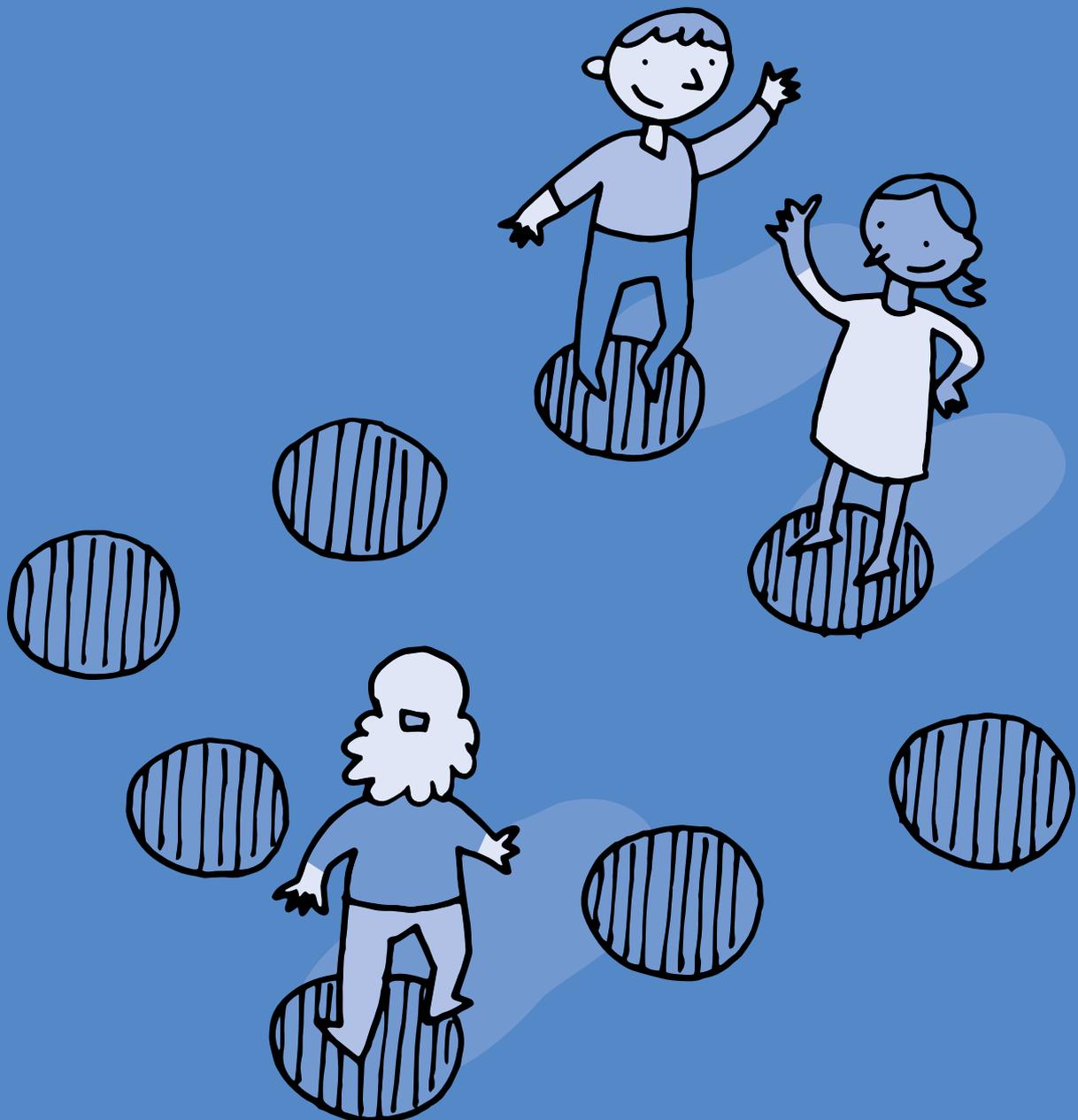


Scénario 1

Algorithmique débranchée



1^{re} • 2^e

Algorithmique débranchée

- 🎯 **Objectif du Plan d'études:** découvrir la science informatique en créant et exécutant des algorithmes et des programmes informatiques.
- 🎯 **Compétences opérationnelles, savoir-faire, savoirs:**
 - comprendre et utiliser les algorithmes¹
 - exécuter/créer un algorithme simple
 - savoir qu'un algorithme est une succession d'étapes permettant de résoudre un problème et d'effectuer une tâche
- 💡 **Intentions pédagogiques:** ce scénario contribue à enseigner les concepts des sciences du numérique (concept de machine, d'algorithme). Les élèves découvrent comment une machine peut être programmée sans jamais utiliser un robot et comprennent de quelle manière on peut les déplacer.
- ⚙️ **Description de l'activité de base:** les élèves s'initient à la découverte de la manière dont l'humain donne des ordres à une machine particulière: le robot. Ils·elles prolongent cette première expérience en programmant les déplacements d'un lutin sur un quadrillage et en enrichissant progressivement leurs programmes.

Séances	Résumé	Matériel
1: Le robot <i>idiot</i> Durée: 35 minutes	les élèves doivent programmer un robot qu'incarne l'enseignant·e (principe du jeu de rôle).	• pas de matériel particulier. Cette activité peut se faire en salle de classe, salle de gymnastique ou de rythmique
2: Programmer le déplacement d'un lutin Durée: 35 minutes	les élèves doivent trouver comment déplacer un lutin sur un quadrillage	• quadrillage au sol (peint dans une salle de l'école) ou sur une bâche • cartes-instructions à imprimer et à découper («Fiche 1»)
3: Programmer de manière séquentielle Durée: 40 minutes	les élèves doivent trouver comment déplacer un lutin sur un quadrillage (évaluation formative)	• quadrillage au sol (peint dans une salle de l'école) ou sur une bâche • feuilles de parcours à imprimer («Fiche 2» et «Fiche 3») • cartes-instructions à imprimer et à découper («Fiche 1»)

¹ Le mot *algorithme*, qui peut paraître complexe, est tout à fait utilisable avec les élèves, mais dans ce scénario le mot *programme*, également correct (puisqu'on programme un robot, puis un lutin) est le plus souvent utilisé.

Séance 1

Le robot *idiot*

 **Résumé:** les élèves doivent programmer un robot qu’incarne l’enseignant·e (principe d’un jeu de rôle)

 **Matériel:** pas de matériel particulier, cette activité peut se faire en salle de classe, salle de gymnastique ou de rythmique.

Temps 1.1: Le jeu du robot sans quadrillage

Modalités de travail: en grand groupe

 **Durée:** 15 minutes

Pour commencer cette séquence débranchée, l’enseignant·e va demander de l’aide aux élèves pour le guider, car il·elle ne sait plus comment aller du tableau à son bureau (par exemple).

Un début de séance possible peut consister à dire aux élèves: *Je vais jouer le rôle d’un robot.*

*Attention je deviens un robot et je vais commencer en étant dans la position **éteint**.*

À ce moment précis, l’enseignant·e baisse la tête, ferme les yeux et reste les bras ballants. Le démarrage consiste alors pour les élèves à trouver un ordre (une instruction) qui permettra au robot de s’allumer avant de lui donner les premières instructions. Cela peut être un ordre oral: *Allume-toi*, un ordre par le toucher (le bras, la tête...).

On pourra proposer à un élève de jouer le rôle du robot par la suite en lui expliquant ce qu’il·elle va devoir faire. Cela permet à l’enseignant·e de garder la maîtrise du groupe.

Si cela est possible, un travail en demi-groupe facilitera les échanges langagiers entre l’enseignant·e et les élèves. Si rien ne se passe, par exemple parce que les élèves ne sont pas assez autonomes pour imaginer cet ordre initial, dire aux élèves *Je ne suis pas allumé* puis si rien ne vient: *Dites-moi de m’allumer.*

Les élèves vont proposer des ordres (*va tout droit, va vers la porte*, etc) pour guider l’enseignant·e, qui s’appliquera à obéir scrupuleusement et au pied de la lettre à toutes les indications qu’on lui fournit.

Cette activité plaît beaucoup aux élèves quand l’enseignant·e fait semblant de se cogner contre le mur lorsqu’un élève donne comme instruction d’avancer en oubliant de préciser de combien de pas par exemple.

Temps 1.2: Mise en commun à l'oral

Modalités de travail en grand groupe ou en demi-groupe

 **Durée:** 10 minutes

Au bout d'une dizaine de minutes, l'enseignant·e prend en charge la synthèse à l'oral de ce que les élèves ont vécu.

L'enseignant·e fait la synthèse des différents ordres donnés par les élèves

- les ordres liés au déplacement (comme *avance*; *tourne*; *recule*...)
- les ordres liés à la précision du déplacement (*avance de deux cases*; *tourne à gauche*; *tourne à droite*...)

En particulier, en complément des points déjà repérés, l'enseignant·e fait le point sur les instructions de déplacement utilisées, leur clarté qui permet au robot d'agir de telle ou telle manière. Les élèves vont reprendre certaines instructions qui ont démontré leur efficacité pendant les exercices précédents. Ils·elles en inventent peu de nouvelles. Ces déplacements sont souvent *va vers la fenêtre*, *va vers la porte*, *va vers le tableau*, etc. Ces instructions font appel à des référentiels absolus, identifiables par tous. Amener les élèves à comprendre que :

- il ne faut pas parler tous en même temps pour que le *robot* ait le temps d'obéir aux indications dans l'ordre. Ce séquençement est la base de la programmation dite *séquentielle*.
- il faut découper l'action complexe en actions plus simples (le degré de détail exigé est laissé à l'initiative de l'enseignant·e).
- les indications doivent être explicites (libre à l'enseignant·e de jauger le degré d'implicite tolérable pour l'exercice). L'enseignant·e introduit ici le terme *instruction*.

Temps 1.3: Les élèves jouent le rôle du robot

Modalités de travail en grand groupe ou en demi-groupe

 **Durée:** 10 minutes

Pour renforcer ces notions, l'enseignant·e propose aux élèves de jouer, à leur tour, au *jeu du robot*. Selon les conditions, il est possible d'utiliser la salle de rythmique ou la salle de classe en petits groupes. L'enseignant·e précise le rôle de chacun·e: l'élève qui joue le robot et le programmeur qui donne les ordres.

À tour de rôle, les élèves se relaient pour jouer le robot, et leurs camarades définissent une mission: par exemple partir de la porte d'entrée pour aller chercher un livre sur l'étagère au fond de la classe. Les élèves *programmeurs* vont donner, une par une, les instructions nécessaires au robot.

L'enseignant·e souligne ici qu'il faut un langage commun, que le robot et les programmeurs comprennent. C'est ce qu'on appelle un *langage de programmation*.

