

FMTTN P5-P6 Algorithme, logigramme et programmation

Une activité déconnectée, un logigramme et un premier programme avec Scratch

Objectifs

- Découvrir la programmation à l'aide d'une activité déconnectée
- Introduire les notions d'algorithme et de programme
- Réaliser un logigramme
- Faire le lien avec des compétences mathématiques.

Les attendus du référentiel repris dans le programme sont rencontrés.

Compétences exercées

Programmation et algorithmique:

- Les déplacements, la rotation et les boucles
- Le logigramme

Mathématiques :

- La construction de polygones réguliers à l'aide des angles externes
- Multiplication et division
- Les déplacements sur une grille

1 Activité déconnectée / logigramme

1 période maximum (20 minutes suffisent)

En fonction du matériel ou des locaux dont dispose l'enseignant.e, on utilise n'importe quel damier sur lequel on place des objets qui représentent des obstacles et des couloirs par lesquels le robot pourra circuler pour aller d'un point à l'autre.

On peut donc travailler en « live » avec des élèves qui se déplacent sur des dalles ou grands carrelages ou avec des pions sur des damiers de jeux d'échecs.

Il va de soi que le déroulé proposé ci-dessous est juste... une proposition. Au niveau d'une 5P, il y a d'ailleurs de fortes chances que ce genre d'exercice (un déplacement sur une grille) a déjà été effectué... Dans ce cas, on pourra réaliser cette activité très rapidement et au titre de rappel.

L'objectif de cette activité est de faire comprendre à l'élève que pour assurer la réussite des missions, il faut que les instructions soient dictées de façon claire, univoque et que celui/celle qui les exécute les comprenne et les applique fidèlement.

1.1 Proposition de déroulement de l'activité

Matériel

- Deux ou trois modèles de grilles pour les missions .
- une grille vierge sur un A3
- Des petites cartes représentant des mouvements simples (voir annexe) et le vaisseau qui effectuera le déplacement. Les petites cartes peuvent être remplacées par un pion ou n'importe quel petit objet.

Acte 1 : le capitaine et son pilote.

Missions :

- ▼ Atteindre une case « destination » en un minimum d'instructions : mission 1 et 2 (voir annexe)
- ▼ Effectuer une trajectoire carrée et revenir à la position de départ

Désigner deux élèves, un capitaine et un pilote. Le capitaine et le pilote sont dos à dos.

Le pilote

- dispose d'une grille vierge et d'une petite carte qui représente son vaisseau
- il effectue les déplacements sur base des instructions du capitaine
- Il ne peut pas parler, juste dire OK quand il a réalisé l'instruction

Le capitaine

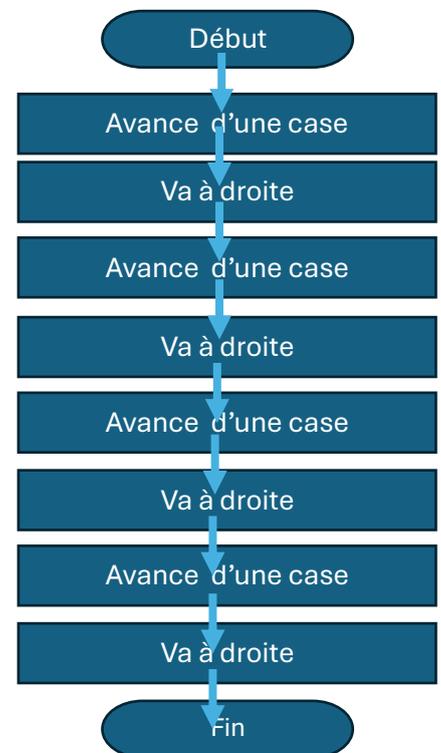
- dit « Départ » quand il commence à dicter ses instructions et « Terminé » quand il a fini.
- Dicte ses instructions comme bon lui semble

Pendant la mission, une personne/l'enseignant note sur une feuille ou au tableau les instructions sous forme d'un logigramme.

