Scénario 5 • SI • 6e

Retour vers la cité • Square CT

SI • 6e
Retour vers la cité • Square CT

Objectifs du Plan d'études romand (PER):

EN 22 – S'approprier les concepts de base de la science informatique…

2 … en encodant, décodant et en transformant des données
3 … en utilisant différentes machines et en découvrant le fonctionnement des réseaux
4 … en créant, en exécutant, en comparant et en corrigeant des programmes

Algorithmes et programmation

- Création et comparaison de programmes avec des séquences, des tests conditionnels et des boucles à l'aide d'un langage de programmation visuel pour résoudre des problèmes simples

Liens disciplinaires:

- MSN 22 – Nombres ; MSN 23 – Problèmes additifs, multiplicatifs ; MSN 25 – Modélisation
- SHS 21 – Relation Homme-Espace ; SHS 23 – Outils et méthodes de recherche

Intentions pédagogiques:

C'est en passant par la rédaction de programmes plus ou moins complexes que les élèves parviennent à comprendre comment une machine peut être programmée. Elles et ils vont apprendre qu'un programme est l'expression d'un algorithme dans un langage de programmation et qu'un algorithme est une succession d'étapes permettant de résoudre un problème, d'effectuer une tâche (et qu'il peut contenir des conditions, des tests ou encore des boucles). Les élèves vont découvrir des environnements de programmation utilisant un langage de programmation simple. Un langage de programmation permet de donner des consignes compréhensibles à la fois par la machine et par l'être humain. Les élèves vont créer des algorithmes et les mettre en oeuvre. Au travers de chacune de ces activités, les élèves exercent leur pensée computationnelle (ou Computational Thinking (CT) en anglais qu'on retrouve dans le titre du scénario!) grâce aux stratégies mises en oeuvre pour résoudre les situations: faire abstraction des critères des tapis, identifier les problèmes, formaliser un algorithme…

Description générale:

Les élèves programment des actions en enrichissant progressivement leur langage de programmation avec des instructions conditionnelles et des boucles qui sont en quelque sorte des concepts-outils rendant plus ou moins complexe un algorithme.

Les activités sont réalisées de manière débranchée. Pour des raisons liées à l'utilisation de l'espace, elles sont essentiellement conçues pour être menées en salle de gym bien qu'elles puissent se réaliser également en salle de classe. Elles utilisent comme support le kit Square CT[1]. Un fil conducteur permet de scénariser les différentes activités qui peuvent être ou non suivies par l'enseignante ou enseignant.

Il existe un scénario préalable en 5e, proposé avec ce matériel, permettant une progression pour les deux années (5e et 6e).

[1] Le kit Square CT (article n°50005343 disponible à la DAL) est composé d'une sacoche en tissu contenant 27 tapis de formes, de couleurs et de motifs différents ainsi que de leurs découpes respectives. Ces tapis sont créés dans un matériau écologique (EVA foam), adapté à la production dans les ateliers communautaires et collaboratifs régionaux (makerspaces).
Les images des tapis sont disponibles *via* ce lien court: [56-01-01].

À propos de l'utilisation des tapis :

Square CT est un ensemble d'activités progressives permettant aux élèves d'expérimenter et d'intégrer physiquement certaines notions de base de la science informatique grâce à des déplacements, des postures et autres activités corporelles. C'est pourquoi il est préférable de réaliser les différentes activités présentées dans ce document en utilisant les tapis Square CT disposés dans une salle assez grande. Toutefois, dans le cas où l'on ne pourrait pas avoir accès à une telle salle, ou dans le cas où l'on voudrait revenir rapidement sur une notion ou une activité, de petites cartes à imprimer et à découper représentant ces tapis dans une taille très réduite sont présentes à la fin de cette séquence (voir fiche 8), permettant par exemple de reproduire tous les parcours sur une table ou sur un tableau.

Dans ce scénario, les élèves vont devoir lire, créer et exécuter des algorithmes, c'est-à-dire des suites d'instructions ordonnées. Étant donné qu'elles et ils seront amenés à *exécuter* ces algorithmes, notamment en jouant le rôle d'une machine (les robots Blobs), nous utiliserons également le terme programme (un programme est la traduction d'un algorithme dans un langage de programmation de manière à ce que cet algorithme soit *compréhensible* et exécutable par une machine).
Nous utiliserons donc dans ce scénario les termes algorithmes et programmes sans réellement marquer de différences entre eux. Il n'est donc pas nécessaire d'insister ou de préciser la différence entre ces deux termes qui peuvent être utilisés l'un pour l'autre, sans distinction.

Mise en contexte :

En 2042, au-delà de frontières lointaines, des élèves accompagnés de leurs enseignantes et enseignants voyagent et découvrent de nouvelles villes inexplorées. De nombreuses maisons (les tapis) jonchent le sol et accueillent leurs investigations. L'exploration de ces maisons demande de leur part beaucoup de collaboration et de réflexion pour aborder ces nouvelles cités (CT) qui leur posent bien des problèmes à résoudre. Comment communiquer avec les Steams, ces personnages qui nous invitent à les rencontrer et les connaître ? Comment les aider à organiser leurs Square CiTies ?

Résumé du scénario :

En 5e, les élèves ont appris à identifier les maisons des habitants de Square CT et à communiquer avec eux en leur adressant des messages grâce à leurs robots Blobs. En 6e, les Steams sollicitent les élèves pour définir plus finement les critères de leurs maisons et les faire réaliser différentes actions liées notamment aux conditions qu'ils rencontrent.

Séance	Résumé	Matériel
1. Retrouvailles **Durée:** 45 minutes	• Vivre des instructions conditionnelles en associant les connecteurs logiques.	• kit Square CT (maisons) • papier, crayon • 1–2 cerceaux (ou tout autre élément permettant de délimiter une petite zone) • fiches 1.1 à 1.6 (matériel pour la classe) • fiche 1.7 (1 par élève)
2. Les saluts **Durée:** 45 minutes	• Transcrire les algorithmes de la séance 1 en les représentant sous forme de blocs puis de logigrammes. • Utiliser des boucles.	• algorithmes rédigés lors du temps 1.2 • fiches 2.1 et 2.2 (1 par élève) • kit Square CT (maisons) • papier, crayon
3. Livraison de colis **Durée:** 45 minutes	• Lire un programme texte et l'expérimenter sous différents rôles.	• kit Square CT (maisons) • fiche 3.1 (matériel pour la classe) • fiche 3.2 (1 par groupe) • 12 petits objets par équipe (cubes, jetons…)
4. Logigrammes **Durée:** 45 minutes	• Rédiger et exécuter des logigrammes.	• kit Square CT (maisons) • fiche 4.1 et/ou 4.2 (1 par élève) • fiche 4.3 (1 par élève)
5. Chasse au trésor (facultatif) **Durée:** 25 minutes	• Expérimenter un programme représenté en format texte et sous forme de logigramme qui intègre des instructions conditionnelles imbriquées.	• kit Square CT (maisons et Steams (= formes)) • fiches 5 à 7 (1 par groupe + enseignant·e)

Pour bien comprendre...
Présentation du matériel

Les critères

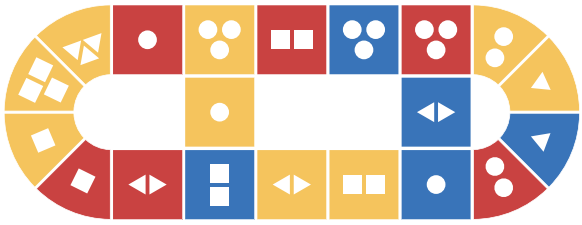
- 2 contours: arc ou dalle
- 3 couleurs: rouge (rose), bleu, jaune
- 3 formes de fenêtres: triangle, carré, rond
- 3 quantités de fenêtres: 1, 2 ou 3

Les éléments

Une **maison**: un tapis qui contient une ou plusieurs **fenêtres**

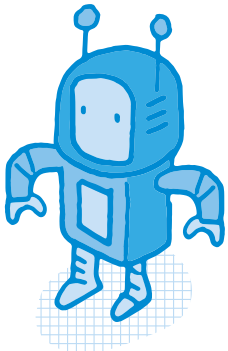
Une **rue** ordonnée: une suite de tapis avec des critères communs

Une **cité (ville) désordonnée**: des tapis réunis sans association de critères ou éléments communs



Les **Steams**: petites formes représentant les personnages des villes

Les **Blobs**: les robots qui sont simulés par les élèves



Séance 1
Retrouvailles

Résumé:
- Vivre des instructions conditionnelles en associant les connecteurs logiques.