Séance 2

Programmer le déplacement d'un lutin

 **Résumé:** les élèves doivent trouver comment déplacer un lutin sur un quadrillage.



Matériel:

- quadrillage au sol (peint dans une salle de l'école) ou sur une bâche
- cartes-instructions à imprimer et à découper («Fiche 1», *Les cartes-instructions pour programmer le lutin*)

Temps 2.1: Déplacement d'un lutin

Modalités de travail en grand groupe ou en demi-groupe



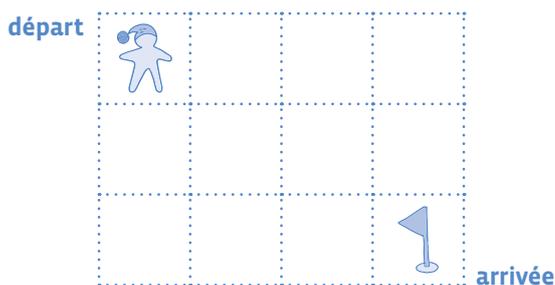
Durée: 15 minutes

Au tableau ou au sol, l'enseignant·e a préparé un quadrillage de 3 lignes et 4 colonnes. Dans une des cases, la silhouette ou la photo d'un lutin est punaisée (ou aimantée) et dans une autre case, une cible (par exemple, une maison ou une fleur).

Variantes:

La contextualisation de cet exercice, et de ceux des séances suivantes, variera évidemment d'une classe à l'autre. Le *lutin* peut être la peluche fétiche de la classe, la mascotte de l'école, le héros d'un livre ou d'une comptine.

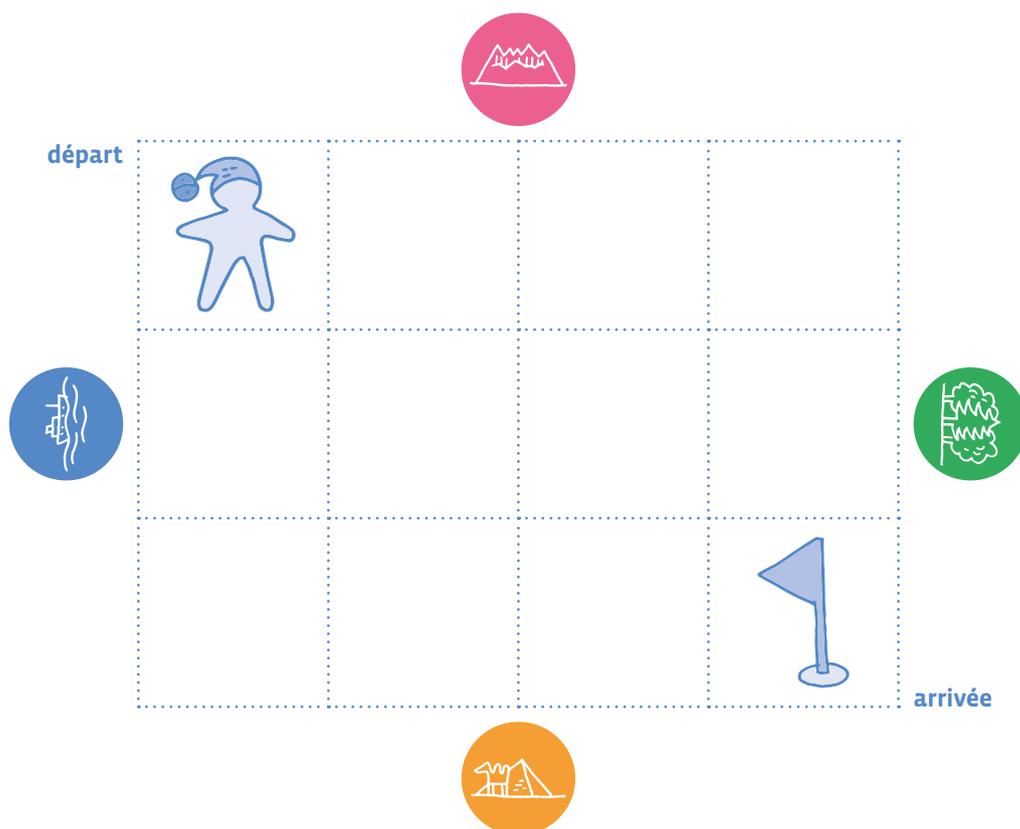
Nous utilisons ici le vocable *lutin* par simple souci de cohérence avec les logiciels de programmation visuelle Scratch et Scratch Jr qui seront détaillés dans des séquences ultérieures. Dans le même ordre d'idée, la *cible* doit ostensiblement attirer le *lutin*: un pirate convoitera des pièces d'or, un enfant voudra une sucette, etc.



L'enseignant·e demande aux élèves de guider, case par case, le lutin vers sa cible. Celui-ci ne peut pas se déplacer en diagonale. Les élèves doivent inventer le langage. Ils·elles se heurtent à des difficultés de latéralisation, à des difficultés de représentation spatiale. Que signifie *haut* ou *bas* pour le lutin, quand il est posé sur une table ou au tableau?

Avant que les élèves ne se découragent de cette difficulté, l'enseignant·e ajoute *au lointain*, le *lutin repère des paysages colorés*. Il·elle dessine alors quatre symboles colorés bien distincts pour concrétiser les quatre directions absolues: un désert jaune (qu'ici nous placerons par exemple au sud), une forêt verte (à l'est), une chaîne de montagnes roses (au nord), et une mer bleue (à l'ouest).

Les motifs choisis pour les directions absolues (océan, montagne, forêt et désert) doivent être illustratifs et permanents. Quel que soit le mouvement du lutin, il faut à tout prix que ces directions soient lointaines (en dehors de la carte comme des points inaccessibles) et qu'elles ne changent jamais (si nous avons choisi *la Gare* comme direction, l'instruction changerait de signification et donc d'absolu dès que la gare serait atteinte, ce qui est réhibitoire dans un langage de programmation).



Grâce à ces indicateurs, les élèves peuvent désormais donner des instructions explicites et comprises par tous. *Avance d'une case vers la forêt verte, Avance d'une case vers la montagne rose, etc.* Nous nous trouvons ici dans une logique de déplacement allocentrée.

Note pédagogique :

Langage allocentré : directions absolues
(va vers la fenêtre, la porte...)

Langage autocentré : directions relatives
(tourne vers la droite, avance, tourne vers la gauche, recule...).

Les élèves n'ont pas besoin de connaître ces termes.

Temps 2.2: Déplacement d'un lutin avec les cartes instructions

Modalités de travail en grand groupe puis en individuel

 **Durée:** 10 minutes

L'enseignant·e utilise alors les cartes-instructions de la «Fiche 1» pour écrire le programme au tableau. Il y a une infinité de programmes possibles pour résoudre ce problème, mais parmi les plus simples, les élèves proposeront certainement ceux-ci :

1:



2:



3:



La consigne de lecture de ces programmes est simple: on les lit de gauche à droite, et on applique les instructions l'une après l'autre.



L'enseignant·e souligne qu'il y a plusieurs programmes possibles, et non une solution unique chaque élève peut avoir sa propre solution!

Complément:

Ces instructions et ces indicateurs directionnels pourront être imprimés et coloriés sur des supports plus grands pour que les élèves puissent utiliser ce langage de programmation pour revivre le jeu du robot dans la cour, sous le préau, en salle de gymnastique ou de rythmique, et s'appropriier progressivement les instructions et la programmation des déplacements.

Temps 2.3: La synthèse avec l'enseignant·e

Modalités de travail en grand groupe ou en demi-groupe

 **Durée:** 10 minutes

Conclusion

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

- en combinant (suivant) des tâches simples, on peut réaliser une tâche complexe.
- un programme est écrit dans un langage que le lutin et l'élève peuvent comprendre.

L'enseignant·e pourra réaliser une affiche en reprenant les points de la synthèse et en insérant un exemple de programme et de déplacement du lutin.

Séance 3

Programmer en séquentiel

 **Résumé:** les élèves doivent trouver comment déplacer un lutin sur un quadrillage.



Matériel:

- quadrillage au sol (peint dans une salle de l'école) ou sur une bâche
- feuilles de parcours à imprimer («Fiche 2», «Fiche 3» et «Fiche 4»)
- cartes-instructions à imprimer et à découper («Fiche 1», *Les cartes-instructions pour programmer le lutin*)

Cette séance d'évaluation formative permet de vérifier la bonne compréhension des notions vues précédemment. Elle peut se faire en classe entière à l'oral, ou individuellement.

Il est souhaitable de réaliser cette évaluation sur un quadrillage au sol plutôt que sur une «Fiche 4».

Dans ce dernier cas, prévoir autant d'exemplaires de la Fiches 2 ou 3 que d'élèves, ainsi qu'une «Fiche 4» par binôme.

En 1^{re} – 2^e, on peut prévoir des jetons pour matérialiser le lutin sur la grille et un jeton de lecture pour indiquer l'instruction en cours d'application. À cet âge, les élèves risquent en effet de rencontrer des difficultés pour suivre avec leurs doigts à la fois le déplacement du lutin et la lecture des instructions du programme.

Temps 3.1: Exécuter un programme

Modalités de travail en classe entière, en binôme ou en individuel



Durée: 20 minutes

L'enseignant·e fait ré-émerger les notions vues précédemment: **un programme est une succession d'instructions données au lutin dans un langage** précis. Il·elle propose de s'entraîner sur de nouveaux programmes et de nouveaux parcours.

Il faut absolument privilégier l'activité de manipulation concrète du lutin par les élèves sur un quadrillage au sol. Cela permet d'éviter de prime abord l'utilisation de jeton pour matérialiser le lutin. Le jeton ne sera alors utilisé que pour suivre le programme, ce qui évite la confusion d'utiliser un même objet pour deux tâches différentes.

Dans un deuxième temps, on peut se servir d'une fiche contenant un parcours quadrillé et trois programmes, il s'agit d'exécuter pas à pas chacun des programmes proposés pour trouver sur quelle case se retrouve, finalement, le lutin (peut être représenté par une carte-image, une mascotte ou un élève qui joue ce rôle...).

On place le lutin sur le coin du quadrillage, puisqu'il redémarre de la case où le précédent programme l'emmène. Pour scénariser l'exercice, l'enseignant·e peut dire que le lutin va d'abord cueillir un champignon (parcours vert), puis puiser de l'eau (parcours bleu) avant d'aller à sa maison (parcours rose).

Pour s'assurer que tous les élèves repartent bien du même point à chaque étape, on organisera une activité sur un grand quadrillage au sol plastifié en utilisant des programmes simples. Les élèves peuvent dessiner les ronds de couleur à la fin de chaque parcours (de chaque programme).

Il est intéressant de questionner les élèves sur la pluralité des chemins pour atteindre une case donnée, afin qu'ils·elles se rendent compte qu'il y a de nombreuses possibilités, mais qu'elles ne sont pas toutes égales en terme de complexité, de distances parcourues et de temps passé.

On peut poser la question suivante: Existe-t-il un autre chemin qui permette au lutin de se rendre au champignon, à la fontaine et à la maison? Cela peut donner de nouveaux programmes possibles (qui seront peut être plus longs mais qui seront corrects):

Un autre programme pour se rendre au champignon peut être le suivant:



Un autre programme pour se rendre à la fontaine:



Un autre programme pour se rendre à la fontaine:



En 2^e, on exploite des parcours plus longs, plus complexes («Fiche 2»). Le parcours rose est difficile (faire attention au fait que les élèves ne veulent pas revenir en arrière et prennent une diagonale ou sautent la case). Le parcours impose de revenir sur ses pas, ce qui n'est pas simple pour certains élèves (pourquoi revenir en arrière?).