Par exemple →:

Quand la mission est terminée, le capitaine se retourne et on commente ensemble ce qu'il s'est passé.

- ✓ Est-ce que la mission s'est bien passée ?
- ✓ Si non, pourquoi ? Si oui, pourquoi ?
- ✓ Qu'est-ce qu'il faudrait faire pour que la mission se passe mieux ?



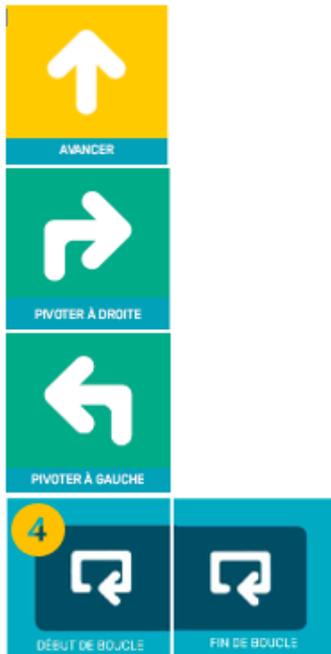
L'ensemble des instructions qui ont servi à l'exécution de la mission peut être appelé « **algorithme** » et le **logigramme** est une représentation graphique de l'algorithme.

Autres exemples d'algorithmes (interroger les élèves):

- Une recette de cuisine
- Le raisonnement que je tiens quand je choisis mes vêtements avant de partir à l'école
- La feuille de route d'un pilote de rallye
- Beaucoup plus compliqué : la description des étapes à mettre en œuvre pour faire décoller une fusée

Acte 2 : le capitaine et le robot

On reprend le même processus et les mêmes missions mais cette fois, le capitaine prépare la mission en utilisant les cartes qu'il dispose en ligne sur la table. Quand il a terminé son programme, le robot se retourne et effectue la mission. Le capitaine ne peut intervenir pendant l'exécution du programme.



Ces cartes sont les seules instructions que le robot comprend. Réaliser une ou deux missions avec les trois premières cartes et essayer ensuite avec les boucles.

AVANCER : avancer d'une case

PIVOTER à DROITE : pivoter de 90° vers la droite

PIVOTER à GAUCHE : pivoter de 90° vers la gauche

DEBUT DE BOUCLE 4 :

répéter 4 fois une opération

FIN DE BOUCLE

Quand une mission est terminée, on se pose ces questions :

✓ Est-ce que la mission s'est bien passée ?

✓ Si oui/non, pourquoi ?

✓ Comment peut-on réduire le nombre d'instructions grâce aux boucles ?

✓ Est-ce que ce processus est plus fiable que le précédent ?

Pour une trajectoire carrée, le vaisseau doit exécuter ce programme :



Ou, avec une boucle :



L'ensemble des instructions envoyées à un ordinateur (le robot) est appelé un **programme**. Les instructions que l'ordinateur comprend sont appelées un **langage de programmation** ou encore du **code**.

2 Introduction à l'utilisation de Scratch et des cartes de programmation

2.1 Modalités pratiques :

- Durée : deux périodes
- Pour ce premier exercice, **il n'est pas nécessaire de créer un compte Scratch** pour les élèves. Il faudra en revanche leur montrer comment renommer et enregistrer leur projet sur l'ordinateur.
- Idéalement **un pc et un jeu de carte par élève**. Un PC et un jeu de cartes pour deux peuvent aussi faire l'affaire.
- Les premiers projets étant enregistrés (et non dans le Cloud), veiller à ce que les élèves utilisent **le même ordinateur pendant toute la séquence**.

On présume qu'il s'agit d'un premier contact avec Scratch pour les élèves. Si ce n'est pas le cas, passer directement à l'étape suivante.

2.2 Présentation de Scratch :

- L'enseignant.e présente l'application et fait quelques démonstrations. L'enseignant laisse ensuite un peu de temps aux élèves pour prendre la main et faire quelques essais. Le mieux est de leur proposer des petits défis très simples :
 - Place le sprite sur le bord gauche. Trouve et teste un bloc avec lequel le sprite va traverser en deux secondes la scène de gauche à droite.
 - Trouve et teste un bloc qui change la couleur de ton sprite.
 - Trouve et teste un bloc qui...

