

---

# Pour une première approche de la programmation

---

Ces activités pédagogiques s'adressent aux élèves des cycles 2 et 3.  
Elles constituent une base d'apprentissage de la programmation et du codage.

---

# Sommaire

---

- 3 Introduction
- 4 Activités, objectifs et progression

---

## 5 **ACTIVITÉS DÉBRANCHÉES**

- 6 **Activité 1 —**  
Déplacer un objet sur un quadrillage
- 7 **Activité 2 —**  
Programmer des déplacements absolus
- 8 **Activité 3 — Vivre les déplacements relatifs**
- 9 **Activité 4 —**  
Découvrir les déplacements relatifs
- 10 **Activité 5 —**  
Programmer des déplacements relatifs
- 11 **Activité 6 —**  
Découvrir la notion de boucle
- 12 **Activité 7 —**  
Programmer des boucles
- 13 **Activité 8 — Les événements**

---

## 14 **ACTIVITÉS BRANCHÉES**

---

## 16 **FICHES D'ACTIVITÉ**

- 17 **Fiche 1-a**
- 18 **Fiche 1-b**
- 19 **Fiche 2**
- 20 **Fiche 3**
- 21 **Fiche 4**
- 22 **Fiche 5**
- 23 **Fiche 6**
- 24 **Fiche 7**
- 25 **Fiche 8**
- 26 **Fiche 9**
- 27 **Fiche 10**
- 28 **Fiche 11**
- 29 **Fiche 12**
- 30 **Fiche 13**
- 31 **Fiche 14**
- 32 **Fiche 15**

---

# Introduction

---

Ces activités s'adressent aux élèves de cycle 2 ou 3 n'ayant jamais abordé la programmation en classe. Elles sont dites « débranchées », au sens où elles ne nécessitent pas de matériel informatique (ordinateurs, tablettes, robots, etc). Les situations décrites permettent de développer des compétences en accord avec les programmes à travers l'utilisation du jeu « Planète Code » et des activités à mener en parallèle.

Pour les classes qui souhaiteraient aller plus loin, si elles sont équipées d'ordinateurs ou de tablettes, des activités de prolongement « branchées » sont proposées.

## PROGRAMME MATHÉMATIQUES - ESPACE ET GÉOMÉTRIE

Dès le CE1, les élèves peuvent **coder des déplacements**, un savoir qui les conduira 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.

# Activités, objectifs et progression

Liens de progression	OBJECTIFS	ACTIVITÉS							
		1	2	3	4	5	6	7	8
→	Approcher la programmation de déplacements absolus ou logique allocentrée	■							
	Comprendre la notion d'instruction, de programme, de code	■							
→	Savoir programmer des déplacements selon la logique allocentrée		■						
	Comprendre la notion d'optimisation en programmation		■						
→	Comprendre les déplacements relatifs en les expérimentant dans l'espace			■					
	Approcher la programmation de déplacements relatifs ou logique autocentrée				■				
	Savoir programmer des déplacements selon une logique autocentrée				■	■	■	■	
→	Comprendre la notion de boucle						■		
	Savoir programmer des déplacements avec des boucles							■	
→	S'engager dans une démarche de recherche							■	
	Approfondir la notion d'optimisation en programmation							■	
→	Comprendre les algorithmes simples : la notion d'événement « Si... alors... »								■

---

# Activités débranchées

---

# Activité 1 – Déplacer un objet sur un quadrillage

## OBJECTIFS

- Approcher la programmation de déplacements de type haut, bas, gauche, droite (déplacements absolus ou logique allocentrée).
- Comprendre la notion d'instruction, de programme, de code.



INDIVIDUEL  
RECHERCHE

### ÉTAPE 1

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 spatiale (le robot n'est pas orienté).

Insister sur le fait que le robot ne comprend que des **instructions précises** comme « Descends 2 fois, va 1 fois à gauche » et non pas « Rentre à ta base ».



COLLECTIF  
MISE EN  
COMMUN

### ÉTAPE 2

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 le valider.

En fonction des travaux réalisés, montrer ceux de quelques groupes. Ainsi les élèves vont remarquer que plusieurs chemins sont possibles et que chaque groupe a probablement utilisé des instructions différentes.

La restitution des groupes est l'occasion de développer le lexique sur la programmation.

Une suite d'instructions est un algorithme. La liste de tous les symboles est appelée un langage de programmation. Traduire l'algorithme dans ce langage permet d'écrire un programme.

On convient avec les élèves d'un langage simple et efficace : on code les déplacements par des flèches vers le haut, le bas, la droite, la gauche.

Un exemple est proposé au tableau avec les cartes de la **FICHE 1-a**.

D'autres situations peuvent être représentées en plaçant simplement le point de départ du robot et la base à différents endroits sur le quadrillage.



TRACE  
ÉCRITE

### ÉTAPE 3

La classe synthétise collectivement ce qui a été appris au cours de cette séance.

On dessinera par exemple la grille de déplacement avec le vaisseau, son point de départ et d'arrivée, et on écrira les instructions de déplacement.

# Activité 2 — Programmer des déplacements absolus

## OBJECTIFS

- Savoir programmer des déplacements selon la logique allocentrée.
- Comprendre la notion d'optimisation en programmation.



INDIVIDUEL  
RECHERCHE

### ÉTAPE 1

L'enseignant rappelle ce qui a été appris lors de la séance précédente : la manière dont on peut **donner des instructions** précises au robot, le fait que **plusieurs chemins** soient possibles, la notion de **programme** et de **code**.

L'enseignant demande ensuite de coder le déplacement de la navette à l'aide du langage établi précédemment sur la **FICHE 3** **en utilisant le moins possible de signes**. Le travail peut se faire individuellement ou en binômes.



COLLECTIF  
MISE EN  
COMMUN

### ÉTAPE 2

Lorsque les programmes sont écrits, un premier groupe (ou élève) présente à la classe son programme. Il est validé si dans la classe les programmes des autres élèves ne sont pas plus courts. Sinon on étudie une nouvelle proposition qui permet le déplacement avec moins de signes.

Le lexique s'enrichit. L'enseignant indique alors que cette recherche autour d'un programme le plus court possible (et par conséquent le plus rapide) s'appelle : **l'optimisation**.



INDIVIDUEL  
ENTRAÎNE-  
MENT

### ÉTAPE 3

Les élèves peuvent alors résoudre le problème de la **FICHE 4** : la navette doit se déplacer vers sa base en évitant les obstacles.



TRACE  
ÉCRITE

### ÉTAPE 4

La classe synthétise collectivement ce qui a été appris au cours de cette séance. On peut par exemple dessiner la grille de déplacement avec le vaisseau et écrire un programme « long » puis un programme optimisé.



PROLONGE-  
MENT

### ÉTAPE 5

Voir **ACTIVITÉ BRANCHÉE 1**.

## Activité 3 — Vivre les 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 représente une étape indispensable vers la programmation de déplacements des robots, qui utilise cette logique la plupart du temps.

### OBJECTIFS

- Comprendre les déplacements relatifs en les expérimentant dans l'espace.
- Savoir programmer des déplacements selon une logique autocentrée.



COLLECTIF  
ACTIVITÉ  
DÉCROCHÉE  
EN  
EXTÉRIEUR

### ÉTAPE 1

Cette séance consiste à faire vivre aux élèves une situation concrète de logique autocentrée.

- Chaque groupe d'élèves travaille sur un damier.
- L'enseignant demande à un élève de se placer aléatoirement sur le damier. Il représentera la « navette spatiale ».
- Les autres élèves du groupe définissent une case sur le damier qui représentera la base spatiale mais n'en informent pas la « navette spatiale ». Ils se mettent ensuite d'accord sur les instructions à donner à la « navette » pour qu'elle rejoigne sa base.
- L'enseignant demande à ce que les élèves écrivent l'intégralité du programme avant de commencer à donner les instructions à la « navette » et rappelle que cette dernière ne peut effectuer qu'une seule action à la fois : avancer, 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 navette » oralement, l'une après l'autre. Celui-ci les exécute au fur et à mesure qu'on les lui donne. À la fin de l'exécution du programme, « l'élève navette » doit être arrivé sur la case définie par ses camarades comme étant la base spatiale.
- L'exercice se répète jusqu'à ce que chaque élève ait eu au moins une fois le rôle de navette spatiale.
- Si certains groupes sont très à l'aise, l'enseignant peut leur proposer d'ajouter des obstacles et/ou d'agrandir le damier.

À partir de maintenant, les élèves peuvent travailler leurs compétences en utilisant le jeu Planète Code au niveau 1.