Si besoin, le proposer comme exercice facultatif.

parcours vert (depuis le coin du départ)

parcours bleu (depuis le point vert)

parcours rose (depuis le point bleu)

Solution:

parcours vert (depuis le coin du départ)

parcours bleu (depuis le point vert)

parcours rose (depuis le point bleu)

Temps 3.2: Exécuter un programme

Modalités de travail en classe entière, en groupes ou en individuel selon le niveau

 **Durée:** 20 minutes

En 1^{re}, il est conseillé de le faire en classe entière avec l'enseignant·e.

En 2^e, ça peut être soit une activité de groupe soit une activité individuelle.



Si possible, l'enseignant·e commence par un travail avec un quadrillage au sol et des cartes instructions à manipuler.

Le passage par une activité où le corps de l'élève est mis en jeu permettra une plus grande appropriation des concepts plus abstraits qui sont ensuite travaillés sur une fiche.

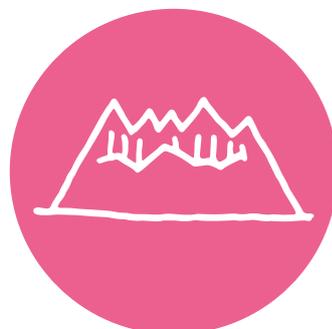
L'enseignant·e distribue la «Fiche 4», qui propose de programmer le lutin pour l'amener à destination en évitant un obstacle (l'étang). Tout comme à la séance précédente, il y a plusieurs programmes possibles.

En cas de difficulté, l'enseignant·e propose de nouveaux exercices semblables, avant de poursuivre.

Pour écrire un programme, l'enseignant·e peut proposer aux élèves un modèle avec un carré comme point de départ et une flèche pour indiquer dans quel sens les élèves doivent écrire les instructions.



Les cartes-instructions pour programmer le lutin



Un parcours découpé en plusieurs chemins (1^{re})

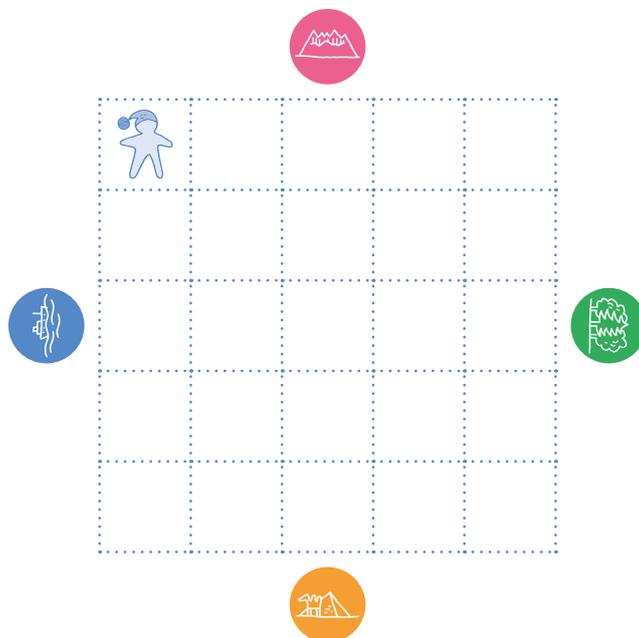
parcours vert (depuis le coin du départ)



parcours bleu (depuis le champignon)



parcours rose (depuis la fontaine)



Consigne

L'enseignant·e donne la consigne au fur et à mesure afin de ne pas surcharger cognitivement l'élève.

1. Le lutin suit le parcours pour trouver un champignon: place un champignon vert dans la case où il s'arrête.
2. Le lutin part du champignon vers une fontaine avec de l'eau: place une fontaine bleue dans la case où il s'arrête.
3. Le lutin part de la fontaine bleue vers la maison rose: place une maison rose dans la case où il s'arrête.

Un programme découpé en sous-programmes (2^e)

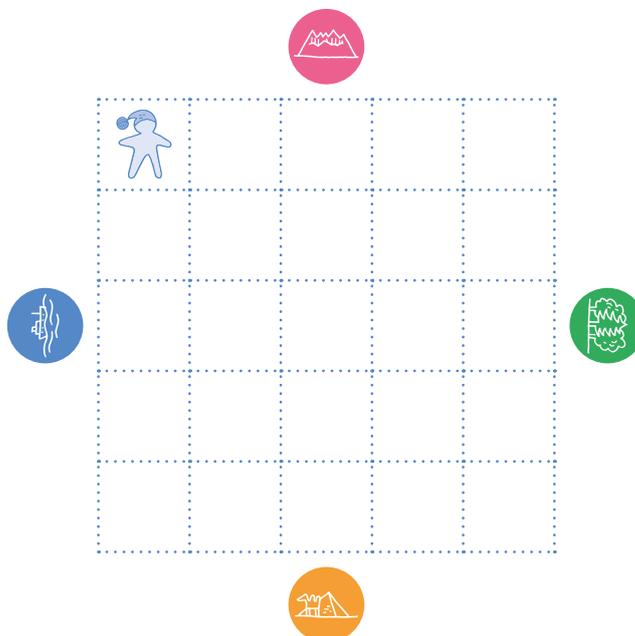
parcours vert (depuis le coin du départ)



parcours bleu (depuis le point vert)



parcours rose (depuis le point bleu)



Consigne

L'enseignant·e donne la consigne au fur et à mesure afin de ne pas surcharger cognitivement l'élève.

1. Le lutin suit le parcours pour trouver un champignon: place un champignon vert dans la case où il s'arrête.
2. Le lutin part du champignon vers une fontaine avec de l'eau: place une fontaine bleue dans la case où il s'arrête.
3. Le lutin part de la fontaine bleue vers la maison rose: place une maison rose dans la case où il s'arrête.

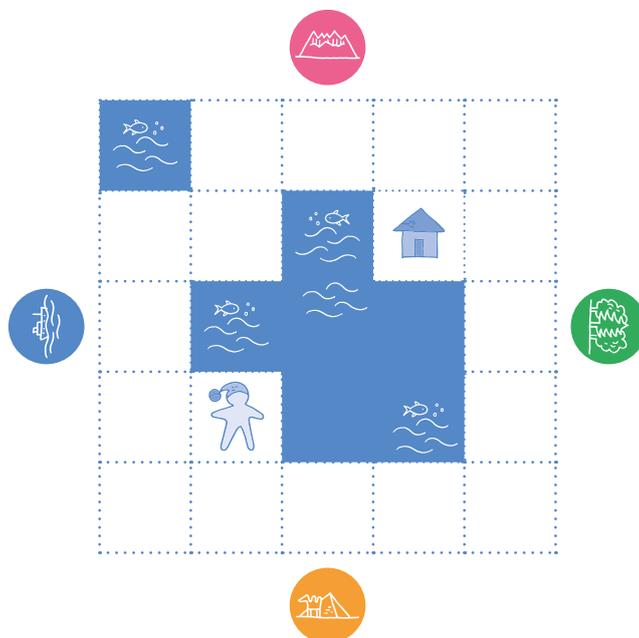
Un parcours plus complexe

Consigne

Construis un programme permettant au lutin de rejoindre sa maison.

Attention, il ne peut pas aller dans l'eau!

Programme:

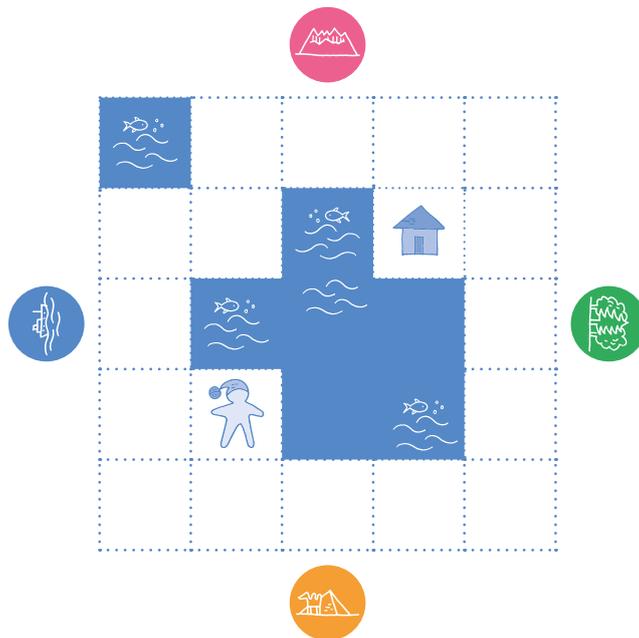


Consigne

Construis un programme permettant au lutin de rejoindre sa maison.

Attention, il ne peut pas aller dans l'eau!

Programme:



3^e • 4^e Algorithmique débranchée

🎯 **Objectif du Plan d'études:** découvrir la science informatique en créant et exécutant des algorithmes¹ et des programmes informatiques.

🎯 **Compétences opérationnelles, savoir-faire, savoirs:**

- comprendre et utiliser les algorithmes
- exécuter/créer un algorithme simple
- exécuter/créer un algorithme avec une boucle à une instruction
- exécuter/créer un algorithme avec une condition
- **savoir qu'un algorithme:**
 - est une succession d'étapes permettant de résoudre un problème, d'effectuer une tâche
 - peut contenir des instructions, des tests, des boucles
 - peut ne pas donner une solution parfaite
- **savoir que:**
 - une condition est une expression qui est soit vraie, soit fausse
 - une boucle permet de répéter plusieurs fois les mêmes instructions
 - certaines boucles, dites *infinies*, ne s'arrêtent jamais

💡 **Intentions pédagogiques:** Ce scénario contribue à enseigner les concepts des sciences du numérique (concept de machine, d'algorithme). Les élèves découvrent comment une machine peut être programmée sans jamais utiliser un robot et comprennent de quelle manière on peut les déplacer.

⚙️ **Description de l'activité de base:** Les élèves programment des déplacements en enrichissant progressivement leur langage de programmation avec des instructions conditionnelles et des boucles.

Dans le PER (Plan d'études romand), en mathématiques, on trouve dans Explorer l'espace (Repérage dans le plan et dans l'espace) l'attente fondamentale suivante: l'élève situe des objets par rapport à lui et par rapport à d'autres objets (devant, derrière, sur, sous, à côté de, entre, à l'intérieur de, à l'extérieur de) et l'élève détermine sa position ou de celle d'un objet (devant, derrière, à côté, sur, sous, entre, à l'intérieur, à l'extérieur, à gauche, à droite,...) selon différents points de repères en 3^e et 4^e.

L'enseignant·e souligne ici qu'il faut un langage commun, que le robot et les programmeurs comprennent. C'est ce qu'on appelle un *langage de programmation*.