Si l'enseignant ne se sent pas à l'aise pour cette présentation, il peut inviter les élèves à utiliser le tuto « Prise en main » et si on a du temps, éventuellement laisser les élèves suivre le tuto « Anime un nom » <https://scratch.mit.edu/ideas>

2.3 Utilisation des cartes de programmation

Expliquer le principe :

- Une carte défi et au verso, les consignes.
- On peut créer, proposer des variantes à une condition : avoir fini tout le jeu de cartes.

3 Les Cartes : « Des polygones de tous les côtés »

Le programme complet: <https://scratch.mit.edu/projects/1072097200/editor/>

Quelques remarques préliminaires

Cette séquence couvre bien **plus que les attendus du référentiel**, et c'est voulu. Pour couvrir les attendus du référentiel, on pourrait s'arrêter à la carte 8. Je déconseille formellement vous s'arrêter là. Il faut laisser les élèves aller le plus loin possible et essayer des variantes au programme, jouer avec les formes et les couleurs. Et pour cela, amener tous ceux qui le pourront jusqu'aux dernières cartes.

Profitions des exercices de programmation pour **exercer aussi des compétences mathématiques** et donc donner du sens aux notions vues dans ce cours, voire à celles qui seront vues plus tard. Ici, les élèves vont apprendre en pratiquant comme le disent les anglo-saxons : « Learning by doing ». « Apprendre en faisant » veut aussi dire que l'enseignant doit accepter que **l'élève ne va pas tout comprendre tout de suite**. Les élèves doivent apprendre un maximum par eux-mêmes et en expérimentant. Vouloir tout expliquer tout de suite n'est pas une bonne stratégie.

3.1 Cartes 3 & 4 : prépare un sprite et un arrière-plan

Rien à signaler.

3.2 Cartes 5 & 6 : Quelques réglages

Ici, on s'intéresse à deux propriétés du sprite : sa position sur la scène que l'on peut modifier en la déplaçant avec la souris ou en modifiant la valeur de x et y dans la fenêtre des **propriétés**.



Le terme **propriété** est important en programmation. Dans ce cas, les propriétés d'un objet (= sprite) sont :

- son nom
- sa position horizontale (x = abscisse) et verticale (y = ordonnée)
- sa visibilité (oui/non)
- sa taille en % de la taille du costume
- sa direction en degrés

Le jeune se familiarisera avec ces propriétés en pratiquant.

À ce stade, l'élève est invité à placer Beetle au milieu de la scène en tapant 0 et 0 dans les champs X et Y et à le diriger vers la droite en tapant 90 dans le champ Direction (si nécessaire..)

3.3 Cartes 7 & 8 : Beetle effectue une trajectoire carrée

On entre dans le vif du sujet. L'élève essaie d'assembler les blocs proposés, teste, duplique et doit obtenir la trajectoire carrée et un retour à la position de départ.

La pause de 1 seconde ou d'une demi-seconde (0.5, le séparateur décimal est le point) est indispensable. Sans elle, le mouvement est trop rapide et on ne le voit pas.



Fin de page 8. Une petite procédure est suggérée pour **l'initialisation**. Il est important que l'élève comprenne que dans un programme, il est souvent utile de commencer par positionner l'objet au point de départ de l'action. Ici, très concrètement, cela veut dire remettre l'objet au centre de la scène et le tourner dans la bonne direction.