# Activité 4 – Découvrir les déplacements relatifs

## OBJECTIFS

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



INDIVIDUEL  
RECHERCHE

### ÉTAPE 1

L'objectif de cette étape est expliqué aux élèves. Comme pour les activités suivantes, il s'agit de programmer le déplacement de la navette mais avec une contrainte supplémentaire : la navette est déjà orientée, elle ne peut qu'avancer et tourner sur elle-même. Il va falloir en tenir compte et donc modifier les instructions par rapport aux étapes précédentes dans la [FICHE 5](#).



COLLECTIF  
MISE EN  
COMMUN

### ÉTAPE 2

Après quelques minutes de recherche, les élèves font des propositions pour coder l'instruction : « pivoter ». Pour aider les élèves à se représenter les déplacements, on pourra les faire rejouer en se mettant « à la place » de la navette.

L'enseignant pourra introduire le vocabulaire : déplacements relatifs ou logique autocentrée. En cherchant à optimiser les différents programmes, les élèves pourront remarquer que cette logique autocentrée nécessite un signe de moins que la précédente.

Une erreur d'instruction peut éloigner la navette de sa destination..

Il faudra trouver un consensus sur un code simple et efficace (privilégier les flèches avancer, pivoter à droite, pivoter à gauche) que tous les élèves utiliseront à partir de la séance suivante.

Un exemple est proposé au tableau avec les cartes de la [FICHE I-b](#).

L'enseignant peut préciser ces termes s'il le souhaite :

- *Déplacements absolus* ou *logique allocentrée* : ce sont les déplacements haut, bas, gauche, droite (Nord, Sud, Est, Ouest).
- *Déplacements relatifs* ou *logique autocentrée* : c'est une « vue à la première personne » que les élèves peuvent voir sur un GPS ou dans certains jeux vidéos. Les déplacements seront « aller tout droit », « pivoter à gauche », « pivoter à droite ».



TRACE  
ÉCRITE

### ÉTAPE 3

On peut montrer un exemple de codage allocentré et un exemple autocentré pour mettre en évidence la différence. On pourra utiliser les cartes du jeu Planète Code.

# Activité 5 — Programmer des déplacements relatifs

## OBJECTIFS

– Savoir programmer des déplacements selon une logique autocentrée.



INDIVIDUEL  
RECHERCHE

### ÉTAPE 1

L'enseignant revient avec les élèves sur les instructions qui permettent de faire déplacer la navette. La nouvelle activité de la **FICHE 6** va leur permettre de réutiliser le langage choisi pour un parcours avec des obstacles. La validation ou l'invalidation des programmes, proposés par les élèves, se fait collectivement.



INDIVIDUEL  
ENTRAÎNE-  
MENT

### ÉTAPE 2

On peut augmenter le nombre d'obstacles. Avec la **FICHE 7**, les élèves doivent programmer le déplacement de la navette jusqu'à sa base mais en évitant les 5 obstacles.

L'enseignant fait remarquer que 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 prévu.



TRACE  
ÉCRITE

### ÉTAPE 3

Pour effectuer la synthèse de cette activité, on peut reprendre la trace écrite de l'activité précédente et la compléter si nécessaire.



PROLONGE-  
MENT

### ÉTAPE 4

Voir **ACTIVITÉ BRANCHÉE 2**.

## Activité 6 – Découvrir 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'optimiser des lignes de code.

### OBJECTIFS

- Comprendre la notion de boucle.
- Savoir programmer des déplacements selon une logique autocentrée.



INDIVIDUEL  
RECHERCHE

### ÉTAPE 1

Cette nouvelle activité va permettre aux élèves de réutiliser le langage choisi dans l'activité précédente et chercher à l'optimiser en introduisant une boucle. La [FICHE 9](#) propose une situation avec des instructions répétitives. Les élèves répartis par groupes programment le déplacement de la navette.



COLLECTIF  
MISE EN  
COMMUN

### ÉTAPE 2

La mise en commun des programmes donne l'occasion à l'enseignant de faire observer les **séquences répétitives d'instructions**. Il introduit le langage de boucle et les élèves l'appliquent à leur propre programme lorsque c'est possible. La validation est collective.

Ci-dessous, une façon d'écrire ce programme. Il est possible aussi d'utiliser, à la place des parenthèses, le début et la fin de boucle de la [FICHE 8](#).

```
[ ↑ → ↑ ← ] x 6
```

D'autres situations peuvent être représentées en plaçant le point de départ de la navette et la base à différents endroits sur le quadrillage afin d'exploiter d'autres déplacements avec des boucles.



TRACE  
ÉCRITE

### ÉTAPE 3

Pour synthétiser la nouvelle notion de boucle, les élèves proposent, pour un déplacement de la navette, un programme avec des instructions répétitives et un second avec une boucle qui réduit le nombre d'instructions du premier programme.

# Activité 7 – Programmer des boucles

## OBJECTIFS

- Savoir programmer des déplacements avec des boucles.
- S’engager dans une démarche de recherche.
- Approfondir la notion d’optimisation en programmation.
- Savoir programmer des déplacements selon une logique autocalibrée.



INDIVIDUEL  
RECHERCHE

### ÉTAPE 1

Un bref rappel de la synthèse précédente va permettre aux élèves de retrouver le langage des boucles. Un nouveau déplacement à programmer est proposé aux élèves avec la **FICHE 10**. Contrairement à la **FICHE 9** plusieurs propositions de boucles sont possibles.



COLLECTIF  
MISE EN  
COMMUN

### ÉTAPE 2

Les différentes propositions sont validées ou invalidées collectivement. Voici deux propositions possibles, la première étant plus courte que la seconde.



INDIVIDUEL  
ENTRAÎNE-  
MENT

### ÉTAPE 3

Deux nouvelles grilles sont proposées aux élèves avec la **FICHE 11** et la **FICHE 12**. Les élèves vont découvrir que le programme de déplacements de la navette comporte une boucle ainsi que d’autres instructions indépendantes de celle-ci.

Les programmes validés seront ceux qui correspondent à la consigne. Le nombre d’instructions ne devra pas dépasser le nombre de cases du bandeau prévu pour le programme. Le programme le plus court pour la **FICHE 11** est :



PROLONGE-  
MENT

Voir **ACTIVITÉ BRANCHÉE 3**

À partir de maintenant, les élèves peuvent travailler leurs compétences en utilisant le jeu Planète Code au niveau 2.

## Activité 8 — Les événements

### OBJECTIFS

– Comprendre les algorithmes simples : la notion d'événement « Si... alors... ».



INDIVIDUEL  
RECHERCHE

### ÉTAPE 1

L'enseignant propose aux élèves un problème plus complexe. La direction de la navette étant bloquée, celle-ci n'avance que tout droit. Les élèves doivent donc programmer le changement de direction de la navette (pivoter à droite ou à gauche) lorsqu'elle survole une case de couleur, dans le but de rejoindre sa base. Pour cela, ils doivent écrire en toutes lettres et en français les instructions pour la navette sur le modèle du « Si... alors... »

Exemple : « Si la navette survole une case bleue, alors elle pivote à droite. »

Avec la [FICHE I3](#), ils peuvent faire leur recherche en binôme ou individuellement.

Cette notion d'événement est très présente dans l'univers de la programmation, dans tous les domaines. On propose ici aux élèves de travailler en « pseudo-code » : donner des instructions en langage naturel.



COLLECTIF  
MISE EN  
COMMUN

### ÉTAPE 2

Les élèves font des propositions d'instructions conditionnelles qu'ils ont trouvées. La validation est collective.



INDIVIDUEL  
ENTRAÎNE-  
MENT

### ÉTAPE 3

Les élèves peuvent maintenant programmer le survol des cases bleues, rouges et vertes dans la [FICHE I4](#).

À partir de maintenant, les élèves peuvent travailler leurs compétences en utilisant le jeu Planète Code au niveau 3.

---

# Activités branchées

---

---

# Activités branchées

---

Des séances de prolongement « branchées » peuvent être effectuées avec le logiciel ou l'application tablette gratuite « Tuxbot », développée par les conseillers du département de la Mayenne : <http://appli-etna.ac-nantes.fr:8080/ia53/tice/ressources/tuxbot/index.php>

## ACTIVITÉ 1

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.

## ACTIVITÉ 2

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.

## ACTIVITÉ 3

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 rapidement comprise par les élèves s'ils ont déjà intégré la notion de 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.