Séances	Résumé	Matériel
1: Se déplacer avec quadrillage Durée: 40 minutes	Les élèves jouent au jeu du robot sur un quadrillage (le robot est incarné tout à tour par l'enseignant·e et les élèves).	• 1 quadrillage (bâche quadrillée avec scotch ou peinture, sol quadrillé avec de la craie, cerceaux)
2: Apprendre à programmer avec des tests conditionnels Durée: 45 minutes	Les élèves programment le déplacement d'un lutin sur un quadrillage en ajoutant des instructions conditionnelles.	• Fiche 1, <i>Des cartes des chalets</i> • Fiche 2, <i>Des instructions pour entrer</i> • Fiche 3, <i>Du lutin des cartes conditionnelles</i> • Fiche 4, <i>Les cartes-instructions pour programmer le lutin</i> • «Fiche 5», <i>Parcours à imprimer pour la séance</i>
3: Comprendre ce qu'est une boucle Durée: 45 minutes	Les élèves programment le déplacement du lutin en utilisant moins d'instructions (notion de boucle).	• Fiche 1, <i>Des cartes des chalets</i> • Fiche 2, <i>Des instructions pour entrer</i> • Fiche 3, <i>Du lutin des cartes conditionnelles</i> • Fiche 4, <i>Les cartes-instructions pour programmer le lutin</i> • «Fiche 6», <i>Parcours à imprimer pour la séance</i>

¹ Le mot *algorithme*, qui peut paraître complexe, est tout à fait utilisable avec les élèves, mais dans ce scénario le mot *programme*, également correct (puisqu'on programme un robot, puis un lutin) est le plus souvent utilisé.

Séance 1

Se déplacer avec quadrillage

 **Résumé:** les élèves jouent au jeu du robot sur un quadrillage (le robot est incarné tout à tour par l'enseignant·e et les élèves).

 **Matériel:** 1 quadrillage (bâche quadrillée avec scotch ou peinture, sol quadrillé avec de la craie, cerceaux).

Le jeu peut être mené dans la classe, dans la cour de récréation ou dans un espace de l'école avec un quadrillage au sol. S'il n'y a aucune possibilité de disposer un quadrillage, il est quand même possible de mener la séance, en parlant de *pas* et non de *cases*.

Comment faire pour qu'un robot (sur le quadrillage) puisse se rendre sur une case donnée et prendre le cube placé là ? L'enseignant·e joue le rôle du robot. Les instructions orales ne fonctionnent pas. L'enseignant·e reste statique et attend pour réagir que les élèves lui donnent une instruction visuelle, tactile ou sonore.

Faire jouer le rôle du robot à plusieurs élèves en variant à chaque fois les capacités du robot:

- Il est aveugle ou non voyant. Le robot peut en revanche suivre des instructions orales ou par le toucher.
- Il ne parle pas notre langue. Le robot peut suivre des instructions sous la forme de gestes.
- Il est sourd ou malentendant. Le robot peut suivre des instructions par le visuel ou le toucher.
- Il est aveugle (non voyant) et sourd (ou malentendant). Le robot peut suivre une instruction uniquement par le toucher (toucher léger, ne doit pas le faire bouger).

Les élèves adaptent les réponses possibles à chaque modalité. On pourra alors conclure qu'un robot ne suit les instructions que s'il comprend ce qu'elles veulent dire.

Pour permettre aux élèves d'utiliser différents langages, l'enseignant·e peut proposer progressivement différents éléments:

- couleurs ou éléments du mobilier (je vais vers la fenêtre, je vais vers le bleu)
- points cardinaux
- se déplacer comme le Blue-Bot

Note pédagogique:

- **langage allocentré:** directions absolues (va vers la fenêtre, la porte...).
- **langage autocentré:** directions relatives (tourne vers la droite, avance, tourne vers la gauche, recule...).

Même si les élèves n'ont pas besoin de connaître ces termes, en 3^e et 4^e, il est intéressant de voir la logique allocentrée (ou absolue) puis la logique autocentrée (ou relative). Cette dernière 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.

Temps 1.2: Avec des repères de couleur

Modalités de travail: en grand groupe

 **Durée:** 20 minutes

Dans un deuxième temps, l'enseignant·e va mettre des bracelets de couleur aux poignets de l'élève. Par exemple, on choisira la couleur verte pour le poignet droit et la couleur rouge pour le poignet gauche. Cela permettra aux élèves de se repérer en fonction des couleurs et de se diriger en fonction d'indication de couleur et non de droite ou de gauche pour les élèves encore peu ou pas latéralisés.

On peut également donner des flèches que l'on place au sol devant l'élève. Il·elle se déplace ainsi en fonction de sa position, sans se référer à des repères extérieurs.

Collectivement, l'enseignant·e débat du vocabulaire avec la classe et formalise les choix effectués.

Allocentré:

- vers la porte = avance d'une case vers la porte
- vers la fenêtre = recule d'une case vers la fenêtre
- vers le tableau = va à droite d'une case vers le tableau
- vers l'étagère = va à gauche d'une case vers l'étagère

ou

- haut = avance d'une case vers le haut
- bas = recule d'une case vers le bas
- droite = va à droite d'une case vers la droite
- gauche = va à gauche d'une case vers la gauche

Autocentré en 3^e – 4^e:

- avancer = avance d'une case devant soi
- reculer = recule d'une case derrière soi
- droite = glisse¹ à droite d'une case
- gauche = glisse¹ à gauche d'une case
- pivote droite = pivote sur place d'un quart de tour vers la droite
- pivote gauche = pivote sur place d'un quart de tour vers la gauche

¹ glisse: il s'agit d'une simple translation, d'un pas de côté, sans pivotement

Séance 2

Apprendre à programmer avec des tests conditionnels

 **Résumé:** les élèves programment le déplacement d'un lutin sur un quadrillage en ajoutant des instructions conditionnelles.

 **Matériel:**

- Fiche 1, *Des cartes des chalets*
- Fiche 2, *Des instructions pour entrer*
- Fiche 3, *Du lutin des cartes conditionnelles*
- Fiche 4, *Les cartes-instructions pour programmer le lutin*
- «Fiche 5», *Parcours à imprimer pour la séance*

Préparation

L'enseignant·e place des *chalets* ouverts (porte verte ouverte), fermés (porte rouge fermée) ou indéterminés (noir et blanc, on ne sait pas s'ils sont ouverts ou fermés). Pour cela, les cartes chalets possèdent au recto un chalet noir et blanc et au verso soit un chalet ouvert soit un chalet fermé.



Temps 2.1: Situation déclenchante

Modalités de travail: en grand groupe

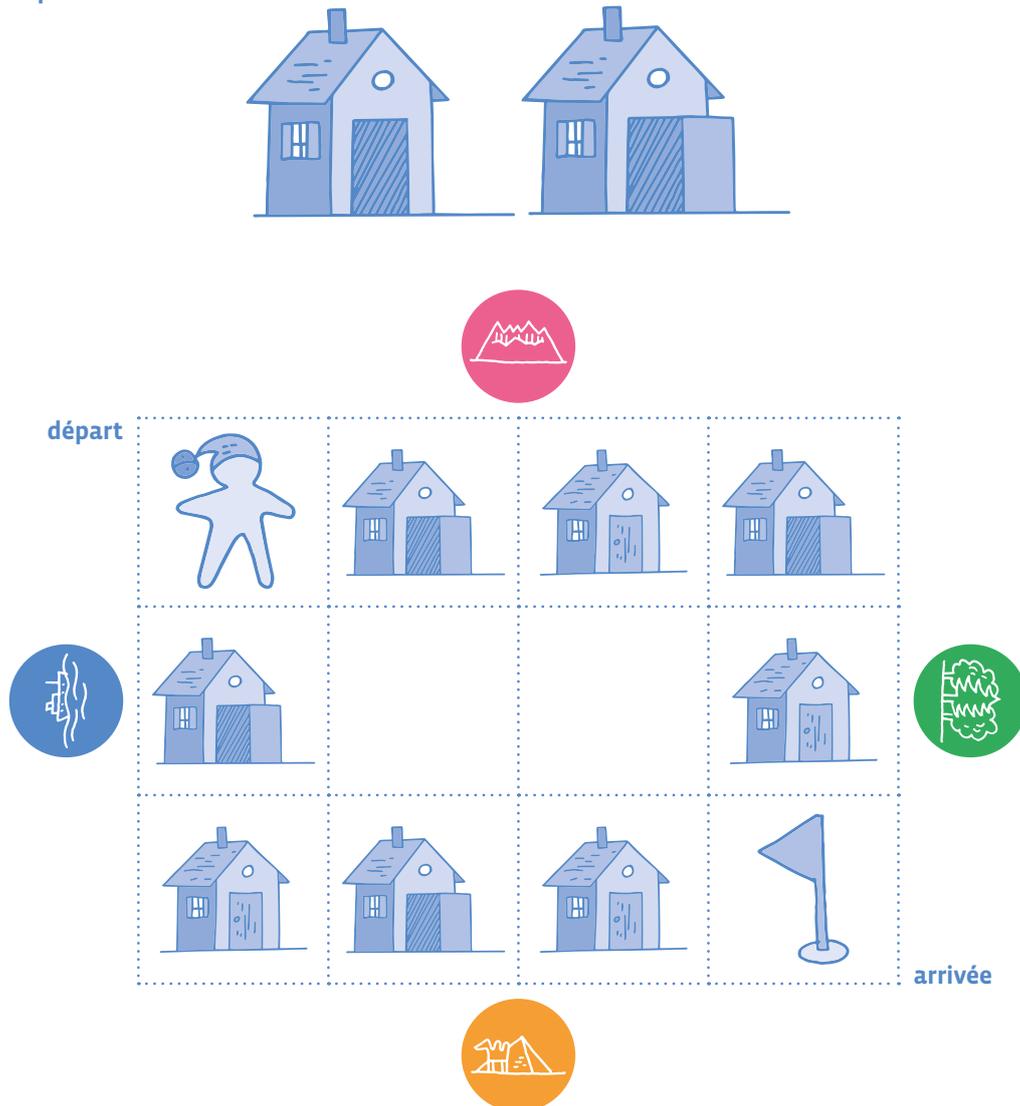
 **Durée:** 15 minutes

Autour d'un quadrillage, nous disposons de quatre symboles colorés bien distincts pour concrétiser les quatre directions absolues: un désert orange (qu'ici nous placerons par exemple au sud), une forêt verte (à l'est), une chaîne de montagnes rose (au nord), et une mer bleue (à l'ouest).

Les motifs choisis pour les directions absolues (océan, montagne, forêt et désert) doivent être illustratifs et permanents. Quel que soit le mouvement du lutin, il faut à tout prix que ces directions soient lointaines (en dehors de la carte comme des points inaccessibles) et qu'elles ne changent jamais (si nous avons choisi *la Gare* comme direction, l'instruction changerait de signification et donc d'absolu dès que la gare serait atteinte, ce qui est réhébitorique dans un langage de programmation). En 5^e, on pourra introduire les points cardinaux, mais en attendant, il faut choisir des horizons *caricaturaux*, mais intuitifs.

Nous ajoutons sur le parcours des chalets ouverts ou fermés (Fiche 1).

Par exemple :

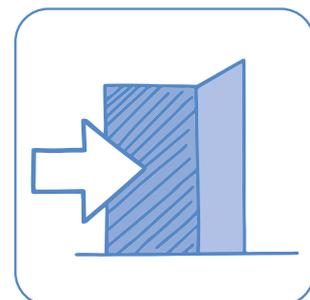


L'enseignant·e présente les cartes-chalets, avec la règle du jeu : si le lutin essaie d'entrer dans un chalet ouvert, il peut y trouver de quoi boire et manger et peut continuer le jeu, si le lutin essaie d'entrer dans un chalet fermé, ce qui est interdit, il doit retourner au début du parcours.