Matériel:
- kit Square CT (maisons)
- papier, crayon
- 1–2 cerceaux (ou tout autre élément permettant de délimiter une petite zone)
- fiches 1.1 à 1.6 (matériel pour la classe)
- fiche 1.7 (1 par élève)

Temps 1.1: Décrire les maisons en variant les critères et les connecteurs logiques

Modalités de travail: en collectif

Durée: 20 minutes

Mise en route

Au préalable, on place les tapis Square CT en pile, positionnés à côté de soi et 1 cerceau au milieu de la salle. On peut également utiliser 2 cerceaux si on préfère séparer les cartes critères des cartes connecteurs (voir Moment 3: mise en commun). On utilise les cartes des fiches 1.1 à 1.5 (cartes critères, cartes connecteurs et le Message A).

Les élèves sont regroupés par équipes de 3 ou 4 élèves et répartis par équipe sur les côtés extérieurs (ou coins) de la salle.

Consigne: Vous vous rappelez, en 5e année, vous aviez rencontré les habitants de Square CT? Eh bien, j'ai reçu un nouveau message de leur part.

On lit le Message A (voir fiche 1.5):

> **Message A**
>
> Chers élèves, voilà bien longtemps que vous n'êtes pas venus nous rendre visite! Nous nous sommes rencontrés grâce à vos Blobs la dernière fois. Saurez-vous reconnaître nos maisons?

Moment 1: Distinguer les critères des maisons

On montre aux élèves les cartes critères imprimées et découpées.

Consigne: De quoi s'agit-il?

Il s'agit des cartes représentant les critères des maisons de Square CT. Certaines représentent des couleurs, d'autres des nombres, d'autres des formes.

On montre aux élèves les tapis Square CT en pile.

Consigne: À chaque tour, je vais vous montrer une maison différente. Pour remporter la maison, vous allez devoir montrer 3 cartes critères qui permettent de la décrire. L'équipe qui a remporté 3 maisons gagne la partie.

On montre aux élèves le cerceau placé au milieu de la pièce où on va y placer les cartes critères, faces cachées et on explique les règles du jeu.

Consigne: Je place maintenant ces cartes critères dans ce cerceau. À mon signal, un élève de chaque équipe va avancer vers ce cerceau, choisir deux cartes, regarder le critère représenté dessus. Elle ou il en choisira une seule, celle qui lui convient le mieux, et replacera alors, face cachée, celle qu'elle ou qu'il ne veut pas, puis rejoindra son équipe avec la carte sélectionnée. Lorsque votre équipe a 3 cartes qui correspondent à la maison en jeu, alors vous criez **SQUARE**, et nous vérifierons votre proposition. Si elle est correcte, vous gagnez la maison. Si elle est fausse, vous perdez une maison.

On donne un exemple, en montrant la maison  par exemple.

Consigne: Pour gagner cette maison, quelles cartes critères sont possibles?

Il s'agit des cartes **Rouge**, **2** et **Carré**.

On lance le jeu: au signal, chaque équipe envoie une ou un nouvel élève qui va devoir courir et retourner deux cartes critères, en choisir une et la ramener à son équipe. À chaque fois que l'on donne un signal, chaque équipe envoie une ou un nouvel élève. L'équipe qui crie **SQUARE** interrompt le jeu et la classe valide sa proposition de 3 cartes critères. À chaque fois qu'une équipe voit sa proposition validée, on présente une nouvelle maison. Continuer le jeu jusqu'à ce qu'une équipe ait remporté 3 maisons.

Moment 2: Se familiariser avec les connecteurs logiques

On montre aux élèves les cartes connecteurs imprimées et découpées (fiche 1.4).

Consigne: Ces nouvelles cartes représentent des connecteurs logiques. Vous allez devoir les utiliser pour connecter les critères que vous allez choisir, selon une certaine logique. Cette fois, pour gagner une maison, vous allez toujours devoir proposer 3 cartes, mais cette fois, l'une de ces 3 cartes devra être un connecteur logique.

On explique en montrant la maison  par exemple.

Consigne: En utilisant une carte connecteur, quels ensembles de 3 cartes sont possibles pour gagner cette maison?

Cette fois, plus de propositions sont valables: carré et rouge, carré ou rouge, carré ou bleu, carré et deux, rouge ou un, etc.

On place les cartes connecteurs, faces cachées dans le cerceau et on mélange toutes les cartes.

Consigne: Chaque fois qu'une ou un des élèves de chaque équipe viendra au centre, elle ou il devra toujours piocher 2 cartes et n'en garder qu'une.

On reprend le jeu (avec les mêmes règles que pendant le moment 1, à ceci près que cette fois 3 cartes dont une carte connecteur doivent être maintenant compilées pour gagner la maison en jeu).

Moment 3: Mise en commun

On demande aux élèves ce qu'elles ou ils ont remarqué:

On a beaucoup plus de chance de gagner une maison si on utilise le connecteur **OU** que si on utilise le connecteur **ET**. En effet, avec les mêmes cartes de critères – par exemple le critère **rouge** et critère **rond** –, avec l'opérateur **OU**, toutes les maisons rouges ainsi que toutes les maisons aux fenêtres rondes - et bien sûr toutes les maisons rouges aux fenêtres rondes – respectent la condition.
Par contre, avec le connecteur **ET**, la condition n'est vraie uniquement si les 2 critères sont respectés (dans cet exemple, uniquement les maisons rouges aux fenêtres rondes).

Dans le jeu, si on pioche une carte critère qui correspond à la maison et ensuite une carte **OU**, on peut piocher n'importe quelle autre troisième carte critère. On gagnera de toute façon la maison si personne ne crie avant.

On demande ensuite aux élèves d'énumérer quelques combinaisons possibles (critères et connecteurs) pour une maison en utilisant le **ET** et en utilisant le **OU**.

Pour faciliter le jeu, on pourra proposer une étape intermédiaire permettant de découvrir la valeur des différents connecteurs. Donner à toutes les équipes un connecteur (d'abord le **OU**) et les élèves courent chercher une carte, puis une deuxième. Avec ce connecteur **OU**, il est probable que certaines équipes se rendent compte que, si la 1re carte critère qu'ils ont piochée correspond à la maison, les élèves n'auront pas vraiment à réfléchir au moment de choisir la 2e et auront donc tout intérêt à courir le plus vite possible, choisir n'importe quelle 2e carte et revenir dans leur équipe pour pouvoir crier **SQUARE**: en effet, si la maison est rouge avec des fenêtres carrées, en utilisant le connecteur **OU**, si une équipe a tiré comme 1re carte une carte rouge, ils savent que même si la 2e carte qu'elles et ils choisissent est bleue, leur proposition sera validée (une maison rouge avec des fenêtres carrées est bien **rouge ou bleue**).

On donne ensuite une carte connecteur **ET** à chacune des équipes. Les possibilités deviennent alors beaucoup plus limitées, car il faut que chacune des 2 cartes corresponde à la maison.

On peut également disposer les 2 catégories dans 2 cerceaux, un bleu et un rouge: dans le cerceau bleu, on placera les cartes critères et dans le cerceau rouge, les cartes connecteurs.