---

# Fiches d'activité

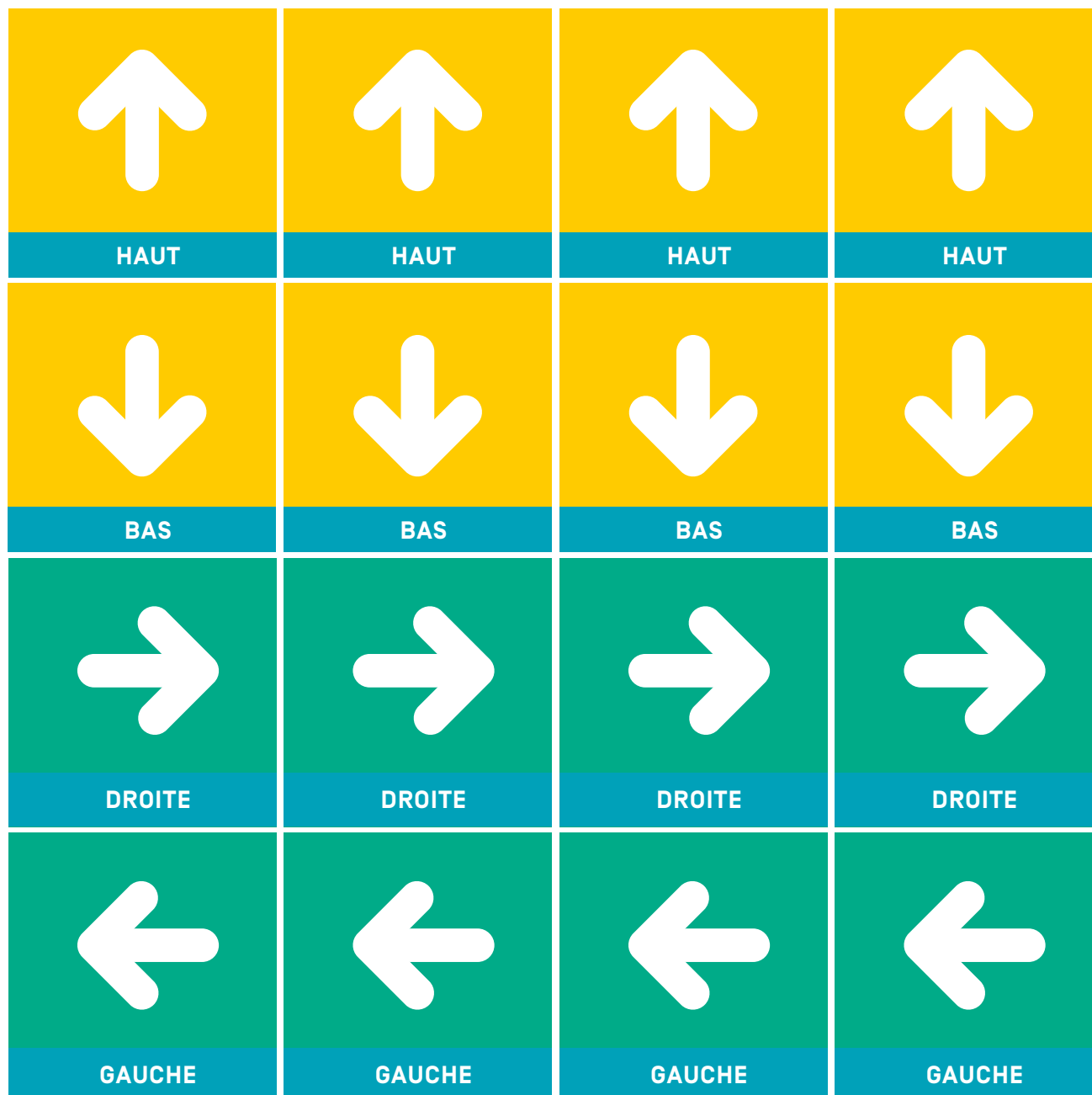
---



---

## Fiche 1-a

















---



---

## Fiche 1-b

---

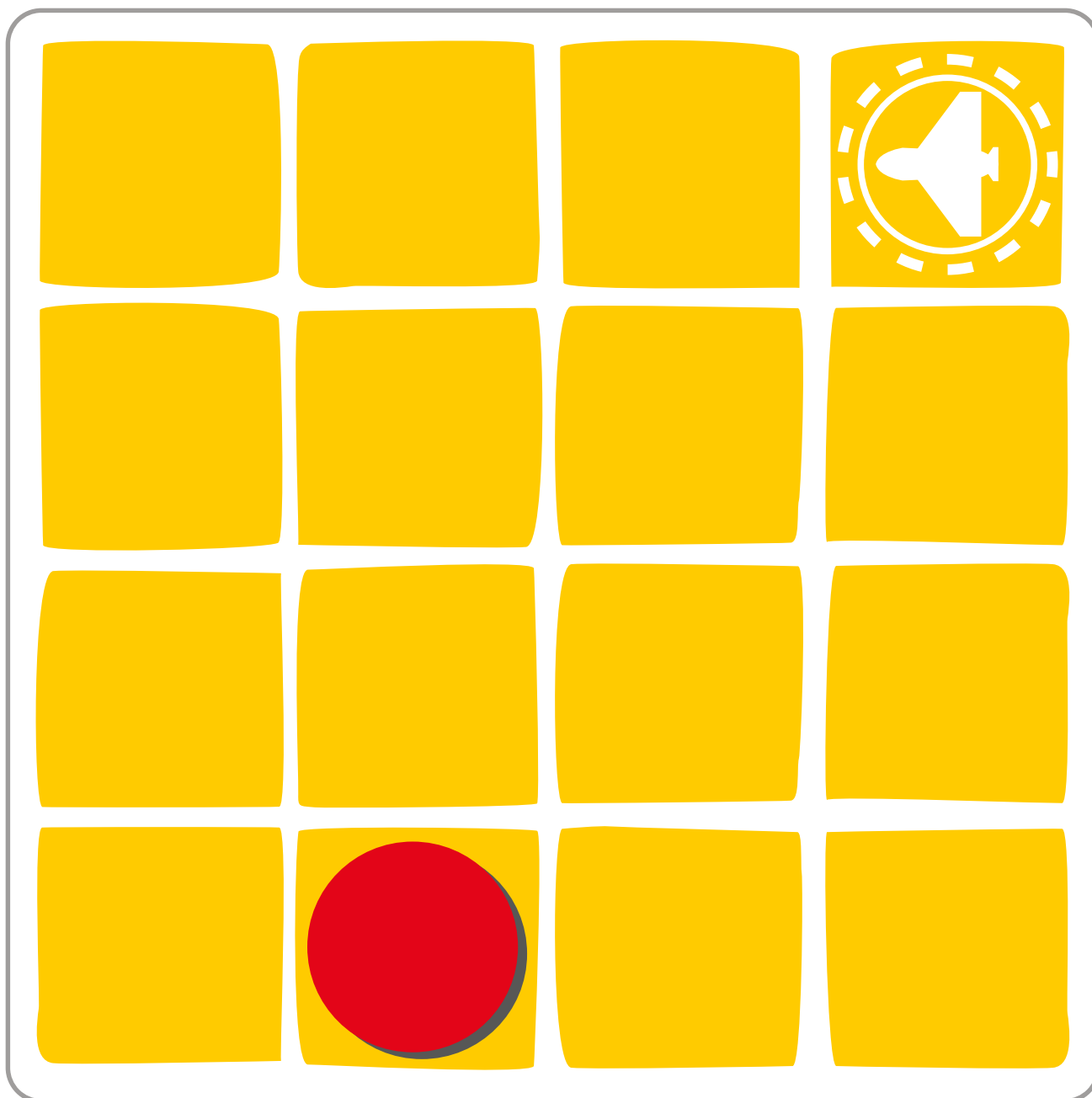
			
AVANCER	AVANCER	AVANCER	AVANCER
			
AVANCER	AVANCER	AVANCER	AVANCER
			
PIVOTER À DROITE	PIVOTER À DROITE	PIVOTER À DROITE	PIVOTER À DROITE
			
PIVOTER À GAUCHE	PIVOTER À GAUCHE	PIVOTER À GAUCHE	PIVOTER À GAUCHE

---

## Fiche 2

---

Donne des instructions pour déplacer le robot, symbolisé par un pion, jusqu'à la base.  
Il ne peut se déplacer que verticalement ou horizontalement.