Il·elle pose ensuite une simple question : Avec le langage de programmation déjà utilisé auparavant, le lutin sait-il entrer dans un chalet? Non, il ne sait que se déplacer.

On peut demander aux élèves comment on pourrait faire pour programmer le lutin avec ce nouveau paramètre. En fonction des propositions des élèves, l'enseignant·e note les idées intéressantes au tableau, discute de la pertinence des différentes idées et introduit ensuite un nouveau mot de vocabulaire au langage de programmation : *entrer dans le chalet* (Fiche 2).

Afin de bien insister sur le fait que cette carte est indispensable pour entrer dans le chalet (si l'on n'a pas cette carte-instruction, alors on n'entre pas), l'enseignant·e propose de résoudre ce premier parcours collectivement, pour permettre au lutin d'arriver à la fin du parcours.



Cette carte signifie : *Entrer dans le chalet.*

L'enseignant·e fournit deux **programmes** différents qui contiennent volontairement des erreurs:

programme 1 (vers le bas):



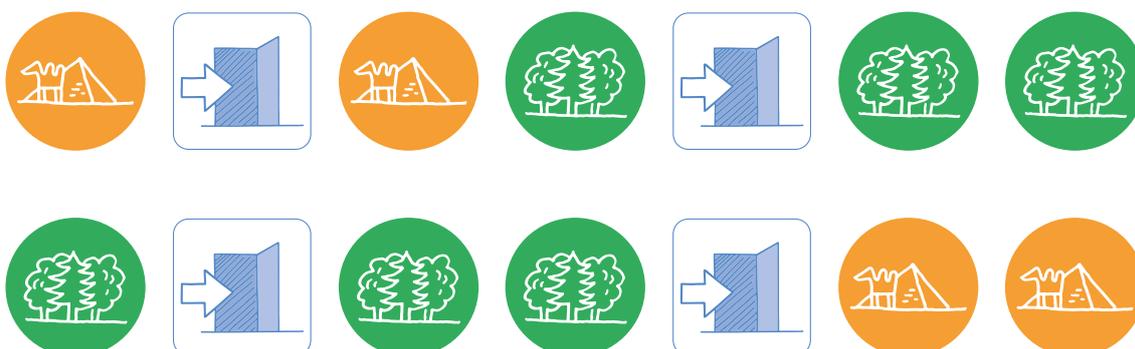
programme 2 (vers la droite):



- l'élève arrive au premier chalet ouvert ; il essaie d'y entrer.
- il arrive au premier chalet fermé ; sans essayer d'y entrer.
- il arrive au deuxième chalet ouvert ; il oublie d'y entrer.
- il arrive au deuxième chalet fermé ; il se trompe et essaie d'y entrer.

Avec cette démonstration, les élèves retiennent qu'être sur la même case qu'un chalet ne signifie aucunement y entrer.

En collectif, les élèves proposent une correction pour déboguer le problème en fonction des deux parcours possibles:

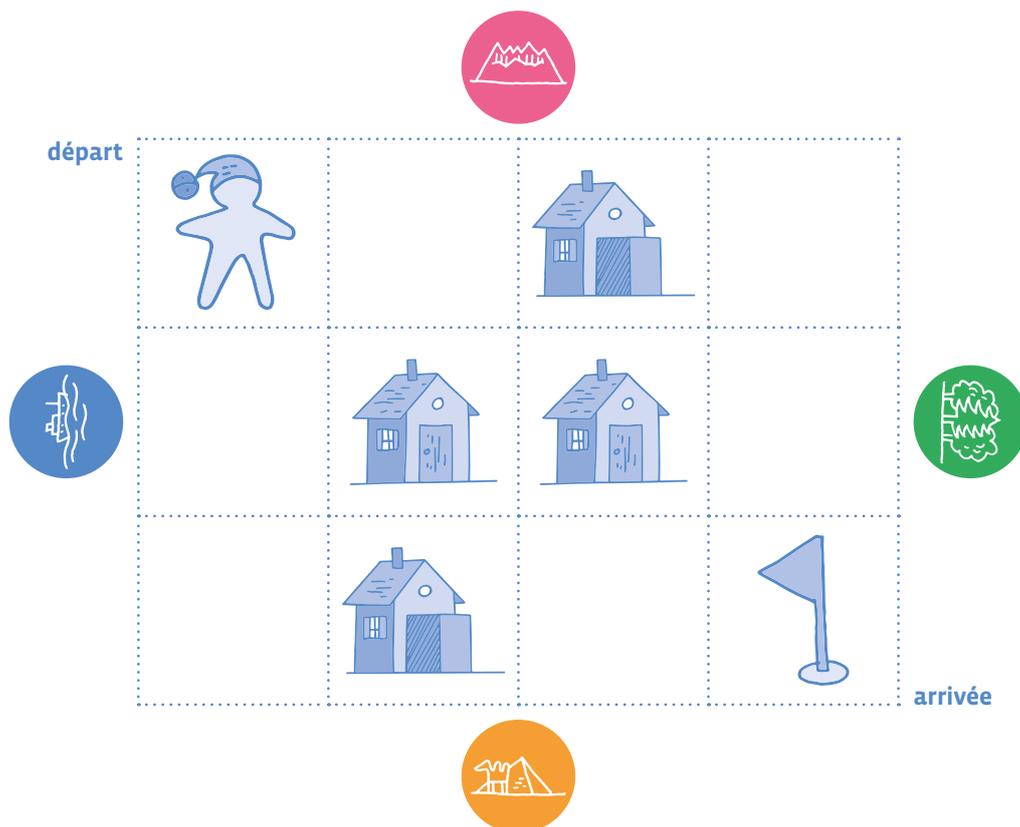


Temps 2.2: Expérimentation, passer dans tous les chalets ouverts en évitant les chalets fermés

Modalités de travail: collectif

 **Durée:** 10 minutes

À la suite du Temps 2.1, l'enseignant·e propose un nouveau parcours, avec pour consigne de passer obligatoirement par tous les chalets ouverts. Par exemple:



Avec la même consigne, la classe aboutit avec cet exemple à un programme comme celui-ci (il existe évidemment de nombreux programmes possibles):



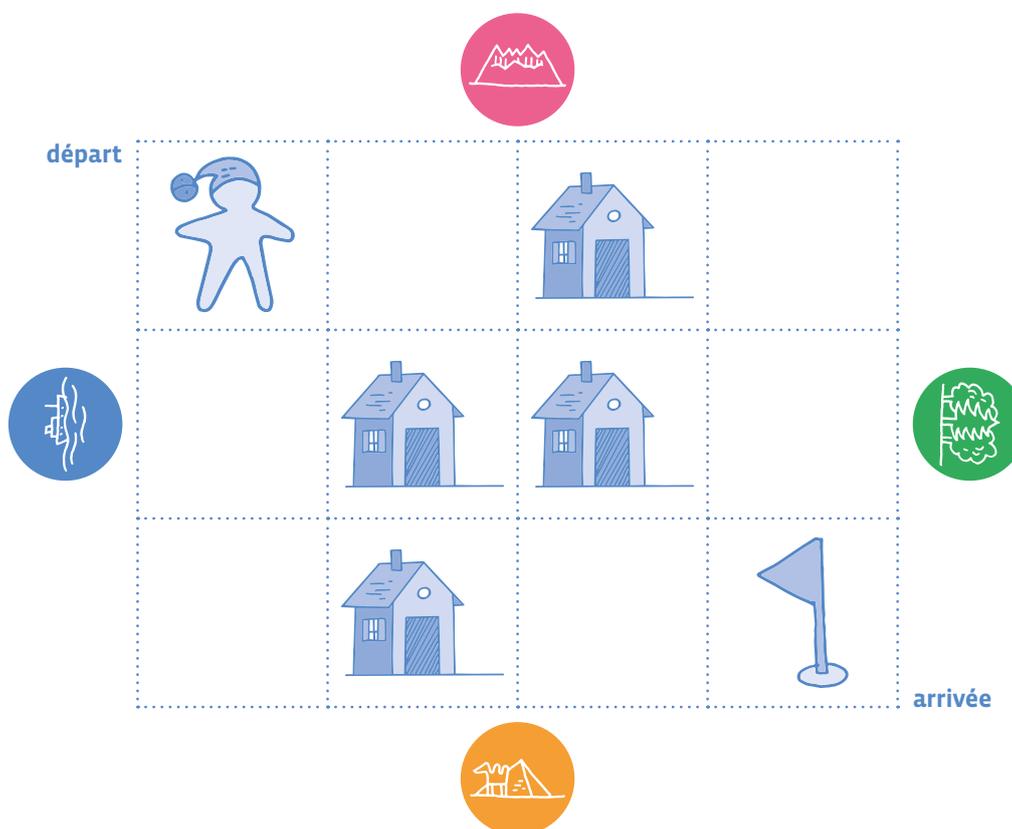
Temps 2.3: Expérimentation, passer dans tous les chalets ouverts dans un parcours anonyme (4^e)

Modalités de travail: collectif

 **Durée:** 10 minutes

Cette fois-ci, l'enseignant·e présente un parcours identique, à une nuance près: les chalets sont indéterminés, on ne sait pas s'ils sont ouverts ou fermés. Sous chaque carte-chalet indéterminée, il y a soit un chalet ouvert, soit un chalet fermé.

Le lutin sait où sont les chalets, mais il ne sait pas à l'avance s'ils sont ouverts ou fermés. Comment faire?



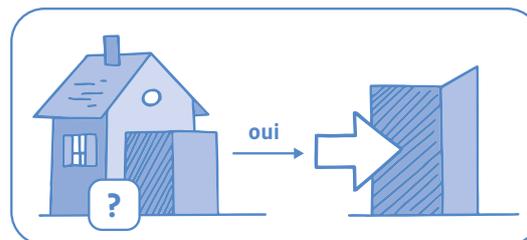
La discussion porte sur le fait que le lutin doit aller sur toutes les cases contenant un chalet dans le but de récolter le plus de provisions possibles, mais qu'il doit vérifier si le chalet est ouvert ou fermé avant de tenter d'y entrer. Dans un premier temps, la classe cherche à verbaliser l'instruction nécessaire.

Si le chalet est ouvert alors on entre.

Il faut choisir une façon de représenter le test et le branchement selon si la condition est remplie ou non.

L'enseignant·e propose alors une nouvelle carte-instruction (Fiche 3). Cette carte est un test; elle est composée d'une condition (ici *le chalet est-il ouvert?*) et d'une instruction (ici *entrer dans le chalet*) à effectuer uniquement si la condition est vérifiée.

En situation, lorsque le lutin pose la question, l'enseignant·e retourne la carte-chalet indéterminée et l'élève peut découvrir si le chalet est ouvert ou fermé.



Cette carte signifie: *Si le chalet est ouvert alors on entre, sinon on passe directement à l'instruction suivante.*

La classe doit donc améliorer le précédent programme avec cette nouvelle instruction, pour aider le lutin à visiter le maximum de chalets ouverts pour récupérer récolter le plus de provisions et arriver à la fin du parcours.