3.4 Cartes 9 & 10, deux défis : « Moins d'instructions SVP ! » et « Laisse une trace »

On invite l'élève à utiliser une boucle avec le bloc **répéter... fois**. Cette étape est cruciale. L'enseignant.e devrait vérifier que l'élève a bien compris l'intérêt de passer du premier code au deuxième



Les avantages du programme comprenant une boucle sur le prédécesseur :

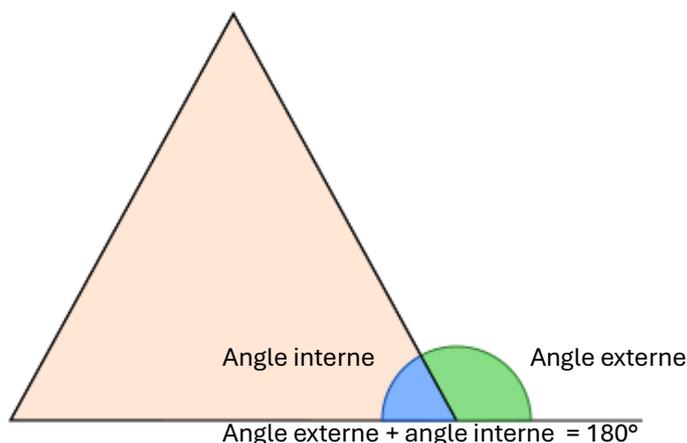
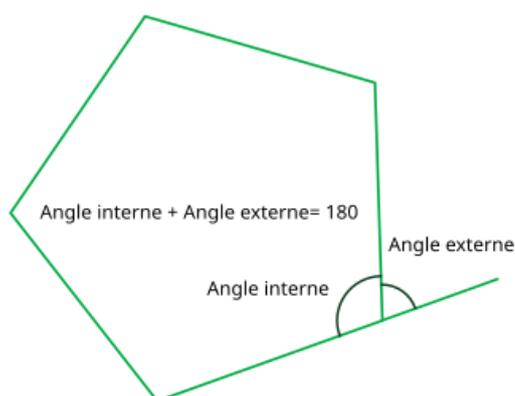
- Le code est plus lisible, moins long
- On limite le risque d'erreur
- Si on doit changer une valeur (par exemple le nombre de pas) , on ne doit le faire qu'une fois.

Etape suivante : ajouter une nouvelle famille de blocs : **Stylo** pour tracer des lignes lors du déplacement.

3.5 Cartes 11 & 12 : un triangle

Il s'agit d'une étape importante. À ce stade, il faut laisser le temps aux élèves de réfléchir à la question suivante. **Comment adapter le programme qui dessine un carré de façon à obtenir un triangle ?** Il n'y a en principe aucun bloc à ajouter ou à retirer. Il faut juste modifier certaines valeurs : le nombre de répétition (= nombre de côtés du polygone) et le nombre de degrés de rotation.

Un raisonnement mathématique est esquissé sur la carte 11. C'est la notion d'angle externe dans un polygone et est utilisée ici. Un schéma est proposé sur la carte 12. Le principe que l'élève devra retenir est : **la somme des angles externes dans un polygone régulier est toujours égale à 360°**.