Comme en 5e, ce scénario va permettre de mettre en évidence la notion de modularité, en essayant différents choix de maisons. Les élèves vont ainsi tester différents algorithmes et prolonger les concepts abordés en 5e par l'ajout des connecteurs **ET** et **OU**.

Temps 1.2: Créer et utiliser des instructions conditionnelles

Modalités de travail: en collectif

Durée: 25 minutes

Durant le temps 1.1, les élèves ont abstrait des critères et découvert les connecteurs logiques, ce qui va leur permettre de maintenant travailler autour de l'un des composants de bases de l'algorithmie, à savoir les instructions conditionnelles.

Pour ce temps, les tapis sont disposés en cercle dans la salle et on utilise les cartes des fiches 1.5 et 1.6 (le Message B et les cartes saluts).

Mise en route

On lit le Message B aux élèves (voir fiche 1.5):

Message B

Chers élèves, vous semblez bien vous repérer dans notre Square CT! Toutes nos félicitations! Nous aimerions mieux vous connaître... On peut commencer par se saluer! Chez nous c'est difficile, chaque maison a sa propre manière de dire bonjour. Par exemple, les habitants des maisons bleues et 2 saluent en levant les 2 bras... Saurez-vous à votre tour nous saluer?

On explique alors que les règles de politesse semblent très différentes sur Square CT que sur Terre...:

Consigne: Sur Terre, lorsque nous sommes invités chez quelqu'un et que l'on arrive chez elle ou chez lui, on va dire **Bonjour**, **Bonsoir**, ou **Salut**, selon certains critères d'âge, de familiarité et du moment de la journée. Les Steams ont d'autres coutumes: ils saluent les personnes avec des mouvements très spécifiques, liés à leur type de maison.

On montre les exemples de saluts de la fiche 1.6, par exemple:

Consigne: À quoi correspond cette carte? Que nous indique-t-elle de faire?

On fait verbaliser que cette carte nous indique une manière de saluer selon des critères (couleur de la maison, nombre ou forme des fenêtres) et un connecteur logique. Selon la maison sur laquelle on se trouve, on va donc devoir effectuer le geste de salut indiqué.

Moment 1: Exécuter les algorithmes des saluts, en réalisant des tests

On dispose les maisons en cercle.
Les élèves se placent sur la maison de leur choix.
On (ou une ou un élève) choisit une carte saluts. On lit la carte. Au signal donné, les élèves qui sont situés sur une maison qui correspond bien aux critères indiqués réalisent l'action indiquée. Les autres élèves ne font rien.
On réalise plusieurs fois l'activité avec les différentes cartes en changeant de rôle ou de maison.
On amène alors les élèves à mettre dans l'ordre les étapes de réflexion:

Consigne: Quelle réflexion avez-vous effectuée dans votre tête pour être sûrs de pouvoir par exemple lever les 2 bras?

Pour évaluer si la condition est vraie ou fausse, on va devoir se poser des questions…

1. Si le connecteur logique de la carte saluts présentée est **ET** (par exemple s'il s'agit de la carte Saluts représentée plus haut - Si bleue et 2 fenêtres alors lever 2 bras), l'élève se pose tout d'abord la question: est-ce que la maison sur laquelle je suis est bleue?
   - Si la réponse est **NON**, l'élève sait directement que la condition est fausse (car la condition est vraie uniquement si les 2 critères sont vérifiés. Comme le 1er ne l'est pas, la condition ne peut être vraie). L'élève ne lève donc pas les bras.
   - Si la réponse est **OUI**, alors les élèves doivent se poser une nouvelle question (car on utilise le connecteur logique **ET**, donc les 2 critères doivent être vérifiés (ici, la maison doit être bleue **ET** avoir 2 fenêtres), à savoir: est-ce que la maison sur laquelle je suis a deux fenêtres?
     - Si la réponse est encore une fois **OUI**, alors on dit que la condition est vraie, l'élève lève les 2 bras.
     - Si la réponse est **NON**, on dit que la condition est fausse, l'élève ne lève pas les bras.
2. Dans le cas où le connecteur logique de la carte saluts présentée est **OU**, par exemple *Si fenêtres rondes ou 3 fenêtres alors se mettre à genoux* les élèves vont tout d'abord se poser la question *est-ce que la maison sur laquelle je suis a des fenêtres rondes?*
   - Si la réponse est **OUI**, elles et ils n'ont pas à se poser de question supplémentaire, car la condition est forcément vraie (le 1er critère est vérifié et comme on utilise le connecteur logique **OU**, il suffit que l'un des critères soit vérifié pour que la condition soit vraie), les élèves doivent donc se mettre à genoux.
   - Si la réponse est **NON**, elles et ils doivent par contre se poser une 2e question avant de savoir si elles ou ils vont ou pas se mettre à genoux: *est-ce que la maison sur laquelle je suis a 3 fenêtres?*
     - Si la réponse est encore une fois **OUI**, alors la condition est vraie et les élèves doivent alors se mettre à genoux.
     - Si la réponse est **NON**, alors la condition est fausse et les élèves ne doivent pas se mettre à genoux.

Cependant, sait-on quoi faire si la proposition est fausse?

Pour l'instant, on a utilisé des instructions conditionnelles de type **SI-ALORS**. Ces instructions indiquent ce qu'il faut faire quand une condition est vraie: **SI** cette condition est vraie, **ALORS** on fait ceci. Si la condition n'est pas vérifiée, on ne fait rien.

Mais on peut aussi créer des instructions conditionnelles qui permettent d'indiquer ce qu'il faut faire si la condition n'est pas vérifiée : cette fois, ces instructions conditionnelles auront une structure du type : **SI-ALORS-SINON**.

Par exemple : **SI** tu es sur une maison jaune **ET** fenêtres rondes, **ALORS** lève les 2 bras, **SINON** assieds-toi.

Dans la vie de tous les jours, on désire parfois faire quelque chose uniquement si une condition est vraie. Par exemple : *S'il pleut dehors, alors je prends mon parapluie*. En informatique, on peut également indiquer à la machine de ne faire quelque chose (exécuter certaines instructions spécifiques) uniquement si une condition est vérifiée.

Avec cette activité, on introduit cet élément de base de la programmation que sont les instructions conditionnelles. Comme son nom le suggère, une instruction conditionnelle permet d'exécuter une ou plusieurs instructions uniquement dans le cas où une condition est vérifiée (avec une structure du type **SI-ALORS**) et éventuellement une ou plusieurs instructions dans le cas où cette condition n'est pas vérifiée (on utilisera alors une structure de type **SI-ALORS-SINON**)

Il est donc important de bien reprendre cette structure **SI-ALORS** dans chaque formulation après avoir tiré une carte critère, par exemple :

- Si c'est une maison avec des fenêtres rondes, alors levez la main (critère formes)
- Si la maison est rouge, alors levez la main (critère couleurs)
- Si la maison a 2 fenêtres, alors levez la main (critère nombres)

Moment 2 : Rédiger des instructions conditionnelles

On fait remarquer que les actions proposées par les Steams ne sont pas très créatives.

On répartit les élèves par petits groupes et on leur propose de créer elles-mêmes et eux-mêmes leurs instructions conditionnelles de type **SI-ALORS-SINON** permettant d'indiquer de nouvelles manières de saluer (par exemple *Si la maison est rouge et a des fenêtres rondes, **alors** taper 3 fois dans les mains, sinon taper 3 fois sur ses cuisses*). Leur laisser libre choix dans leur rédaction sur des feuilles libres.

On reprend le jeu présenté durant le Moment 1 en utilisant ces nouvelles instructions conditionnelles réalisées par les élèves. À chaque fois, on demande aux élèves :

- Quelle est la condition ? (Si c'est une maison jaune et fenêtres rondes par exemple)
- Je teste en vérifiant les critères de ma maison
- C'est vrai : alors je lève les 2 bras
- C'est faux : alors je m'assois

Consigne : Est-ce qu'il y a des instructions conditionnelles qui ne peuvent jamais être vérifiées ?