Par exemple:



On remarque que l'instruction apparaît bien 4 fois, pour chacun des chalets, car on ne sait pas à l'avance si les chalets sont ouverts ou fermés.

Conclusion

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

- Dans un programme, des tests disent quelle instruction effectuer quand une condition est vérifiée.
- Test conditionnel: *Si le chalet est ouvert alors on entre.*

Séance 3

Comprendre ce qu'est une boucle

 **Résumé:** les élèves programment le déplacement du lutin en utilisant moins d'instructions (notion de boucle).

-  **Matériel:**
- Fiche 1, *Des cartes des chalets*
 - Fiche 2, *Des instructions pour entrer*
 - Fiche 3, *Du lutin des cartes conditionnelles*
 - Fiche 4, *Les cartes-instructions pour programmer le lutin*
 - «Fiche 6», *Parcours à imprimer pour la séance*

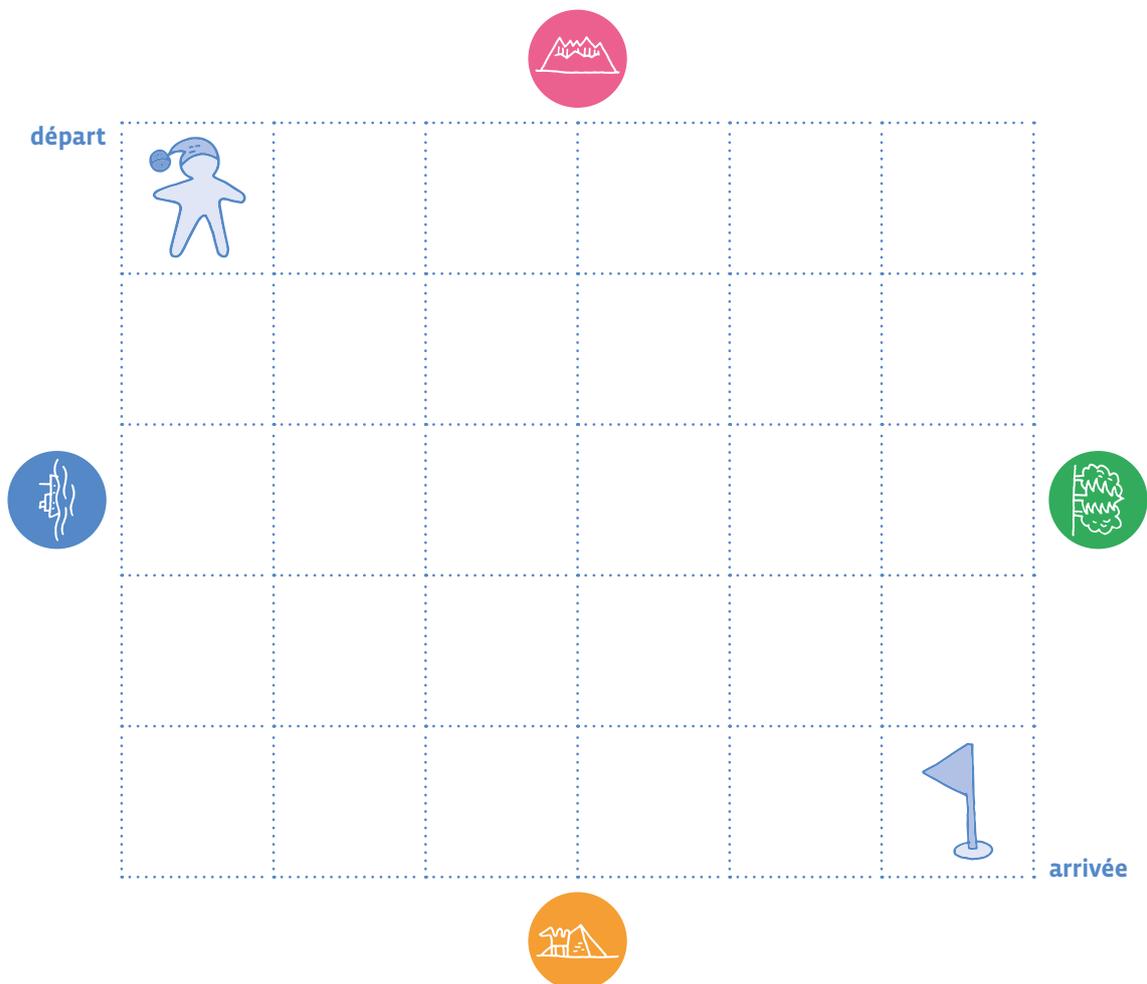
Lorsque les parcours deviennent longs ou complexes, les élèves prennent conscience de l'importance de simplifier l'écriture d'un programme ils-elles découvrent les boucles qui permettent d'éviter les répétitions.

Temps 3.1: Situation déclenchante

Modalités de travail: en grand groupe

 **Durée:** 15 minutes

L'enseignant·e présente un nouveau parcours, encore plus grand que les précédents (5 lignes par 6 colonnes), sans chalets.



Il·elle demande aux élèves d'écrire un programme permettant au lutin d'aller jusqu'à la case **arrivée**.

Parmi les propositions les plus simples des élèves, on trouvera



Ou encore:



Il est intéressant de faire remarquer aux élèves qu'il y a de nombreux programmes qui fonctionnent et permettent au lutin de rejoindre la case arrivée.

De nombreuses cartes sont nécessaires pour ce programme, pourtant très simple:

L'enseignant·e demande aux élèves comment on pourrait le raccourcir. En cas de besoin, il·elle leur fait remarquer qu'il y a beaucoup de répétitions. Plutôt que d'utiliser la même carte plusieurs fois, n'est-il pas possible d'indiquer, sur la carte, qu'on va répéter l'instruction plusieurs fois ? La classe discute des diverses propositions qui sont faites.

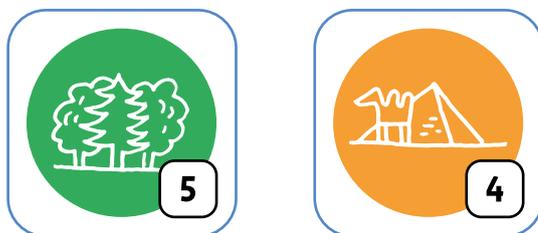
Il est conseillé de marquer, au crayon à papier sur la carte (ou au stylo effaçable si elle est plastifiée), le nombre de fois que celle-ci doit être appliquée. Cette notation a l'avantage d'être compatible avec le design des boucles dans le logiciel *Scratch Jr*.



Cette carte signifie: Avance de 5 cases vers la forêt verte.

L'enseignant·e explicite la notion de **boucle**, qui est la répétition d'une même instruction, ce qui permet de simplifier l'écriture (et la compréhension) d'un programme.

La classe utilise alors des boucles pour simplifier le programme précédent, qui devient:



A l'oral et/ou à l'écrit, l'élève va dire ou écrire:

Le lutin avance de 5 cases vers la forêt puis de 4 cases vers le désert.

La classe cherche ensuite, parmi les programmes proposés précédemment, lesquels peuvent être simplifiés en utilisant des boucles.

Une séance peut consister à passer des repères allocentrés à des repères sous la forme de flèches.
Demander la réécriture de ce programme en utilisant des flèches (→ ↓).

Solutions:

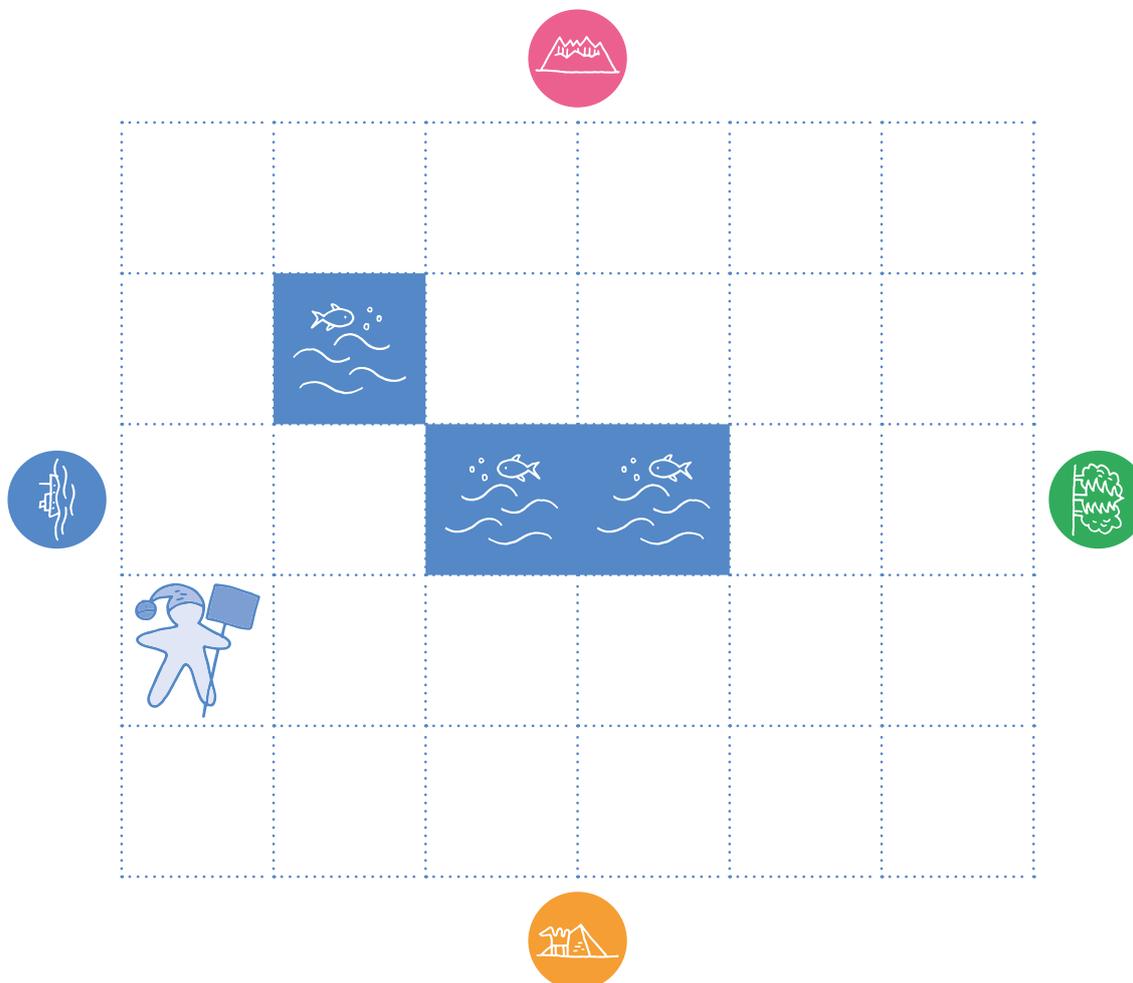
- → → → → → ↓ ↓ ↓ ↓ ↓
- 5 → 4 ↓ (en utilisant des boucles)

