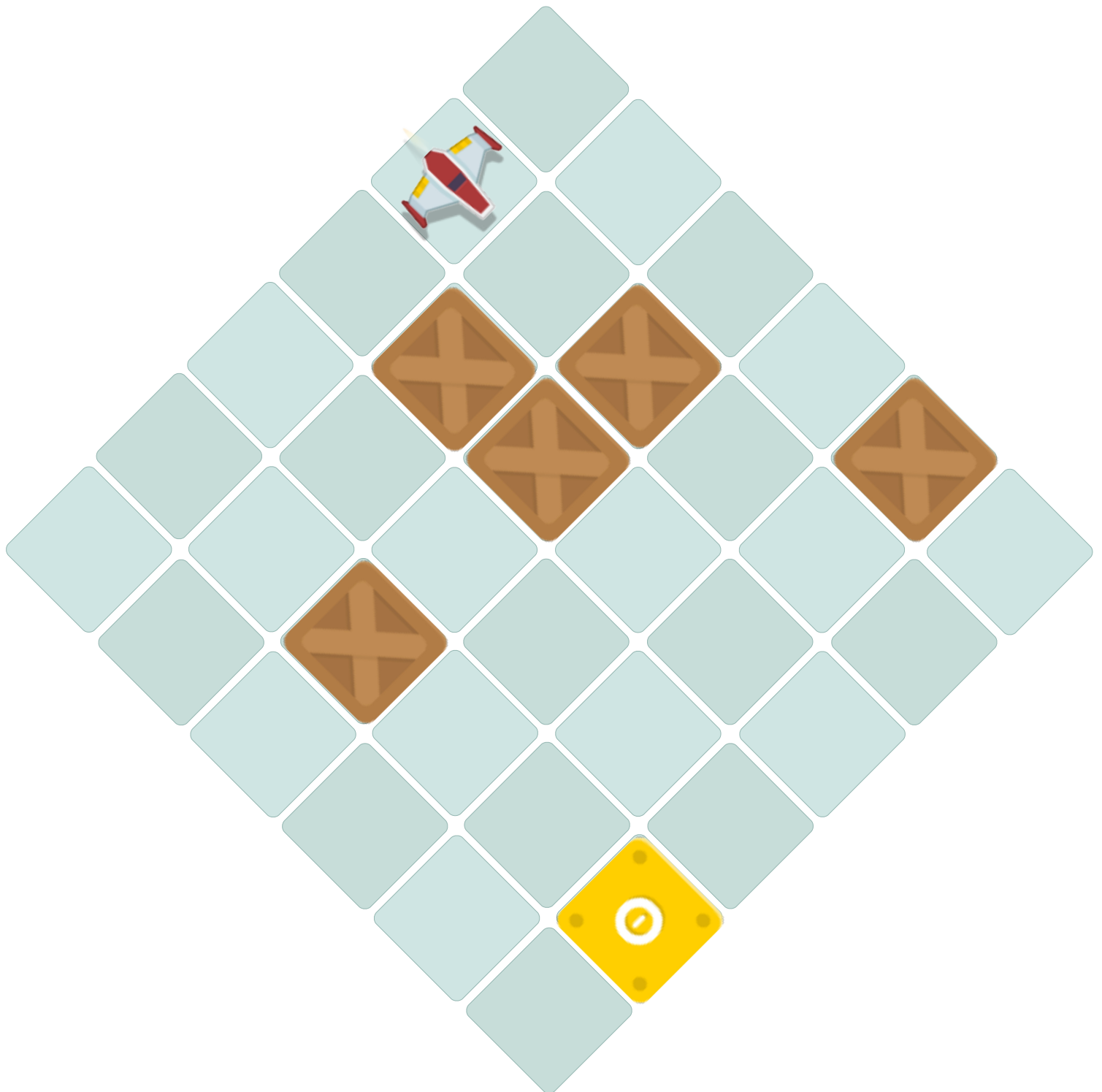
--	--	--	--	--	--	--	--	--	--	--



</> Fiche 7

La batterie de ce robot volant est faible, écris dans le tableau ci-dessous des instructions pour qu'il aille sur sa base de d'atterrissage 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.