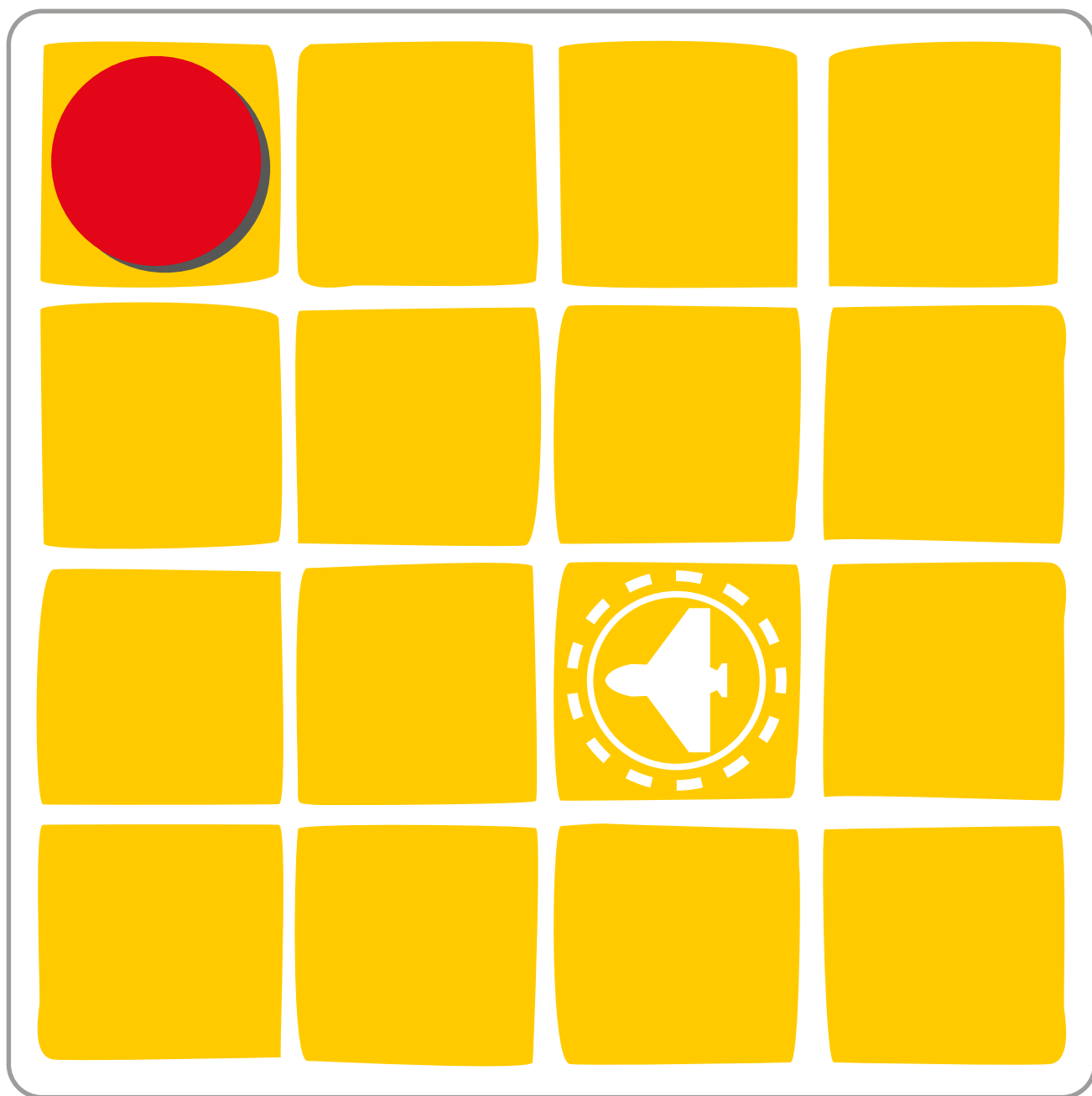


---

## Fiche 3

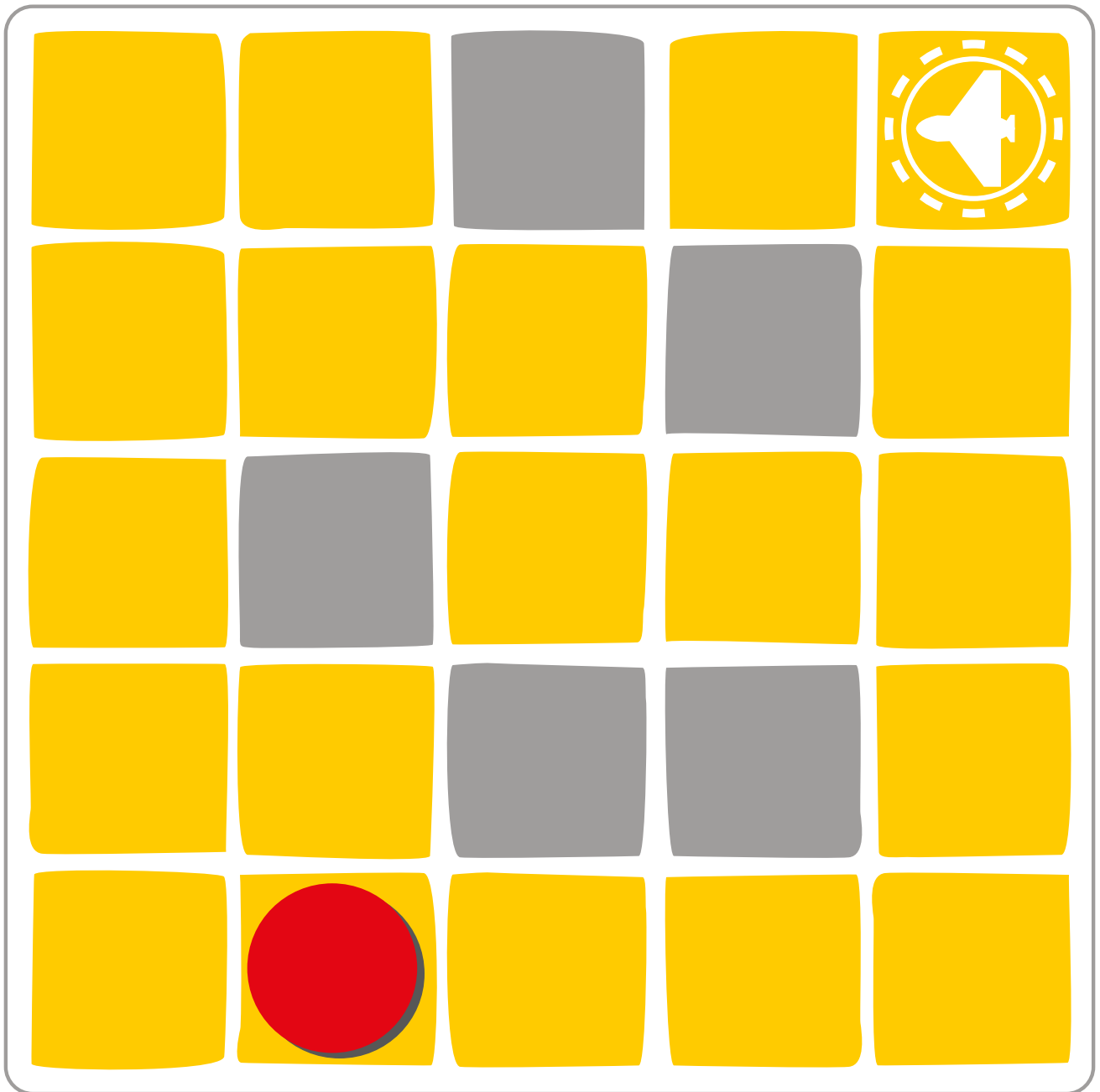
---

Donne des instructions pour déplacer le robot, symbolisé par un pion, jusqu'à la base, le plus vite possible. Il ne peut se déplacer que verticalement ou horizontalement.



# Fiche 4

Écris dans les cases ci-dessous les instructions pour déplacer le robot, symbolisé par un pion, jusqu'à la base, le plus vite possible. Il ne peut se déplacer que verticalement ou horizontalement. Attention, il faut éviter les obstacles (cases grises).