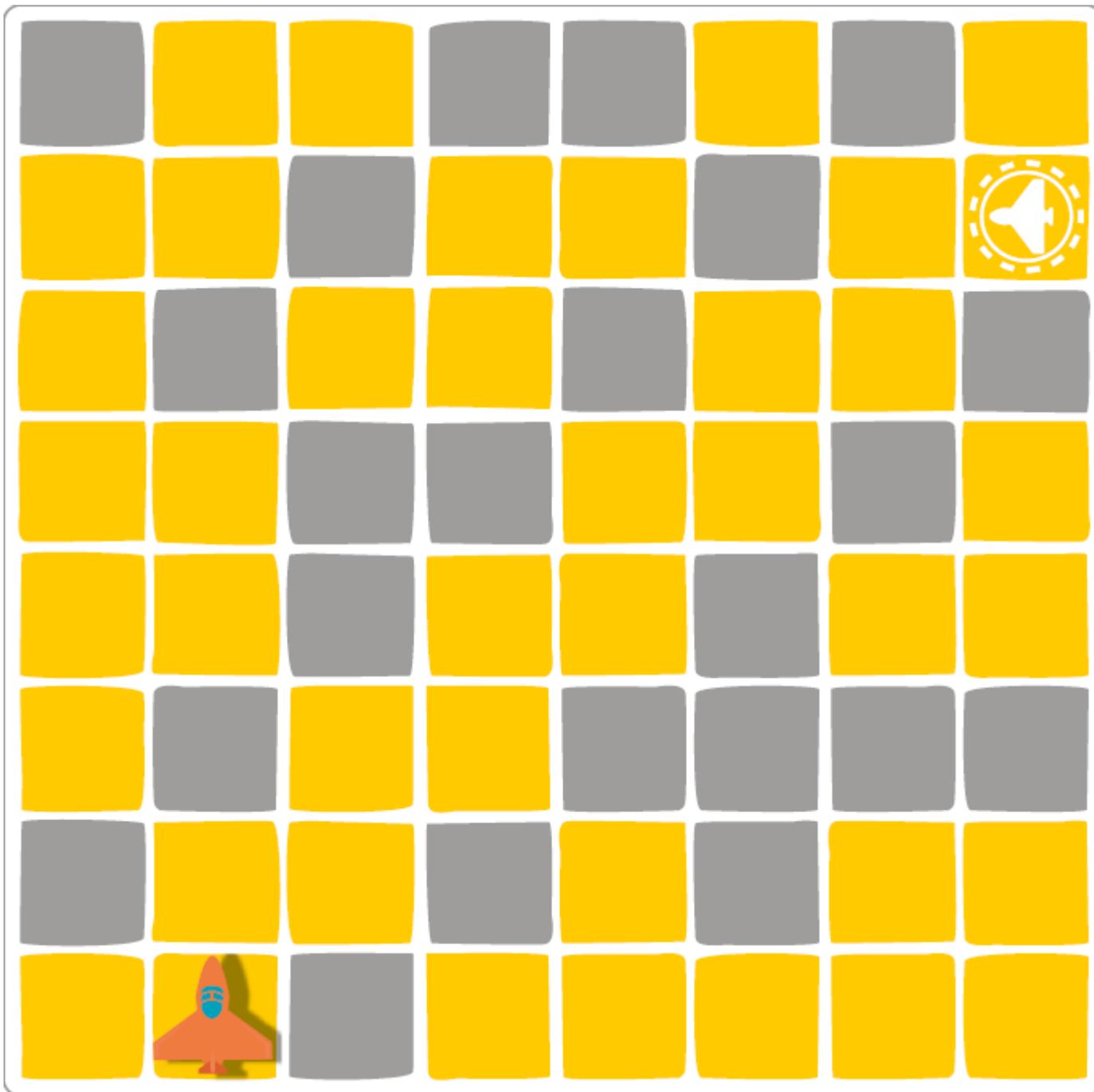


3.6 Cartes 13 & 14 : des rosaces

Dans l'exemple proposé, on reproduit 10 octogones en pivotant à chaque occurrence de $360/10^\circ = 36^\circ$ pour obtenir une rosace complète. Ici aussi, on laissera du temps aux élèves pour multiplier les polygones et dessiner des rosaces, jouer avec les couleurs, déplacer les figures etc...

Mission 1

Attendre l'arrivée en utilisant le moins possible d'instructions et sans toucher les cases grises.



Niveau de la batterie du vaisseau. Chaque case représente une carte. Utilise le moins de cartes possible ! Le vaisseau doit arriver sur la case et dans la position indiquée.