--	--	--	--	--	--	--	--	--	--	--



[

DEBUT DE BOUCLE

[

DEBUT DE BOUCLE

]

FIN DE BOUCLE

]

FIN DE BOUCLE

×2

×3

×4

×5

×6

×7

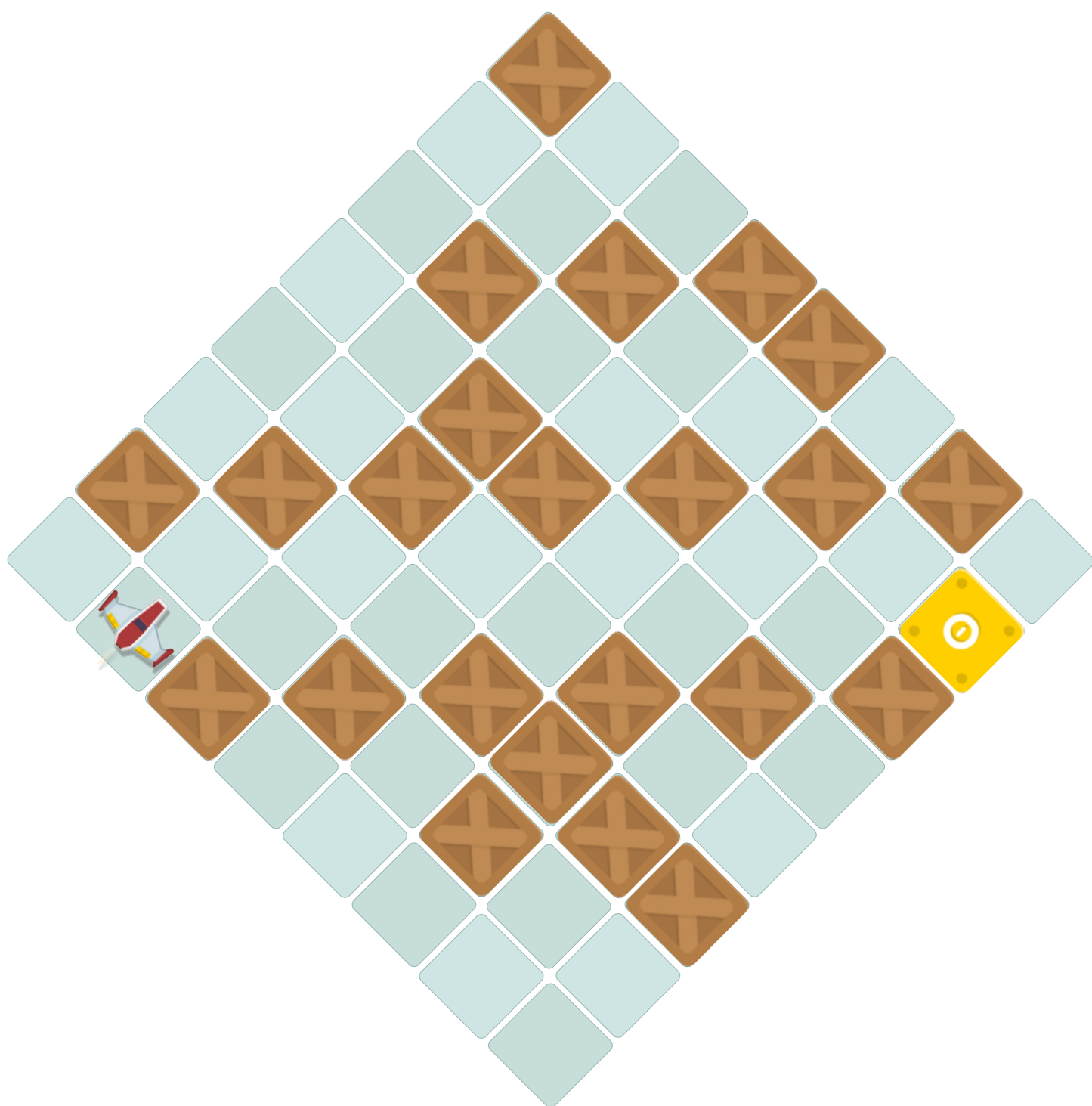
×8

×9



Fiche 9

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.