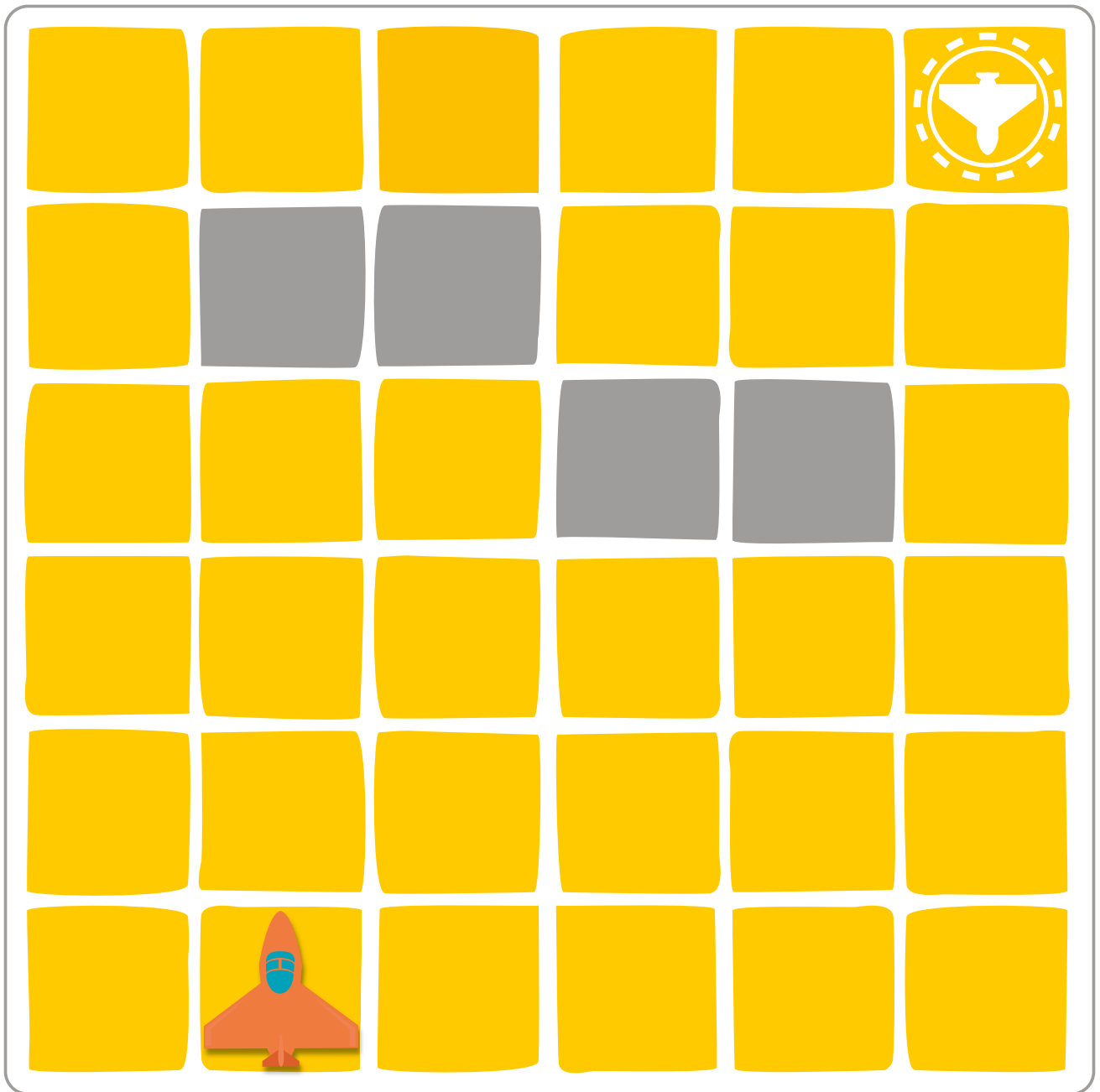
# Fiche 5

La batterie de la navette spatiale est faible. Écris dans le tableau ci-dessous les instructions pour qu'elle rejoigne la base le plus vite possible. Attention, cette navette ne peut se déplacer qu'en avant et ne tourner que sur elle-même. Tu dois en tenir compte dans tes instructions.



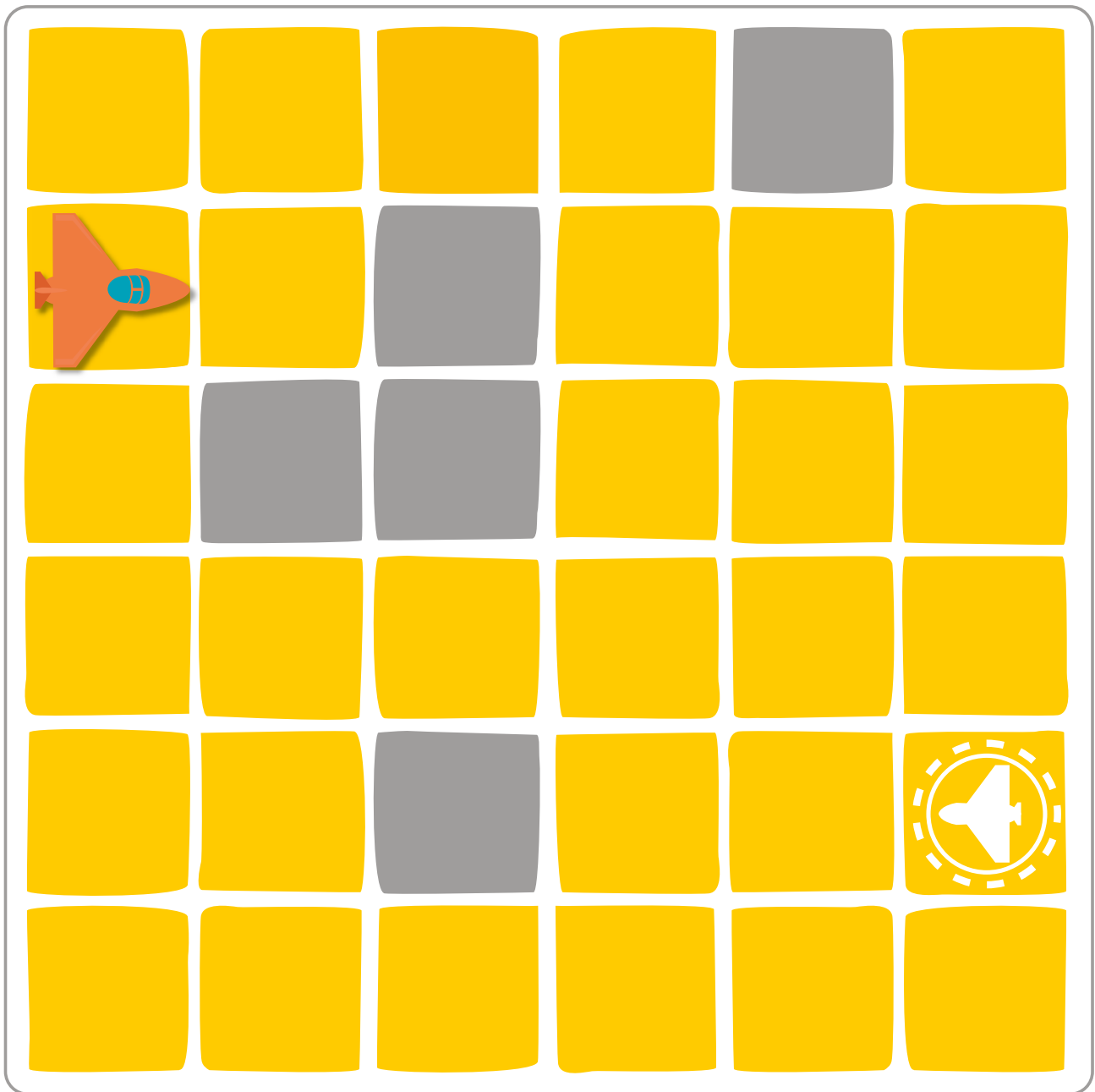

# Fiche 6

La batterie de la navette spatiale est faible. Écris dans le tableau ci-dessous les instructions pour qu'elle rejoigne la base le plus vite possible. Attention aux obstacles et rappelle-toi que cette navette ne peut se déplacer qu'en avant et ne tourner que sur elle-même. Tu dois en tenir compte dans tes instructions.



# Fiche 7

La batterie de la navette spatiale est faible. Écris dans le tableau ci-dessous les instructions pour qu'elle rejoigne la base le plus vite possible. Attention aux obstacles et rappelle-toi que cette navette ne peut se déplacer qu'en avant et ne tourner que sur elle-même. Tu dois en tenir compte dans tes instructions.