Départ :

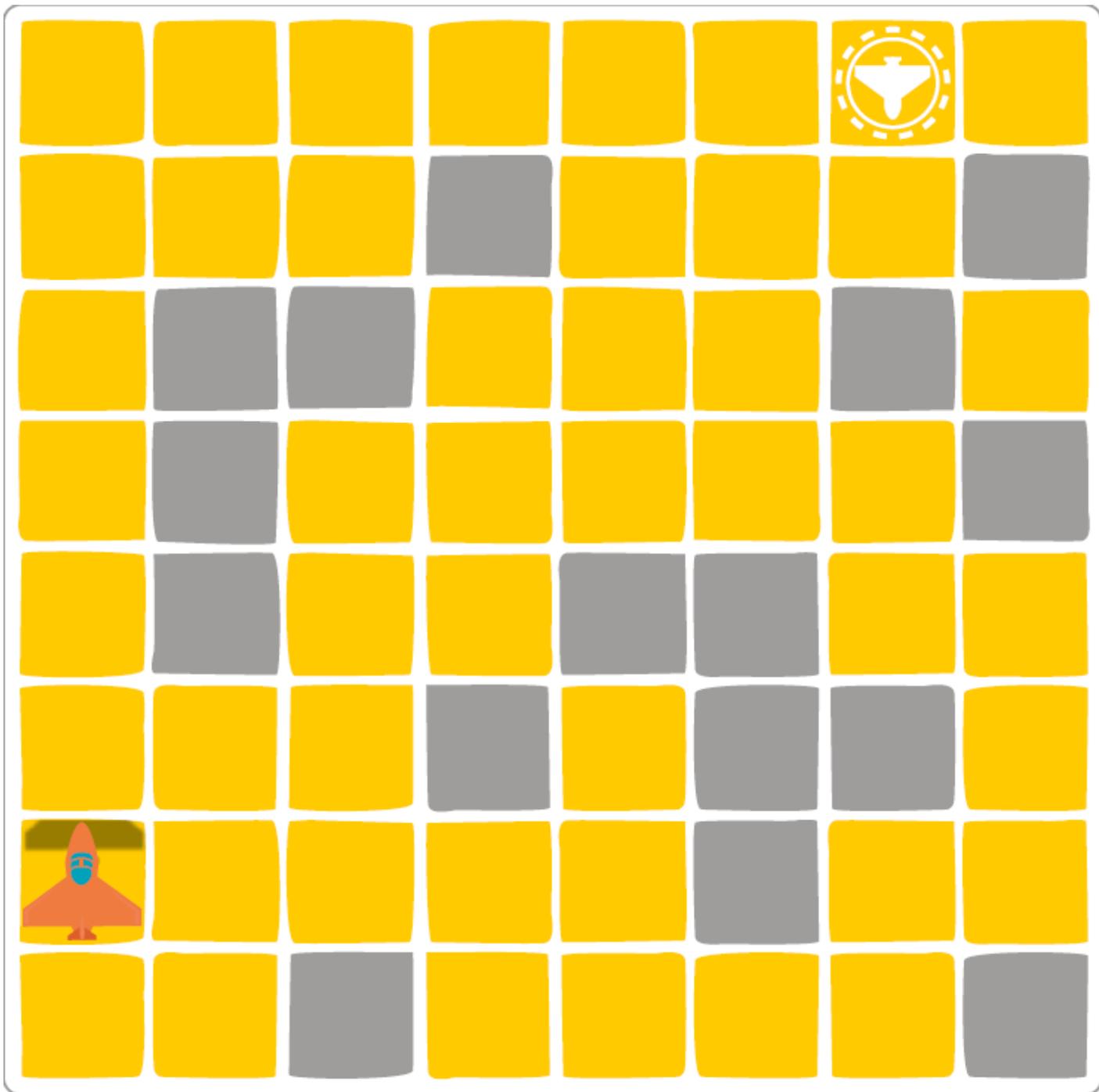


Arrivée :