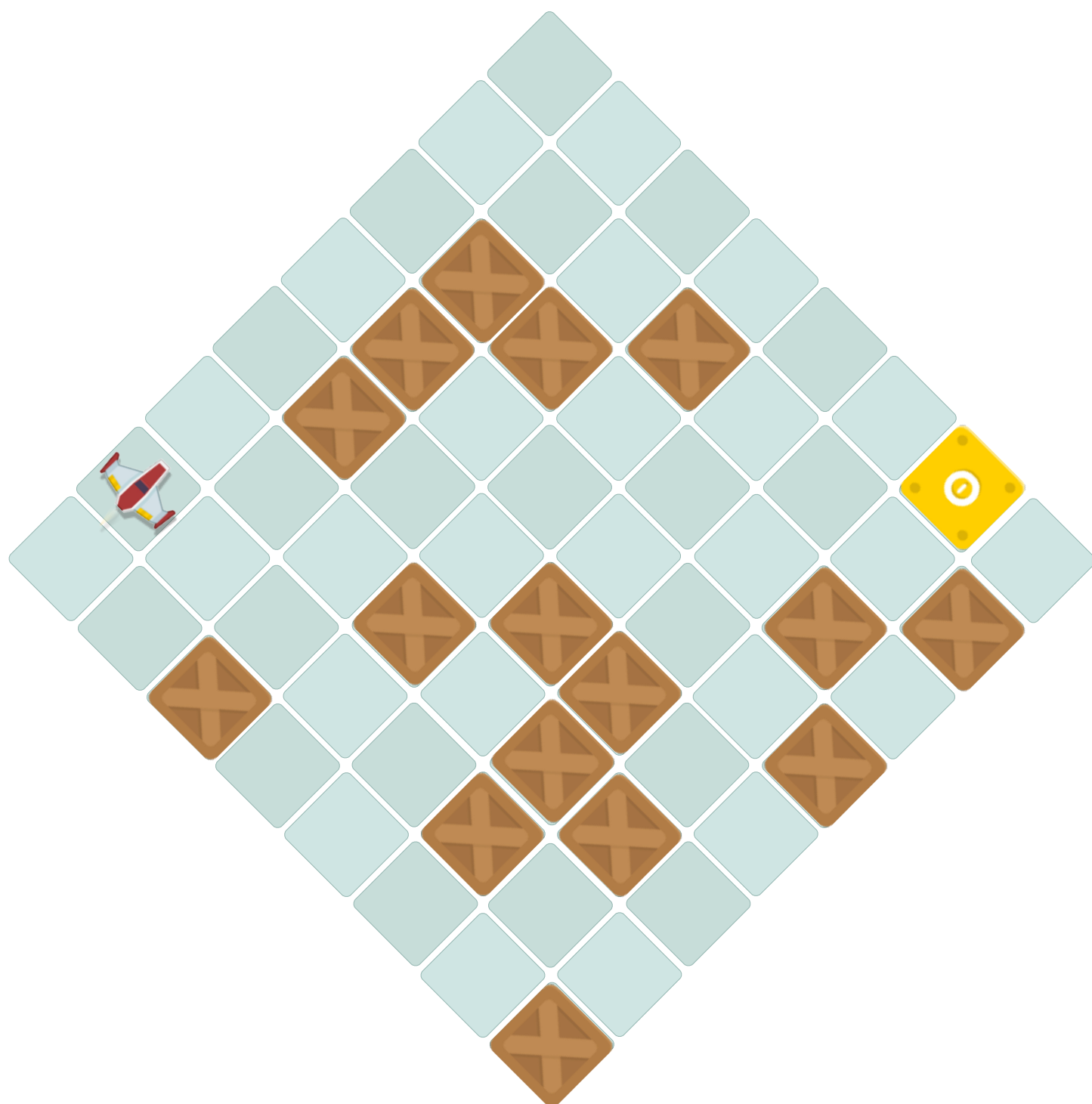




Fiche 10

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 !