---

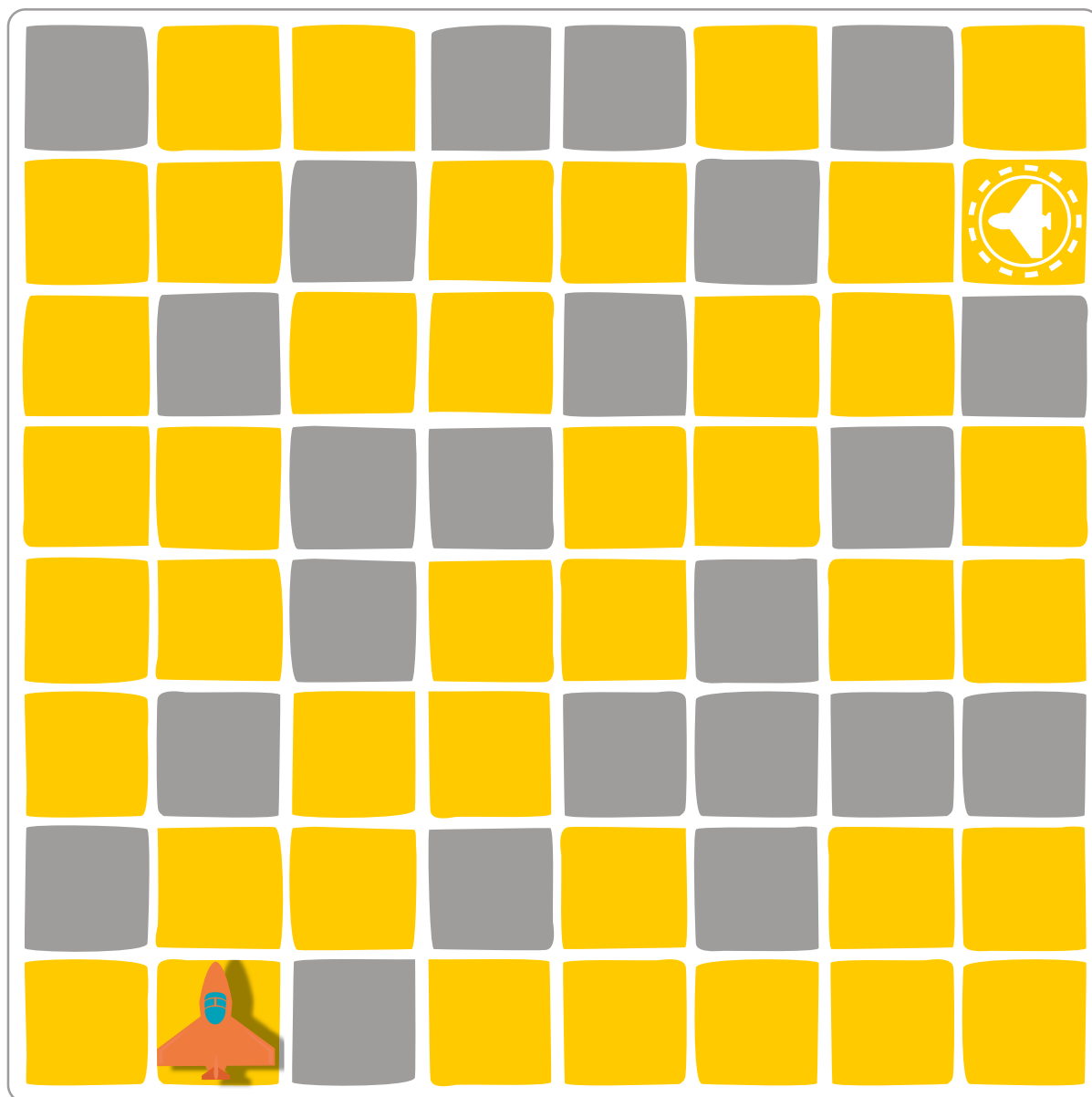
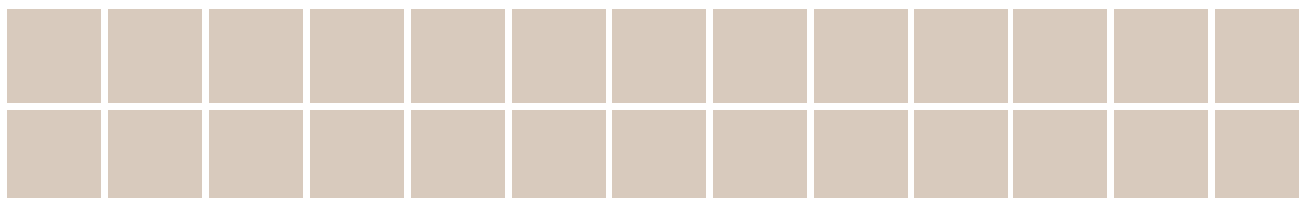
## Fiche 8

---

 DÉBUT DE BOUCLE	 DÉBUT DE BOUCLE	 FIN DE BOUCLE	 FIN DE BOUCLE
x2	x3	x4	x5
x6	x7	x8	x9

# Fiche 9

La batterie de la navette spatiale est faible. Écris dans le tableau ci-dessous les instructions pour qu'elle rejoigne la base le plus vite possible.