Mission 2

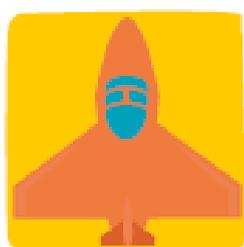
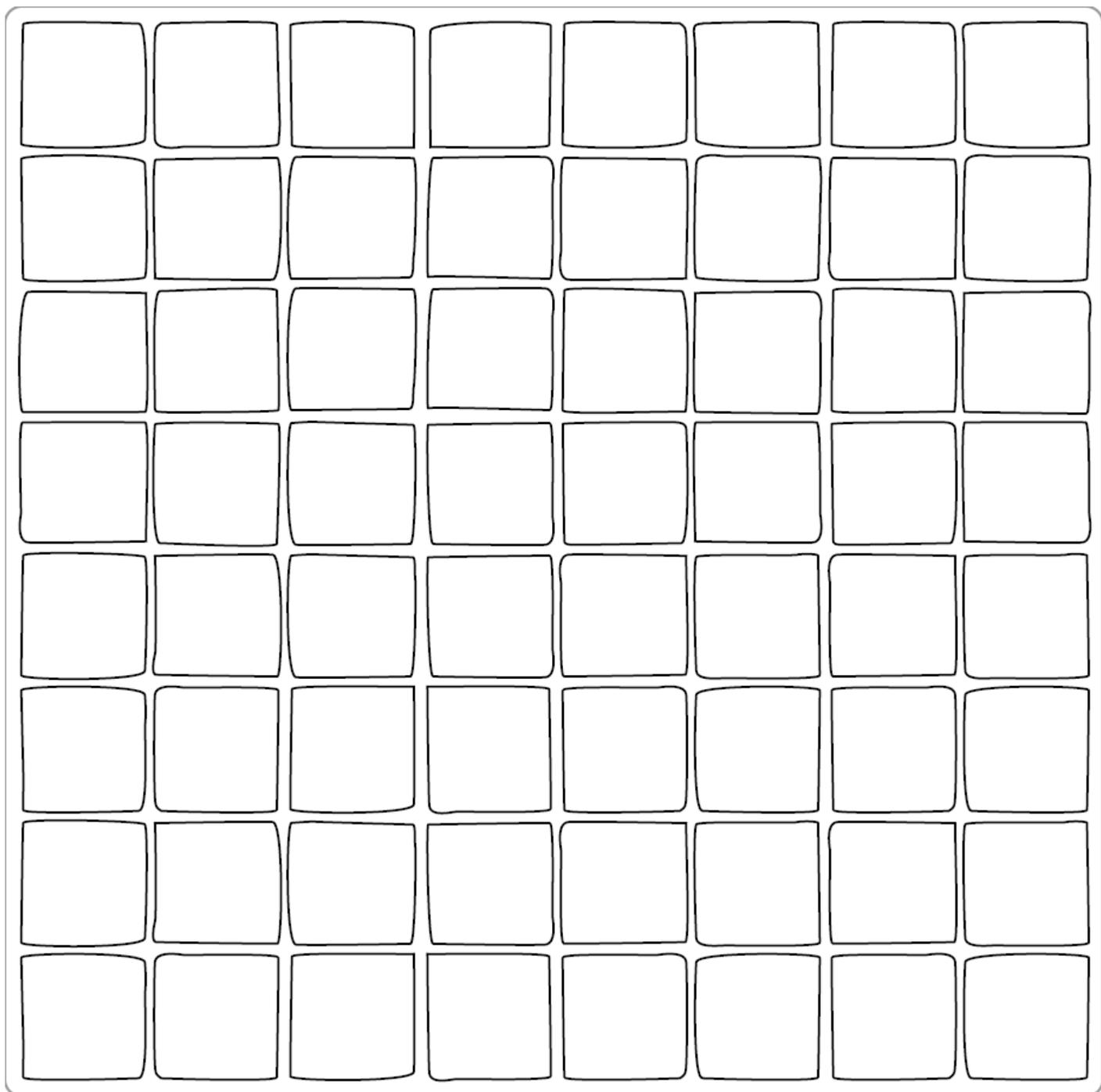
Attendre l'arrivée en utilisant le moins possible d'instructions et sans toucher les cases grises.



Mission 3

Effectuer une trajectoire carrée. Le carré doit avoir un côté de 3 unités (cases).

Le vaisseau et la grille.





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



AVANCER



AVANCER



AVANCER



AVANCER



PIVOTER À GAUCHE



PIVOTER À GAUCHE



PIVOTER À DROITE



PIVOTER À DROITE