Oui, celles qui utilisent le connecteur logique **ET** avec deux critères de la même catégorie (couleur, nombre ou forme) différents. En effet, aucune maison ne peut être rouge **ET** bleue, ou avoir des fenêtres rondes **ET** carrées, etc.

On récupère et on garde les productions pour la séance suivante.

Moment 3 : Mise en commun et institutionnalisation

On distribue à chaque élève la fiche 1.7 que l'on complète en collectif (voir fiche 1.7 – corrigé).

Séance 2
Les saluts

Résumé :
- Transcrire les algorithmes de la séance 1 en les représentant sous forme de blocs puis de logigrammes.
- Utiliser des boucles.

Matériel :
- algorithmes rédigés lors du temps 1.2
- fiches 2.1 et 2.2 (1 par élève)
- kit Square CT (maisons)
- papier, crayon

Temps 2.1 : Rédiger des programmes sous différents formats : texte, blocs et logigramme

Modalités de travail : en collectif

Durée : 30 minutes

Moment 1 : Rédaction sous format texte et blocs

Au préalable, on dispose des tapis en ligne (par exemple 5 lignes de 5 tapis).

On distribue les algorithmes réalisés lors du Moment 2 du temps 1.2 de la séance 1.

On propose alors de reprendre les algorithmes en les réécrivant de manière à ce que les mots SI, Alors et Sinon soient alignés à gauche. Si les élèves le souhaitent ou si les idées d'actions sont pauvres, elles et ils peuvent également en profiter pour améliorer leurs programmes en inventant de nouveaux gestes (par exemple : lever les mains en l'air, toucher le sol avec ses mains, tourner sur soi-même, faire un pas sur le côté, avancer de 2 maisons, etc.). Par exemple :

SI maison jaune **OU** maison rouge
ALORS lever les 2 bras
SINON s'asseoir

SI c'est une maison de couleur jaune **ET** avec des fenêtres rondes
ALORS lever les 2 bras
SINON tourner sur soi-même

Dans de nombreux langages de programmation, certaines instructions sont plus ou moins décalées vers la droite (elles sont indentées) pour indiquer qu'elles font partie d'une instruction conditionnelle donnée.

Par exemple, , nous écrirons ainsi :

SI maison jaune **ET** fenêtres rondes
ALORS tourner sur soi-même
SINON s'asseoir

La suite de ce document reprend cette manière de faire, qui s'avérera essentielle lorsque les instructions conditionnelles imbriquées seront abordées par la suite.

Pour comprendre l'importance de l'indentation, voici deux algorithmes composés d'instructions parfaitement identiques, mais qui diffèrent dans la manière dont ces instructions sont indentées:

SI maison jaune
ALORS tourner sur soi-même
SINON s'asseoir
SI maison avec 3 fenêtres
ALORS taper dans les mains
SINON claquer des doigts
 	
Avancer d'une maison
	SI maison jaune
	  ALORS tourner sur soi-même
	  SINON s'asseoir
	  SI maison avec 3 fenêtres
	  ALORS taper dans les mains
	  SINON claquer des doigts
	
	Avancer d'une maison

Ces deux algorithmes, s'ils étaient exécutés par une machine (ou par une ou un élève jouant le rôle d'un Blob, voir séance 3) produiraient 2 résultats différents:

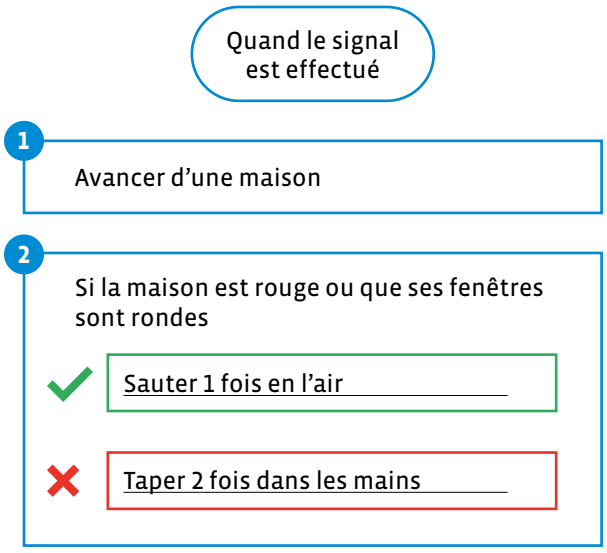
- Si une machine exécutait le programme de gauche, elle vérifierait tout d'abord la couleur de la maison sur laquelle elle se trouve et exécuterait l'instruction correspondante (Tourner sur soi-même si la maison est jaune, s'asseoir si elle ne l'est pas). Ensuite, si cette même maison à 3 fenêtres, la machine taperait dans *ses mains* (si elle en avait!), sinon elle claquerait des doigts. Pour finir, elle avancerait d'une maison.
- Par contre, si cette même machine devait exécuter le programme de droite, elle commencerait de la même manière en vérifiant tout d'abord si la couleur de la maison sur laquelle elle se trouve est jaune. Si c'est le cas, elle tournerait pareillement sur elle-même. Mais contrairement au programme de gauche, elle n'évaluerait pas le nombre de fenêtres: elle avancerait ensuite directement d'une maison. Par contre, quelle que soit la couleur de la maison, elle finirait toujours pas avancer d'une maison à la fin du programme (cette dernière instruction n'est pas indentée).

Moment 2: Rédaction sous format blocs

On distribue la fiche 2.1 à chaque élève. On invite les élèves à faire l'exercice 1: écrire son programme sous forme de blocs de cette fiche, à savoir reprendre le programme qu'elles et ils auront précédemment écrit sous forme de phrases en inscrivant ces dernières dans des cases que l'on va appeler des blocs.

On donne un exemple au tableau en spécifiant que les instructions sont encadrées avec différentes couleurs ou différents signes pour que le programme soit plus facilement lisible et compréhensible. Le cadre vert avec un signe de validation correspond à l'instruction à effectuer si la condition est vraie (ALORS), le cadre rouge avec une croix à celle à effectuer si la condition est fausse (SINON).

Exemple pour rédiger les blocs:



Moment 3: Exécution des programmes

Pour tester leur compréhension, les élèves vont maintenant devoir *exécuter* leur programme.

Dans cette partie, les élèves mettent à l'épreuve les programmes rédigés en testant les instructions données. Le programme fait-il ce qui est attendu? Si non, pourquoi?

Consigne: Vous allez maintenant devoir *exécuter* votre programme. À chaque fois que j'effectuerai le signal que nous venons de définir, chacun d'entre vous devra exécuter son programme. Quand vous serez sur la dernière maison, vous irez vous placer sur le côté.

On désigne différents élèves (autant d'élèves que de lignes de tapis que l'on aura créées au préalable).

Chacune ou chacun de ces élèves se positionne devant une ligne de tapis (1 élève devant chaque ligne) en gardant dans ses mains son programme sous forme de blocs qu'elle ou il a complété pendant le Moment 2.

On définit un signal avec ses élèves: taper des mains, son du tambourin...

Quand les élèves arrivent sur la dernière maison de leur ligne de tapis, on désigne de nouvelles et nouveaux élèves et répète l'exercice.

Une fois que tous les élèves ont exécuté leur programme, on leur demande: *D'après vous, quel est l'intérêt d'avoir une instruction **avancer d'une maison** au début de notre algorithme?*

Avec cette instruction, on instaure la notion de répétition et de déplacement: ainsi, en gardant le même algorithme, le résultat (l'action à effectuer ou non) ne sera pas le même selon la maison sur laquelle on se trouvera.

Note: il est possible pour complexifier d'avoir à lire et à exécuter 2 instructions conditionnelles à la suite pour chaque maison (voir illustration de droite ci-dessous).