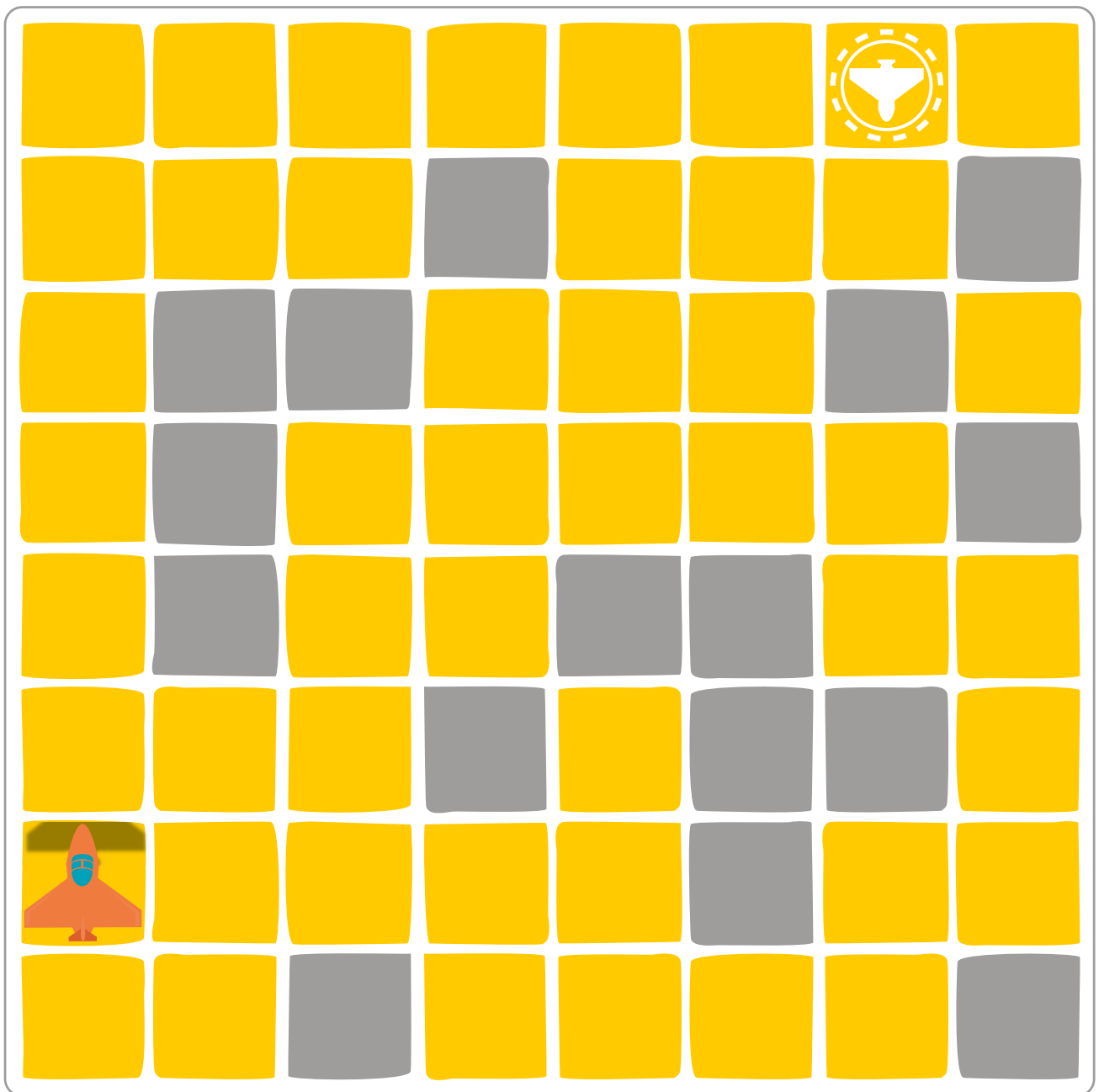


---

# Fiche 10

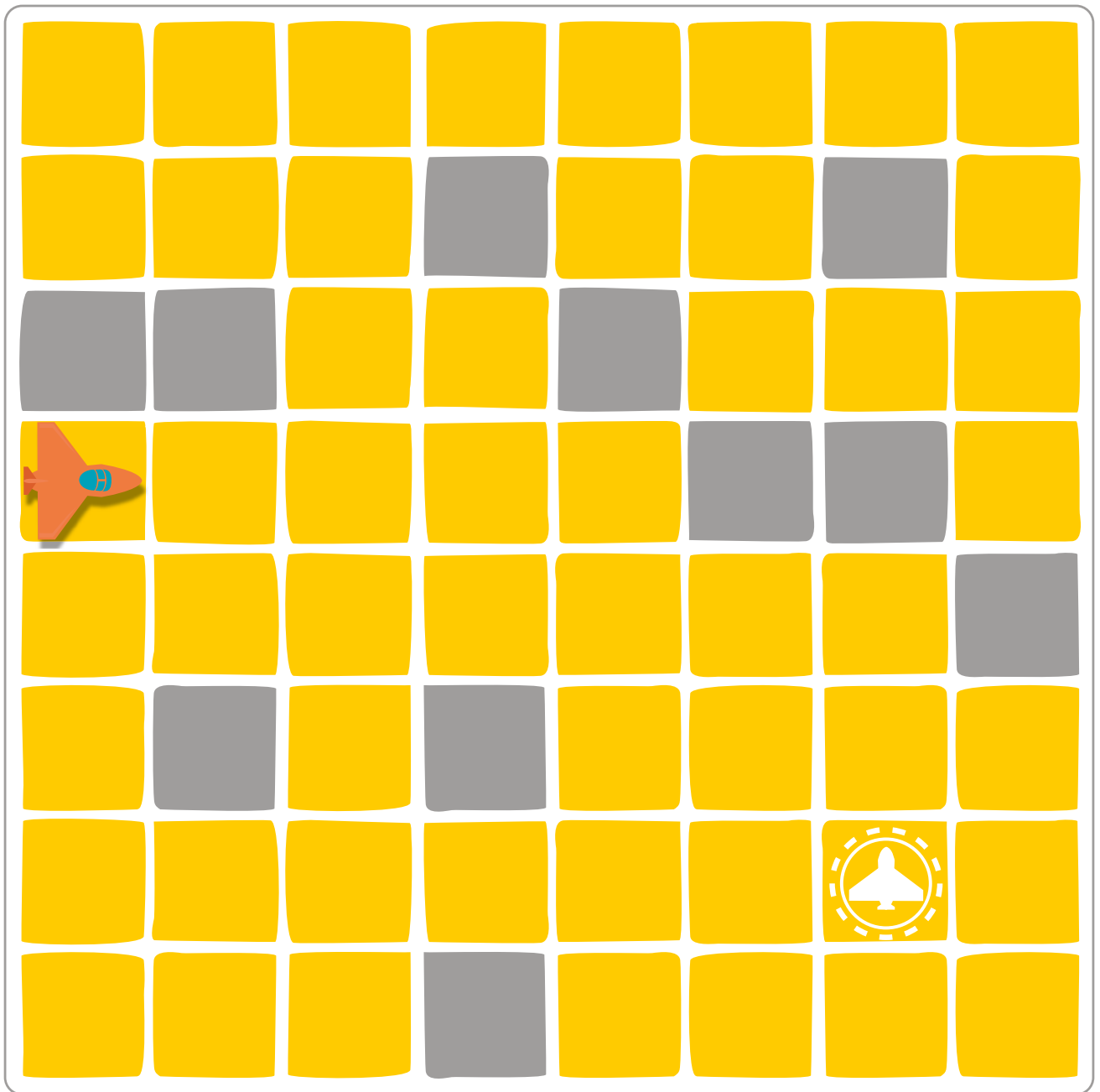
---

La batterie de la navette spatiale est faible. Écris dans le tableau ci-dessous les instructions pour qu'elle rejoigne la base le plus vite possible.  
Attention, tu ne peux pas donner plus d'instructions qu'il n'y a de cases dans le bandeau !



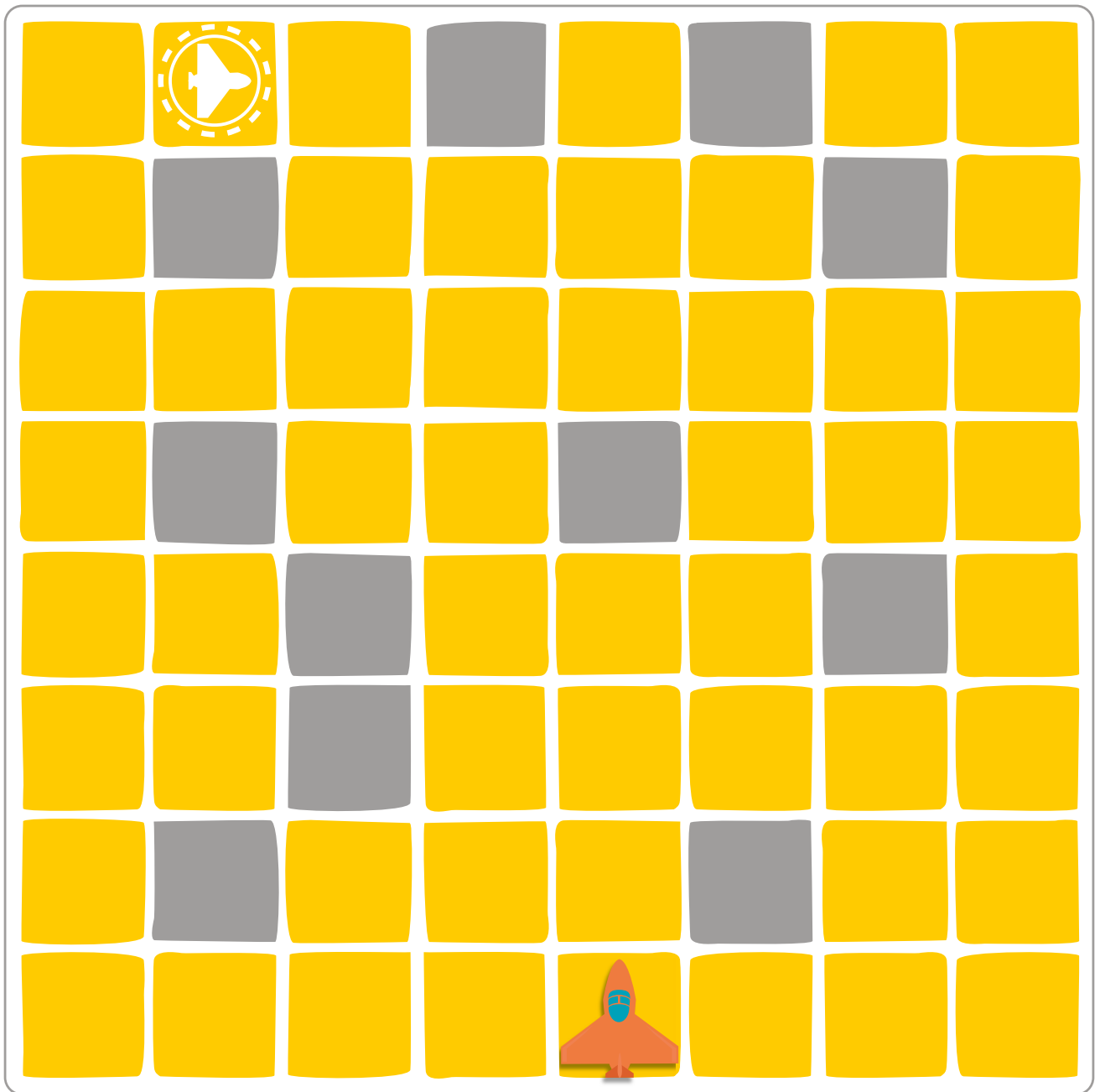
# Fiche 11

La batterie de la navette spatiale est faible. Écris dans le tableau ci-dessous les instructions pour qu'elle rejoigne la base le plus vite possible.  
Attention, tu ne peux pas donner plus d'instructions qu'il n'y a de cases dans le bandeau !