Temps 3.2: Exercice complémentaire

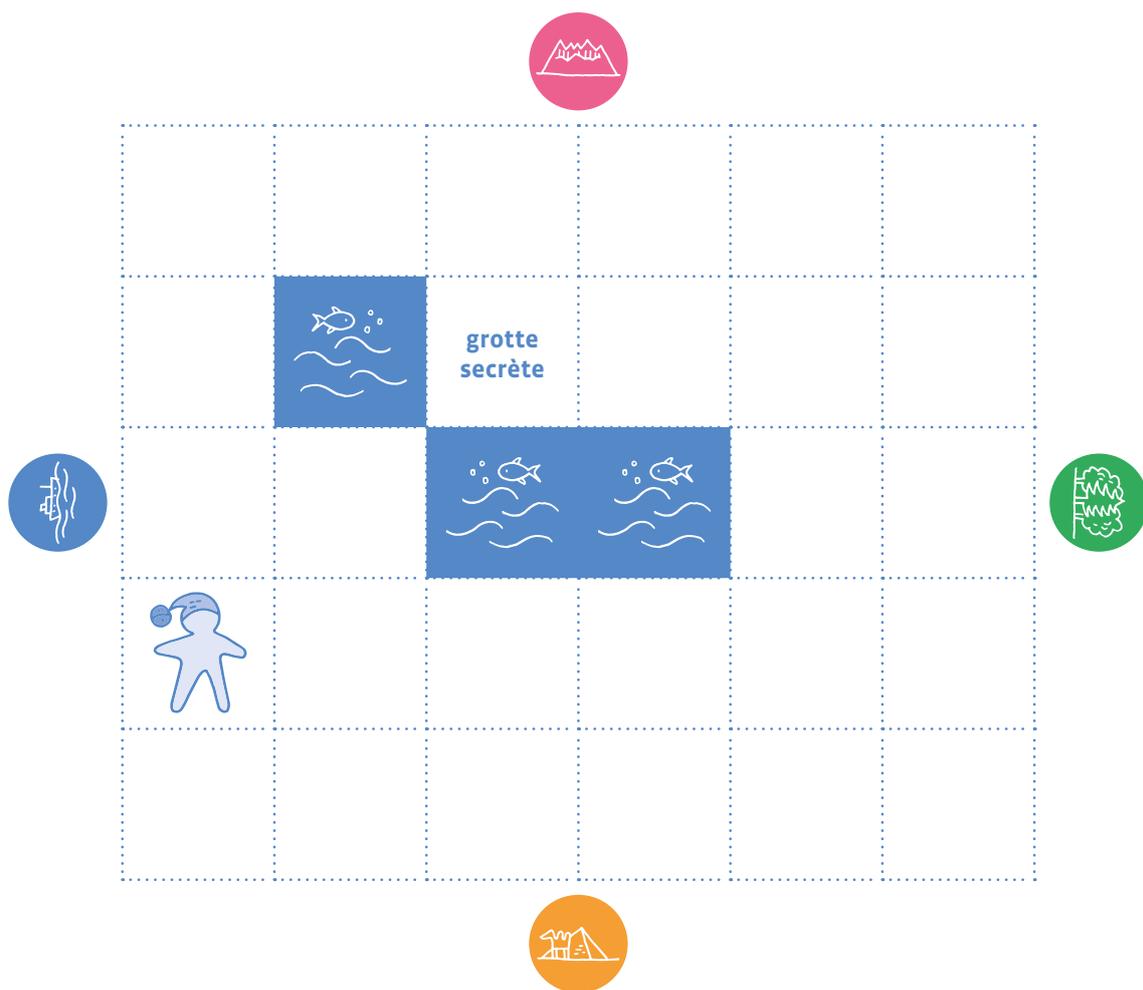
Modalités de travail: collectif

 **Durée:** 15 minutes

Le lutin se promène sur le quadrillage lorsqu'il trouve un panneau sur lequel il y a un programme pour atteindre la grotte secrète:



Où se trouve la grotte secrète dans le quadrillage?



Il souhaite maintenant revenir de la grotte secrète à l'endroit où il a trouvé le panneau.
Écrire la séquence avec des flèches (→ ← ↑ ↓).

Solutions:

- ↑ ← ← ↓ ↓ ↓
- 1 ↑ 2 ← 3 ↓

Temps 3.3: Exercice d'application et de bilan

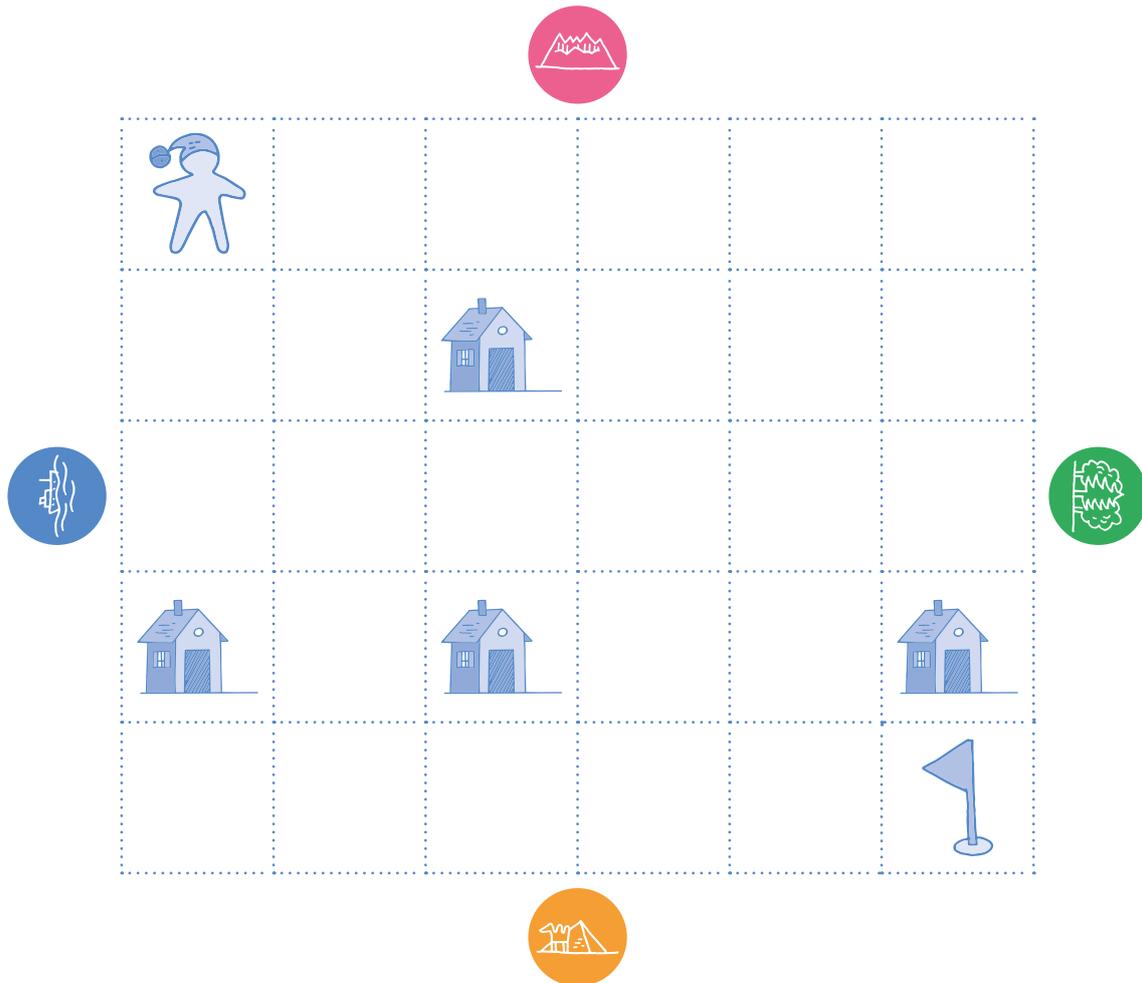
Modalités de travail: collectif

 **Durée:** 15 minutes

L'enseignant·e présente le même parcours, mais sur lequel ont été disposés des chalets:

Exercice d'application et de bilan

L'enseignant·e présente le même parcours, mais sur lequel ont été disposés des chalets:



Il s'agit de programmer le lutin pour qu'il atteigne la case *arrivée* tout en récoltant le plus de provisions possibles dans les chalets ouverts (en utilisant des tests, comme à la séance précédente).

Selon l'âge des élèves, cet exercice peut être proposé par petits groupes ou collectivement.

Une solution possible est:



Conclusion

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

- on peut écrire un programme de différentes façons.
- dans un programme, des boucles permettent de répéter plusieurs fois la même instruction.

On peut prévoir de reprendre la synthèse des élèves dans la trace écrite finale.

Par exemple :

Le lutin sur le quadrillage

- pour faire bouger votre lutin, il faut lui donner des **instructions**

→

↑

←

↓

- quand on met plusieurs instructions, on obtient un **programme**
- pour que le programme soit plus court, on peut faire des boucles

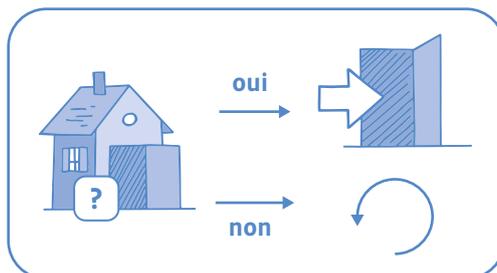
→ 3

Annexes

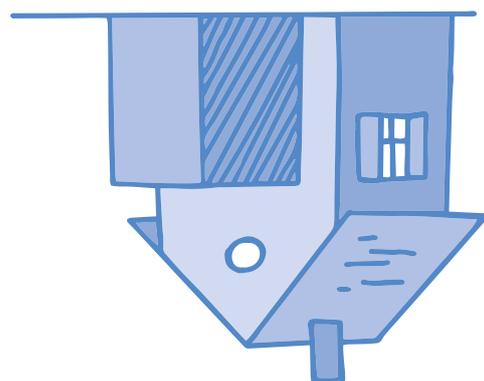
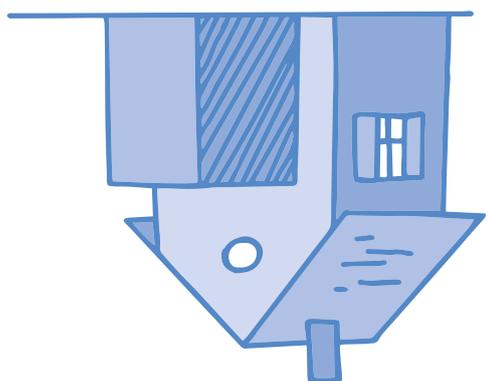
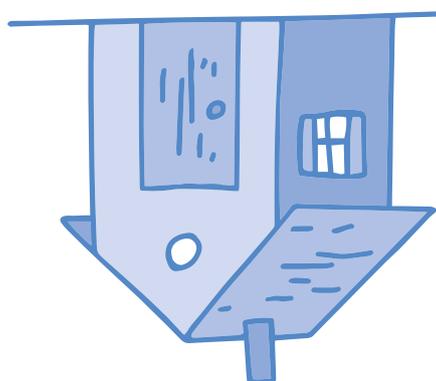
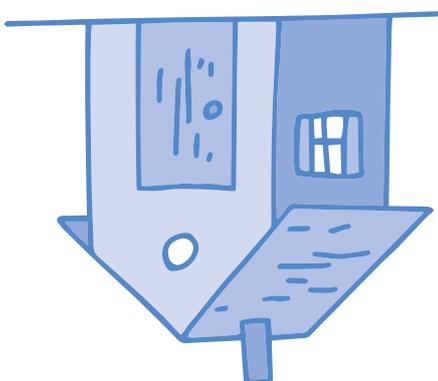
Alternative à la carte-instruction *si... alors...* de la séance 1-2