Quand le signal est effectué

1. Avancer d'une maison
2. Si la maison est jaune
   - ✓ Taper 3 fois dans ses mains
   - ✗ Taper 3 fois sur ses cuisses
3. Si _____
   - ✓ _____
   - ✗ _____

Quand le signal est effectué

1. Avancer d'une maison
2. Si la maison est jaune
   - ✓ Taper 3 fois dans ses mains
   - ✗ Taper 3 fois sur ses cuisses
3. Si une ou des fenêtres sont rondes
   - ✓ Sauter 1 fois à pieds joints
   - ✗ Lever les 2 mains au-dessus de la tête

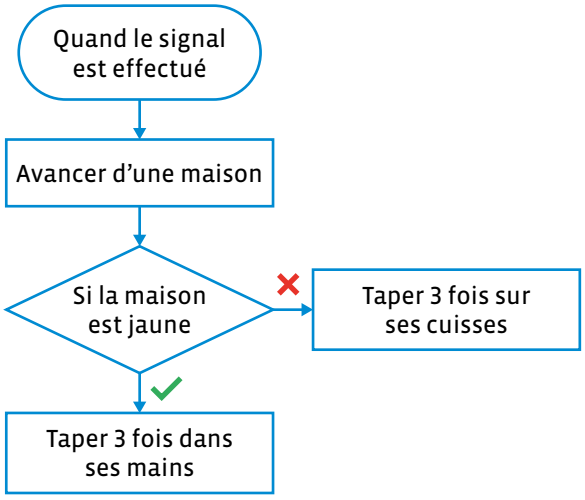
Moment 4: Le logigramme

Consigne: Nous allons maintenant représenter nos algorithmes sous forme de schémas. Les informaticiennes et les informaticiens nomment ces schémas, qui représentent visuellement des algorithmes, des logigrammes.

Au tableau, on reprend les différents symboles qu'utilisent les informaticiennes et les informaticiens pour représenter leurs algorithmes sous forme de logigramme:

- Un ovale indique la première et dernière étape: on peut écrire **Début** et **Fin** dans ces ovales.
- Un rectangle indique les différentes étapes, les différentes instructions, ou groupes d'instructions.
- Un losange indique une instruction conditionnelle: généralement, si la condition est fausse, nous suivons le **chemin** de droite, sinon celui du dessous. On propose aux élèves d'écrire **Vrai** et **Faux**, ou d'utiliser deux symboles différents pour indiquer les deux **chemins** (celui à emprunter si la condition est vraie et celui à suivre si la condition est fausse).
- Les flèches indiquent le sens de lecture

On crée un exemple collectivement:



Comme trace écrite, on demande aux élèves de créer leur propre logigramme représentant le programme qu'elles et ils auront rédigé en bloc ou en texte lors du Moment 2 ou du Moment 3.

On demande aux élèves de compléter l'exercice 2: texte à compléter de leur fiche (fiche 2.1).

Correctif de l'exercice 2:
Le logigramme est un outil utilisé par les informaticiennes et les informaticiens pour présenter leurs **ALGORITHMES**. Les instructions sont écrites dans des **RECTANGLES**, les conditions dans des **LOSANGES**.

En réalisant des tests, on peut savoir si la condition est **VRAIE** ou **FAUSSE**.
Les **FLÈCHES** permettent de visualiser l'ordre, le sens d'exécution des instructions de l'algorithme.

Temps 2.2: La chorégraphie des saluts – Lire et exécuter des programmes – Les boucles

Modalités de travail: en collectif, en classe ou en salle de gym

Durée: 15 minutes

Dans un programme, comme dans la vie de tous les jours, il faut souvent faire des choses répétitives. Lorsque les informaticiennes et informaticiens veulent répéter plusieurs fois les mêmes instructions, elles et ils utilisent des boucles. La syntaxe et le mode de fonctionnement de ces structures de programmation varient selon les langages de programmation, mais il en existe deux grands types:

- Des boucles permettant de répéter des instructions tant qu'une condition est vraie. C'est ce type de boucle que nous allons utiliser dans ce scénario, on les appelle les boucles **Tant que**. Par exemple: répète ces instructions tant que tu vois encore au moins une maison devant toi.
- Des boucles permettant de répéter des instructions un nombre défini de fois. Par exemple: répète ces instructions 10 fois.

Moment 1: Les boucles

On explique aux élèves que pour effectuer une chorégraphie, la même séquence d'instructions va devoir être exécutée à l'identique pour chaque maison par toute la classe ou par groupe.

Consigne: Nous aimerions trouver une manière de représenter cette répétition. Quelles seraient les différentes possibilités?

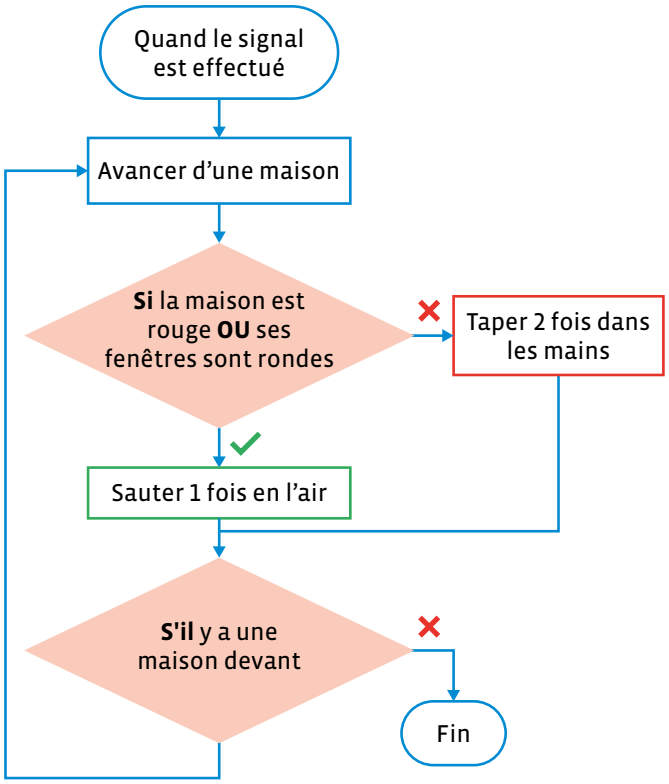
- On compte le nombre de maisons et on répète le nombre de fois nécessaire.
- On répète tant qu'il y a une maison devant, c'est-à-dire répéter la même séquence d'instructions jusqu'à la dernière maison.
- On copie les mêmes instructions plusieurs fois.

Consigne: Quand les informaticiennes et les informaticiens ne veulent pas avoir à écrire plusieurs fois la même série d'instructions, ils utilisent une boucle. Nous allons maintenant reprendre le petit algorithme que vous avez créé précédemment, mais comme nous ne voulons plus avoir besoin de répéter un événement - le signal que j'effectuais - pour répéter notre algorithme, nous allons schématiser une boucle sur notre logigramme. Est-ce que certaines et certains d'entre vous ont une idée sur la manière dont on va pouvoir représenter cette répétition sur notre logigramme en utilisant les symboles que nous avons vus précédemment?

En utilisant une flèche qui revient à une instruction précédente.

On demande alors à chaque élève de compléter le logigramme de la fiche 2.2 en utilisant comme point de départ le logigramme qu'elles et ils auront réalisé pendant le temps 2.1.

Consigne: Dans le logigramme que vous venez de compléter, nous remarquons une boucle «Tant que» (tant qu'il y a une maison devant, on va répéter certaines instructions qui ont déjà été effectuées). Il y a d'autres types de boucles: par exemple, nous pourrions également écrire un algorithme où nous indiquerions le nombre de répétitions (par exemple: répéter 5 fois). La boucle «Tant que» permet de ne pas modifier le logigramme même si nous modifions le nombre de maisons: quel que soit le nombre de maisons, notre algorithme se répétera jusqu'à la dernière.



On récupère les logigrammes réalisés par les élèves.
Dans l'exemple présenté ci-contre:

- Un ✓ et un cadre de la même couleur sont utilisés pour indiquer le chemin à suivre et l'instruction à exécuter quand la condition est vraie.
- Un ✗ et un cadre de la même couleur sont utilisés pour indiquer le chemin à suivre et l'instruction à exécuter quand la condition est fausse.