# Fiche 12

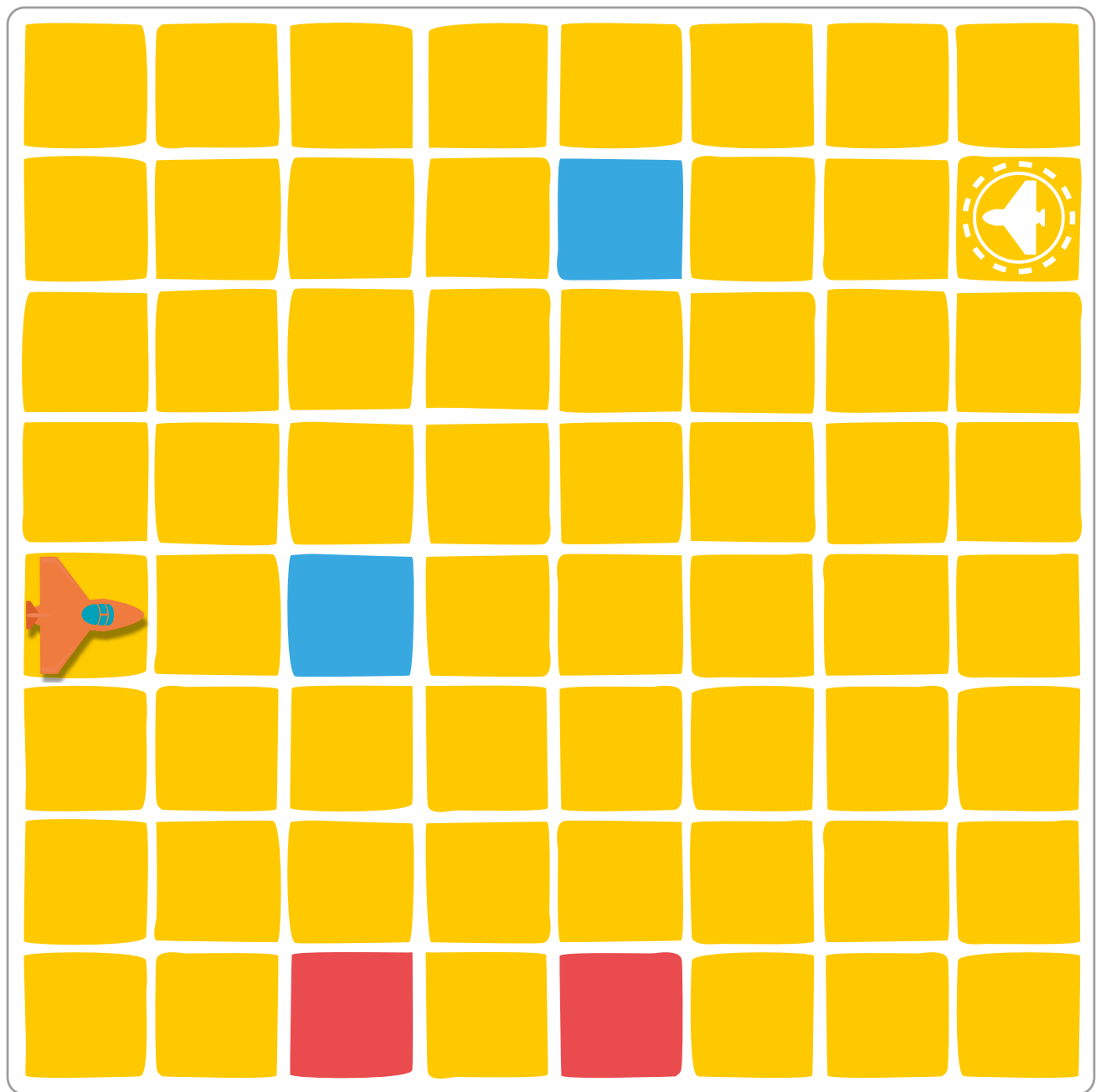
La batterie de la navette spatiale est faible. Écris dans le tableau ci-dessous les instructions pour qu'elle rejoigne la base le plus vite possible.  
Attention, tu ne peux pas donner plus d'instructions qu'il n'y a de cases dans le bandeau !



# Fiche 13

La navette veut rejoindre la base mais sa direction est bloquée et elle ne peut qu'avancer tout droit. Complète les phrases pour programmer le changement de direction de la navette (pivoter à droite ou à gauche) lorsqu'elle survole une case de couleur.

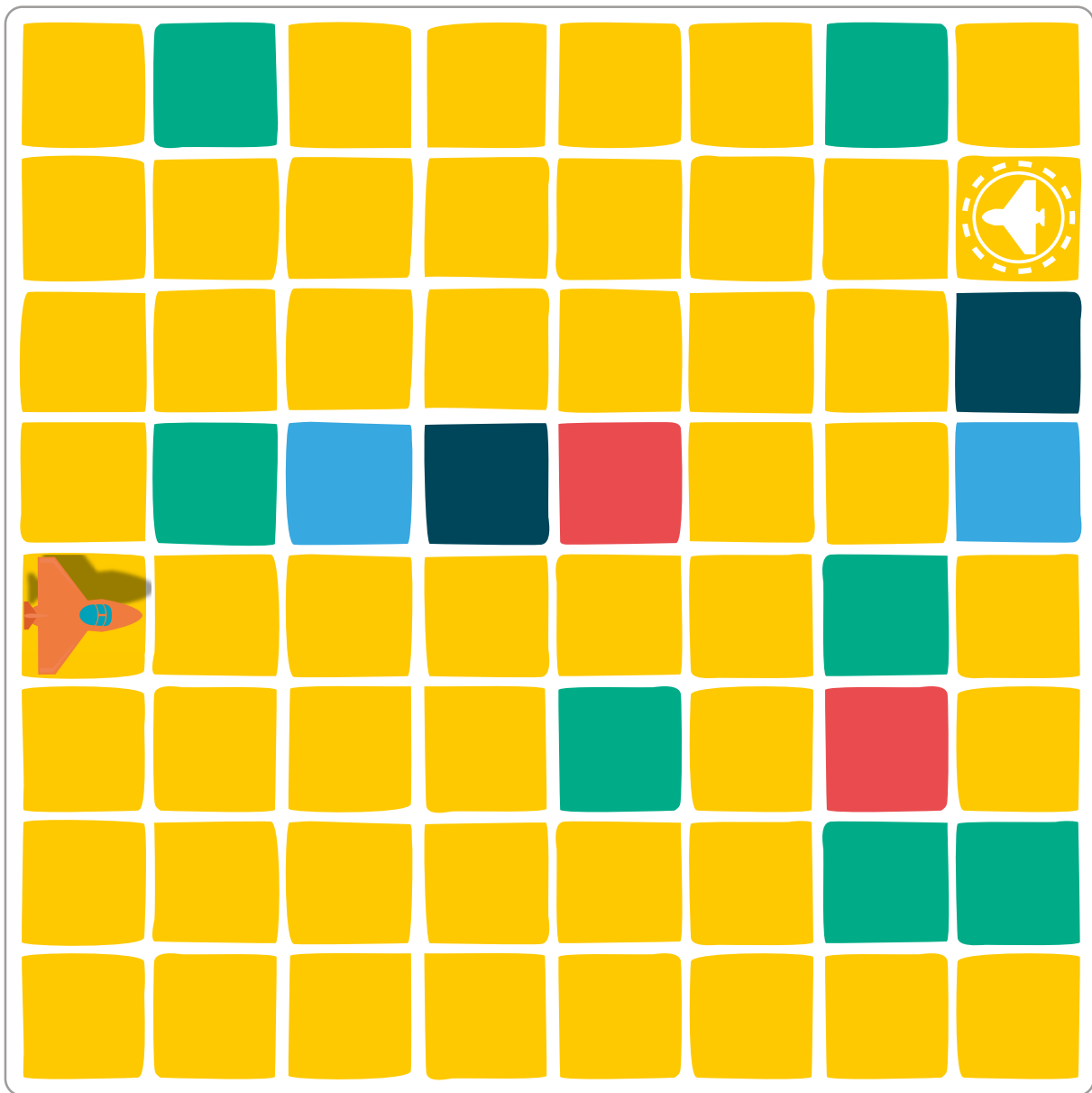
Si la navette survole une case _____	alors la navette _____
Si la navette survole une case _____	alors la navette _____



# Fiche 14

La navette veut rejoindre la base mais sa direction est bloquée et elle ne peut qu'avancer tout droit. Complète les phrases pour programmer le changement de direction de la navette (pivoter à droite ou à gauche) lorsqu'elle survole une case de couleur.

Si la navette survole une case _____	alors la navette _____
Si la navette survole une case _____	alors la navette _____
Si la navette survole une case _____	alors la navette _____



---

# Fiche 15

---

Grille vierge à utiliser pour construire votre propre activité.