--	--	--	--	--	--	--	--	--	--	--	--

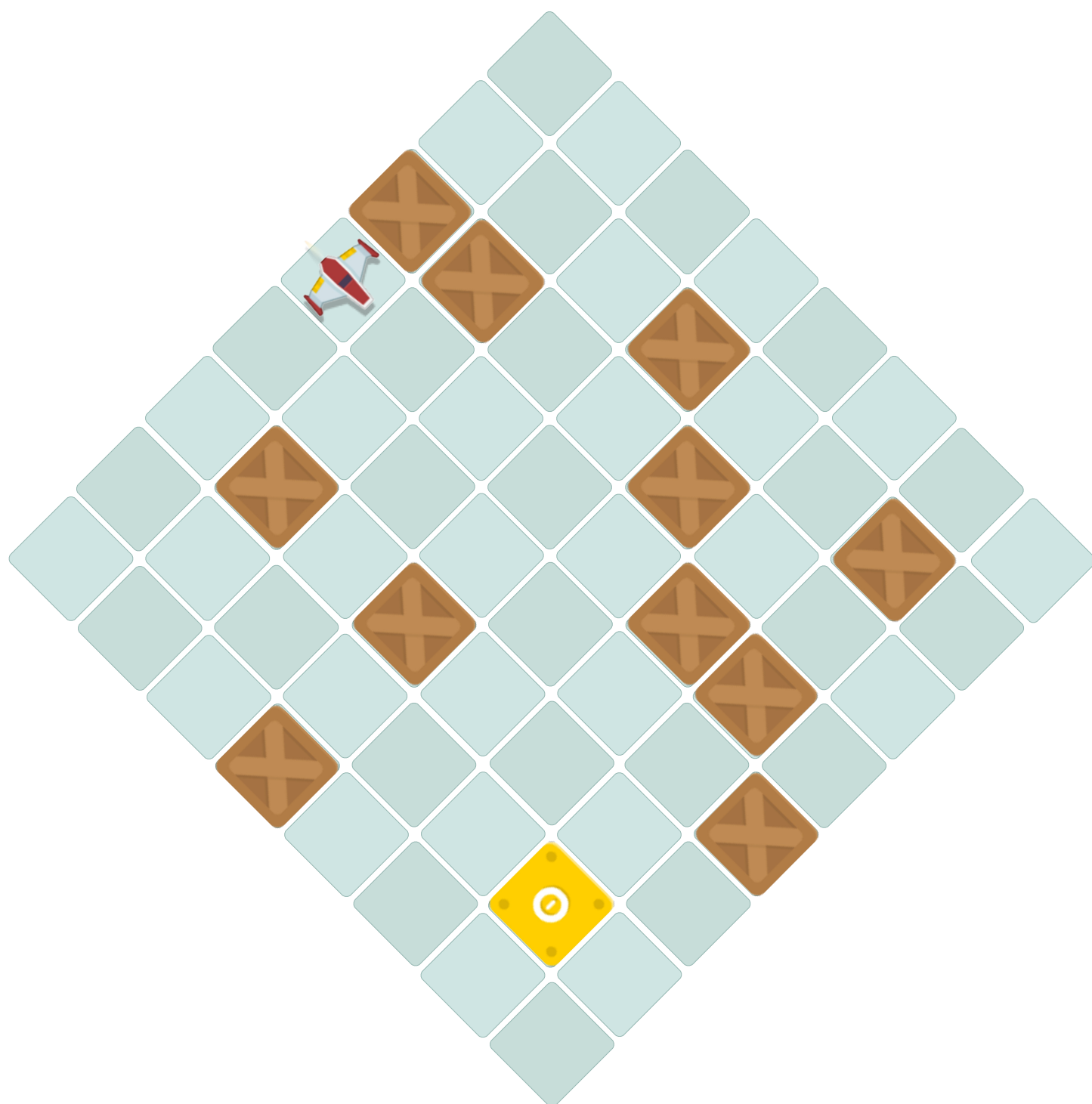




Fiche 11

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

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 !

--	--	--	--	--	--	--	--	--	--	--	--