Pour schématiser une boucle **Tant que**, on utilise une condition (s'il y a une maison devant) dont la sortie **vraie** ramène à l'instruction **Avance d'une maison** et la sortie **fausse** pointe vers le symbole de fin de l'algorithme.

Moment 2: La chorégraphie algorithmique

On dispose préalablement les maisons les uns à la suite des autres, pour créer des **rues** en variant les critères (voir illustration ci-dessous).

On distribue à chaque élève l'un des logigrammes réalisés pendant le Moment 1.

Les élèves sont répartis en groupes au début de chaque rue, en file indienne (autant de groupes d'élèves que de rues)

Au signal que l'on donne, la ou le 1er élève de chaque groupe exécute son programme. Quand tous les élèves sont arrivés sur la dernière maison de leur rue, les prochaines ou prochains dans la file exécutent leur programme, et ainsi de suite jusqu'à ce que tous les élèves soient passés.

Séance 3
Livraison de colis

Résumé:
- Lire un programme texte et l'expérimenter sous différents rôles.

Matériel:
- kit Square CT (maisons)
- fiche 3.1 (matériel pour la classe)
- fiche 3.2 (1 par groupe)
- 12 petits objets par équipe (cubes, jetons...)

Temps 3.1: Comprendre un programme texte et son indentation

Modalités de travail: en collectif, puis en groupes (3-4 élèves)

Durée: 20 minutes

Mise en route

On lit le Message C à ses élèves:

Message C

Chers élèves,

Pas de spectacle aujourd'hui, car nous subissons actuellement un pic de pollution. Pouvez-vous nous aider en nous livrant des vivres? Nous vous conseillons d'utiliser vos Blobs machines car l'air est irrespirable! Nous avons créé des programmes qu'elles devront exécuter...

On affiche la carte Programme 1 livraison.

```
Avancer d'une maison
Si maison avec fenêtre vide
    alors: livrer un colis
Si 2 colis ont été déposés en tout
    alors: retourner à la station (fin du programme)
    sinon: recommencer le programmee
```

Moment 1

Consigne: D'après vous, à quoi correspond ce message?

Il s'agit d'un programme permettant de déposer les colis.

Consigne: Pouvez-vous le traduire en langage plus naturel, par exemple si vous deviez expliquer la marche à suivre à une personne?

Il faut avancer de maison en maison, si la maison sur laquelle tu es a une fenêtre vide tu peux livrer un colis. Dès que tu as déposé deux colis, tu retournes à la station et tu as fini.

Consigne: Que remarquez-vous en ce qui concerne la mise en page?

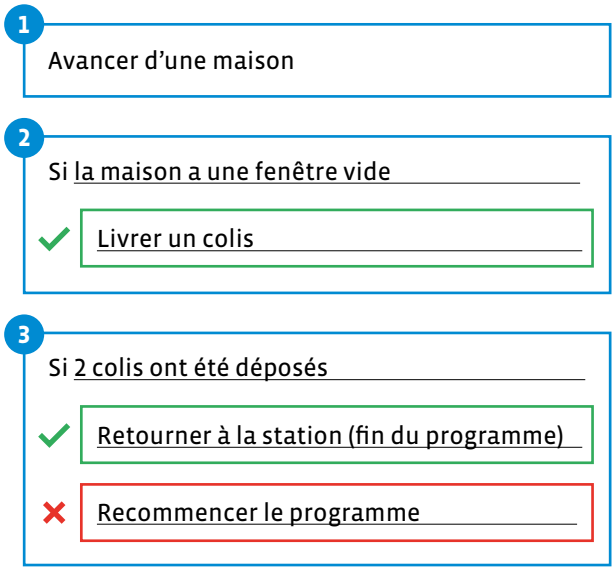
Certaines phrases - celles qui commencent par alors et sinon - sont un peu décalées vers la droite.

Consigne: Comment pourrait-on représenter ce programme pour qu'il soit plus lisible? (rappel de la séance précédente)

En utilisant une représentation par bloc pour structurer le programme et des couleurs pour rendre les branchements (si/sinon) plus visibles.

On réalise au tableau la rédaction du programme en blocs.

Exemple de programme:



Pour conclure, on réalise un exemple commun pour faciliter la mise en œuvre sur le parcours, notamment la description des rôles:

- Chaque équipe détient 3 programmes différents numérotés 1 (Steam triangle bleu), 2 (Steam rond jaune) et 3 (Steam carré rouge). Chaque équipe a également une douzaine de petits objets, les colis qui devront être livrés aux Steams.
- Dans chaque équipe, il va y avoir l'élève-ordinateur, l'élève-robot Blob et l'élève-contrôleur. À chaque tour, les rôles sont échangés: l'ordinateur deviendra robot, le robot deviendra contrôleur et le contrôleur devient ordinateur.
- À chaque tour, le programme lu par l'élève-ordinateur change.
- L'élève-robot Blob se place sur le 1er tapis du parcours. L'élève-ordinateur et l'élève contrôleur se placent à côté du parcours. L'élève-ordinateur tourne le dos à l'élève-robot (il ne le voit pas).
- L'élève-robot Blob va devoir déposer ses deux colis sur les fenêtres vides. Attention, le Blob ne prend pas de décision, il obéit aux instructions. C'est l'élève-ordinateur qui lit et reformule chaque instruction pour le robot Blob. Ce dernier exécute les instructions et vérifie si les conditions sont vraies ou fausses en faisant des signes de la tête (voir exemple ci-dessous). L'élève-contrôleur va transmettre les réponses de l'élève-robot à l'élève-ordinateur, assurera la synchronisation ordinateur-robot (par exemple en demandant à l'ordinateur d'attendre si le robot n'a pas fini d'exécuter une instruction) et vérifiera qu'ordinateur et robot ne fassent pas d'erreur.

Exemple de décomposition des tâches, avec le programme 1:

- L'élève-ordinateur lit (intérieurement) la 1re instruction **Avancer de deux maisons**
- Il dit à haute voix **Avancer de deux maisons** à l'intention de l'élève-robot
- Ce dernier avance de deux maisons
- Quand ce dernier a effectué ce déplacement, l'élève-contrôleur peut dire **ok** pour lui signifier que l'élève-robot a bien fini d'exécuter l'instruction.
- L'élève-ordinateur lit (intérieurement) l'instruction conditionnelle **Si maison avec fenêtre vide alors: livrer un colis**. Il reformule à haute voix **Est-ce que la maison sur laquelle tu es à une fenêtre vide?**
- L'élève-robot vérifie et fait un signe de la tête (oui ou non selon le cas)
- L'élève-contrôleur indique (oralement) à l'élève-ordinateur ce qu'a répondu l'élève-robot. Par exemple **Oui, il y a une fenêtre vide**.
- L'élève-ordinateur dit alors (à haute voix) **Alors livre un colis**

- L'élève-ordinateur lit (intérieurement) l'instruction conditionnelle **Si 2 colis ont été déposés en tout alors: retourner à la station (fin du programme) sinon: recommencer le programme**
- L'élève-ordinateur demande (à haute voix) **Est-ce que tu as livré 2 colis?**
- L'élève-robot fait **Oui** ou **Non** d'un signe de la tête, que l'élève-contrôleur indique (oralement) à l'ordinateur.
- Si l'élève-contrôleur dit **Oui**, alors l'élève-ordinateur dit **Retourne à la station** et l'élève-robot retourne à la station. Si l'élève-contrôleur dit **Non**, alors l'élève-ordinateur recommence le programme en disant (à l'oral) **Avance de deux maisons**

Temps 3.2: Des colis pour les Steams – Expérimenter un programme en jouant différents rôles

Modalités de travail: en groupes (3-4 élèves)

Durée: 25 minutes

On installe 3 parcours de 9 tapis de la même couleur (voir illustration ci-dessous).