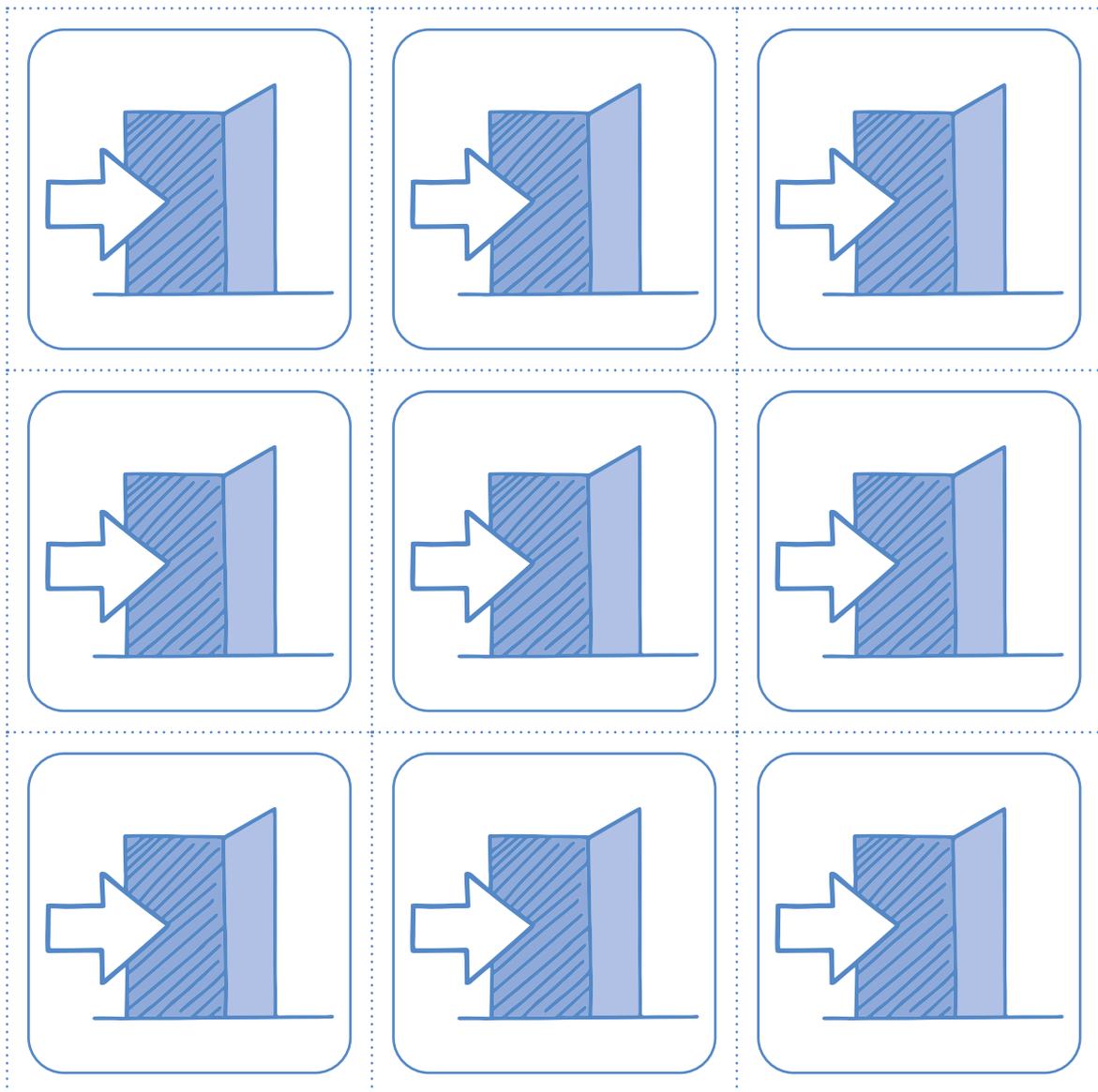
L'enseignant·e peut vouloir une version plus complète mais plus compliquée à manier :

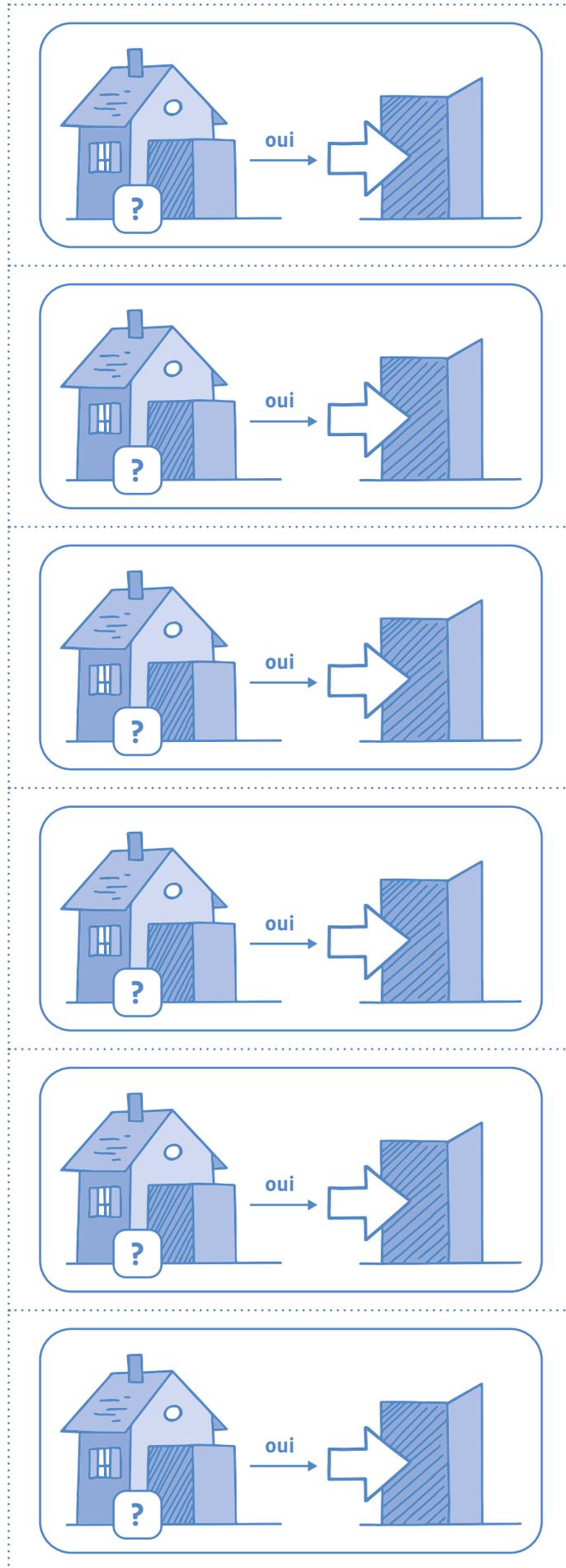
Si... alors... sinon... :



Cette carte signifie: *si le chalet est ouvert alors on entre sinon on revient au départ du parcours.*

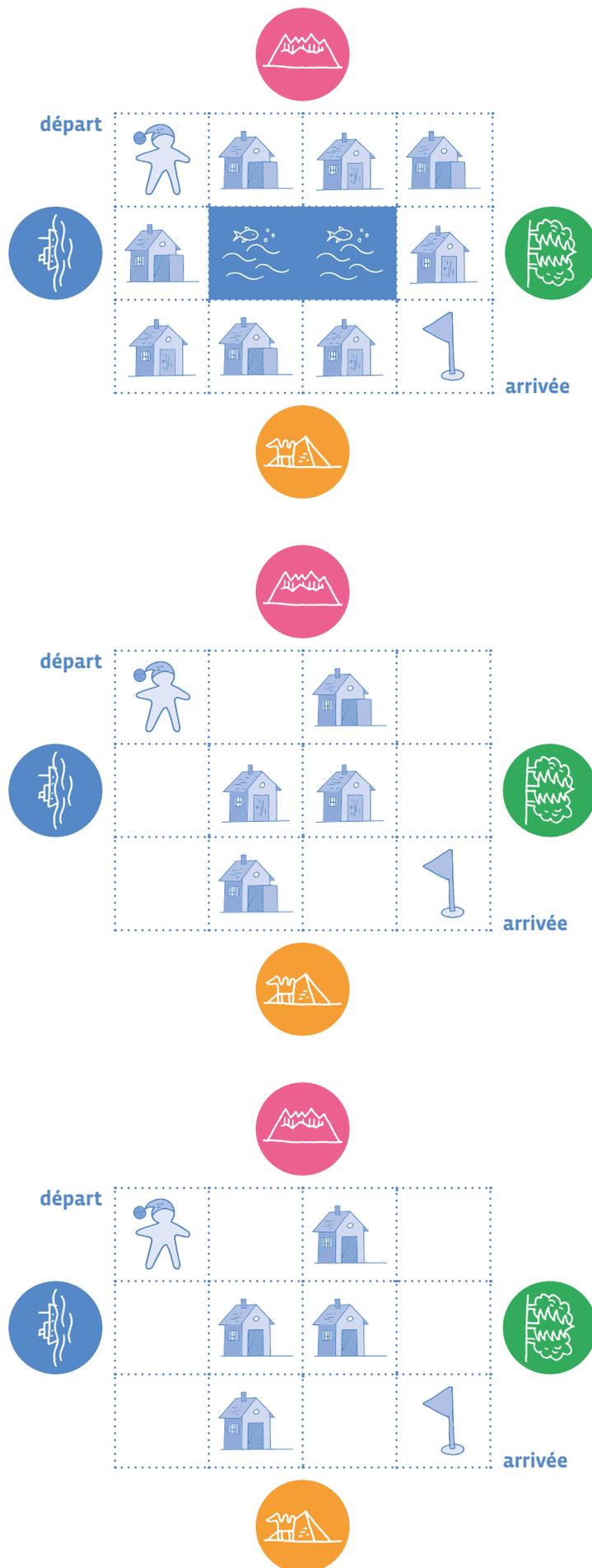






Les cartes-instructions pour programmer le lutin





The image contains three 4x4 grid puzzles. Each puzzle has a 'départ' (start) icon at (1,1) and an 'arrivée' (end) icon at (4,4). The obstacles are represented by icons in specific cells:

- Puzzle 1:** Empty grid.
- Puzzle 2:** Obstacles (blue water icons) at (2,2), (2,3), (3,2), and (3,3).
- Puzzle 3:** Obstacles (house icons) at (2,2), (3,1), (3,2), and (3,4).