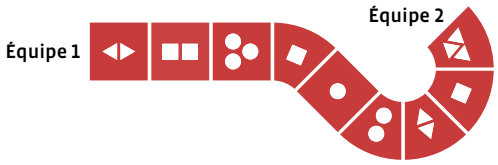
On répartit la classe en 6 groupes au maximum de trois élèves. Si le nombre d'élèves dans la classe n'est pas un multiple de 3, il est possible qu'une équipe soit composée de 2 élèves (avec 1 élève-ordinateur et 1 élève-robot Blob), ou de 4 élèves (avec 2 élèves-contrôleurs). Le rôle des élèves dans chaque groupe change à chaque fois qu'un Blob rentre à la station.

On place une équipe à chaque extrémité d'un parcours (voir illustration ci-dessous, Équipe 1 et Équipe 2): il y a donc 2 équipes sur chaque parcours.

On distribue à chaque groupe d'élèves les 3 programmes livraison découpés (voir fiche 3.2) et 12 petits objets.

On répartit les rôles: 1 élève-robot Blob, 1 élève-contrôleur, 1 élève-ordinateur.

L'élève qui jouera le rôle du robot- Blob se placera avant le 1er tapis du parcours avec deux objets dans les mains. L'élève qui jouera le rôle de l'ordinateur lui tournera le dos et tiendra dans ses mains les programmes. L'élève qui interprétera le rôle du contrôleur se positionnera de manière à pouvoir voir le programme lu par l'ordinateur et vérifier que le robot exécute bien les différentes instructions.



Mise en route

Avant de se rendre sur les maisons, on peut relire le Message C du temps 3.1 aux élèves et redonner ces éléments de mémorisation:

Consigne: Quelle est notre mission aujourd'hui?

Livrer des colis aux Steams

Consigne: Peut-on se rendre sur Square CT?

Non, car c'est trop pollué, il faut envoyer les robots Blobs

Consigne: Comment vont se déplacer ces robots?

Ils vont exécuter les instructions (les programmes) qu'on leur transmet.

On lance le **jeu** en rappelant bien qu'à chaque fois que 2 colis ont été déposés par l'élève-robot, elle ou il retourne à la station et les élèves du groupe changent de rôle.

Quand tous les élèves de chaque groupe ont joué chacun des rôles - ordinateur, contrôleur et robot (Blob) - (même si toutes les fenêtres n'ont pas de colis), on passe à la mise en commun.

Dans le programme 2, on remarque que l'instruction **Reculer d'une maison** est précédée du terme **sinon** et est légèrement décalée vers la droite (elle est indentée, voir encadré du temps 2.1). Dans ce programme 2, cette instruction **Reculer d'une maison** est exécutée uniquement dans le cas où la condition **Si maison avec fenêtre vide** n'est pas vérifiée : les élèves-robots doivent donc reculer uniquement dans le cas où la maison sur laquelle elles et ils se trouvent n'a aucune fenêtre vide.

Dans ce programme 2, l'élève-robot va procéder ainsi :

a) Avancer de deux maisons
b) Livrer un colis si la maison sur laquelle elle ou il est à au moins une fenêtre vide ou reculer d'une maison si cette maison n'a pas de fenêtre vide
c) Vérifier combien de colis elle ou il a livré en tout : si elle ou il en a livré 2, l'élève-robot retourne à la station, par contre, si elle ou il n'en a pas encore livré 2, alors l'élève-robot recommence le programme.

Par contre, dans le programme 3, on retrouve la même instruction **Reculer d'une maison** mais qui cette fois n'est pas indentée et n'est pas précédée par le terme **sinon**. Les élèves-robots doivent donc reculer d'une maison à chaque fois que la maison sur laquelle elles et ils se trouvent a une fenêtre vide ou pas.

Dans ce programme 3, l'élève-robot va ainsi :

a) Avancer de deux maisons
b) Livrer un colis uniquement si la maison sur laquelle elle ou il est a au moins une fenêtre vide
c) Reculer d'une maison
d) Vérifier combien de colis elle ou il a livrés en tout : si elle ou il en a livré 2, l'élève-robot retourne à la station, par contre, si elle ou il n'en a pas encore livré 2, alors l'élève-robot recommence le programme.

Mise en commun

Consigne : Que dois-je faire quand je joue le Blob ?

Attendre et exécuter les instructions sans prendre d'initiative

Consigne : Que dois-je faire quand je suis ordinateur ?

Attendre que le Blob exécute une instruction avant de lui en donner une nouvelle (se synchroniser avec ses actions), bien lire toutes les instructions, répéter ce processus à chaque étape.

Consigne : Que dois-je faire quand je suis contrôleuse ou contrôleur ?

Vérifier que chacun (ordinateur et Blob) attende l'autre, qu'il n'y ait pas d'erreur dans la transmission ni dans l'exécution des instructions, que le robot ne prenne pas d'initiative.

Consigne : Quel est le rôle le plus attrayant ? Le plus fastidieux ? Le plus long ?…
Quelles sont les conditions que vous avez testées/vérifiées ?

Si la maison avait une fenêtre vide et si on avait déposé 2 colis en tout.

Consigne : Est-ce qu'il y a eu des situations qui n'étaient pas prévues par le programme ? Quelles décisions avez-vous prises ? Quelles instructions aurait-il fallu ajouter à nos programmes pour que les robots puissent gérer ces situations ?

Il se peut par exemple que 2 Blobs parviennent sur la même case en même temps. Est-ce que c'est celui qui est arrivé sur cette maison en premier qui dépose son colis ? Ou celui avec le plus grand nombre de colis restant ? Nous aurions par exemple pu rajouter l'instruction **Si un robot est déjà sur cette maison, avancer d'une case**.

Trace écrite possible :

Les machines ne font qu'exécuter des instructions. Elles n'ont pas d'émotions et peuvent répéter indéfiniment, sans se lasser, les mêmes choses. C'est pourquoi les tâches automatisées sont souvent déléguées aux machines.

Séance 4
Logigrammes

Résumé :
- Rédiger et exécuter des logigrammes.

Matériel :
- kit Square CT (maisons)
- fiche 4.1 et/ou 4.2 (1 par élève)
- fiche 4.3 (1 par élève)

Temps 4.1 : Traduire un programme textuel en logigramme

Modalités de travail : en collectif

Durée : 30 minutes

Il s'agira de poursuivre la découverte de l'organisation des logigrammes. Les élèves vont tout d'abord devoir schématiser le programme 1 livraison de la séance précédente sous forme de logigramme (ce programme est inscrit sur les fiches que l'on va distribuer aux élèves).

Au préalable, on crée dans la salle un petit parcours de maisons (quelques tapis les uns à la suite des autres, similaire aux parcours de la séance 3) permettant de tester les logigrammes.

Mise en route

Consigne : Est-ce que vous vous souvenez des symboles et formes géométriques que nous pouvons utiliser pour créer un logigramme ?

Des ovales (pour le début et la fin du logigramme), des rectangles (pour les instructions), des losanges (pour les instructions conditionnelles, avec éventuellement des signes de validation et des croix pour indiquer le chemin à suivre si la condition est vérifiée ou si elle ne l'est pas), des flèches (pour indiquer l'ordre des instructions).

On distribue alors la fiche 4.1 pour guider les élèves qui le demandent ou pour structurer la correction. Pour une situation de recherche, distribuer aux élèves la fiche 4.2 qui nécessite la création de toute la structure du logigramme.

Moment 1

Les élèves complètent le logigramme (voir fiche 4.1) ou le créent complètement (voir fiche 4.2).

Moment 2

On procède à une mise en commun des schémas et on teste en classe avec un parcours de maisons les propositions pour valider les logigrammes. Insister sur les différents éléments structurant le logigramme.

On compare les expériences de lecture entre le logigramme et le programme textuel.

Temps 4.2: Lire un logigramme complexe

Modalités de travail: en individuel ou en binômes

Durée: 15 minutes

On distribue un exemplaire de la fiche 4.3 à chaque élève.

Consigne: Les Steams trouvent que leurs visiteurs humains ont vraiment des prénoms étranges... Ils nous proposent donc un traducteur de prénom Steam sous forme de logigramme qu'il vous suffit de suivre pour traduire votre prénom en Steam. . Traduisez donc votre prénom et inscrivez-le dans le cadre prévu à cet effet!

Séance 5
Chasse au trésor (facultatif)

Résumé:
- Expérimenter un programme représenté en format texte et sous forme de logigramme qui intègre des instructions conditionnelles imbriquées.

Matériel:
- kit Square CT (maisons et Steams (= formes))
- fiches 5 à 7 (1 par groupe + enseignant·e)

Temps 5.1: Lire et exécuter un programme complexe

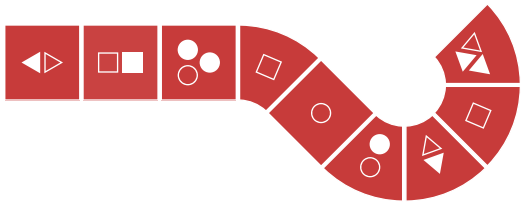
Modalités de travail: en groupes (3-4 élèves)

Durée: 25 minutes

Mise en route

Au préalable, on crée les mêmes 3 parcours de 9 tapis de la même couleur que pour la séance 3.

On dépose également un Steam dans UNE des fenêtres DE CHAQUE tapis (les Steams sont représentés par les petites formes triangulaires, carrées ou rondes provenant de la découpe des fenêtres des maisons). Voir l'illustration ci-dessous:



Sous certains Steams, on cache un cadeau: pièce de puzzle ou un message codé (voir fiche 7) par exemple.

Remarque: Les messages codés de la fiche 7 ont été chiffrés avec le Code César à partir des propositions de messages du scénario 1:

Clé 10

KMBYCC DRO EXSFOBCO **Across the universe**	**LSOXFOXEO** **Bienvenue**	**XYEC FOXYXC OX ZKSH** **Nous venons en paix**
CDKB DBOU **Star Trek**	**ROVVY GYBVN** **Hello world**	**VO ZBYQBKWWO KZYVVY** **Le programme Apollo**

Clé 21

CPWWGZ **Hubble**	**GV APNZZ VMDVIZ** **La fusée Ariane**	**GZ NVWMZ GVNZM YZ GPFZ** **Le sabre laser de Luke**
WPUU G ZXGVDM **Buzz l'éclair**	**GVDFV** **Laïka**	**WGJWN** **Blobs**

Les élèves peuvent découvrir cette méthode de chiffrement en réalisant l'enquête 5 *Comment l'empereur romain Jules César protégeait-il ses messages secrets?*

On lit le Message D:

Message D

Pour vous remercier de vos colis, à vous de trouver nos cadeaux! Nous sommes farceurs et les avons cachés dans nos maisons! Saurez-vous tous les retrouver? Suivez le programme!

Moment 1

On montre le programme A de la fiche 5 aux équipes (plier la fiche pour l'instant - ou la découper - de manière à ne pas montrer le logigramme A tout de suite).

Consigne: Que remarquez-vous dans ce nouveau programme?

Il est plus long mais surtout plus complexe à lire que les programmes des autres séances: il y beaucoup de phrases décalées vers la droite, certaines le sont plus que d'autres.

On peut introduire le terme **imbriqué**.

Consigne: Il y a des instructions conditionnelles à l'intérieur d'autres instructions conditionnelles, elles sont imbriquées.

On montre maintenant le logigramme A de la fiche 5 (en dépliant la fiche si cette dernière avait été pliée) côte à côte avec le programme A.

Consigne: S'agit-il du même algorithme?

Oui, le programme A et le logigramme A représentent bien le même algorithme, sous deux formes différentes: l'une est textuelle, l'autre visuelle.

Consigne: Quelle est la version la plus facile à lire et à suivre?

Le logigramme est sans doute la représentation la plus facile à lire et à suivre.

Les élèves sont répartis en groupes, toujours avec les mêmes rôles que pour la séance 3 du scénario (élève-ordinateur, élève-contrôleur et élève-robot Blob), chaque équipe ayant une fiche 5. Ils essaient d'exécuter le programme en s'aidant **soit de la version textuelle (programme A) ou visuelle (logigramme A).**

Consigne: Quelle représentation avez-vous utilisée? Plutôt la représentation textuelle ou le logigramme? Pourquoi? Est-ce que l'on doit vérifier s'il y a un cadeau sous le Steam sur chaque tapis?

Non, on ne vérifie s'il y a un cadeau sous le Steam uniquement **Si** le tapis a un Steam.

Consigne: Est-ce que le robot vous a semblé faire des tests inutiles? Comment pourrait-on rendre notre programme plus efficient, rapide à exécuter?

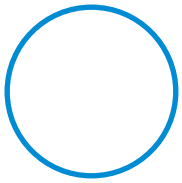
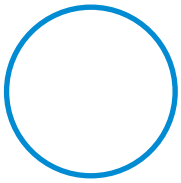
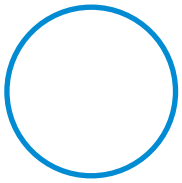
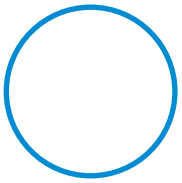
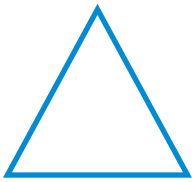
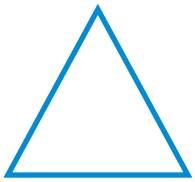
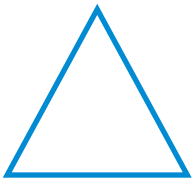
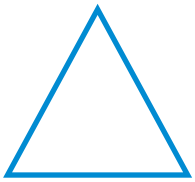
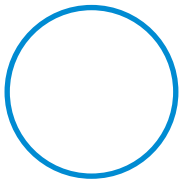
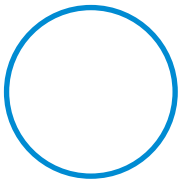
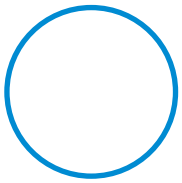
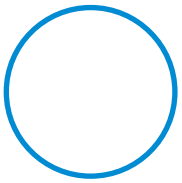
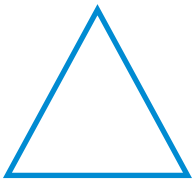
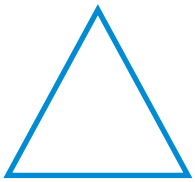
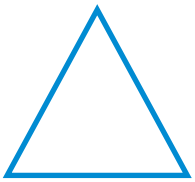
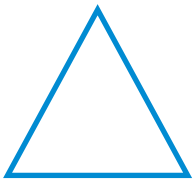
On ne devrait tester la condition **Si récupéré 2 cadeaux en tout** uniquement lorsque l'on trouve un nouveau cadeau. Ça ne sert à rien de tester à chaque fois si on a bien récupéré 2 cadeaux si le tapis sur lequel on se trouve n'a pas de Steams.

On montre alors la fiche 6, avec le programme B et le logigramme B simultanément affichés côte à côte.

Dans ce programme B, on vérifie si 2 cadeaux uniquement ont été rapportés en tout quand on trouve un nouveau cadeau. Cette condition a été imbriquée dans une autre condition (Si cadeau sous le Steam), elle-même imbriquée dans une autre condition (Si maison avec Steam). Quelques instructions supplémentaires ont dû être rajoutées, notamment pour indiquer qu'il faut recommencer le programme. Ce programme est plus long que le programme A, il est plus compliqué à lire pour un humain. Par contre, même s'il y a plus de lignes, plus d'instructions écrites, il sera plus rapide à exécuter par un ordinateur ou un robot, car certaines de ces instructions ne sont exécutées que dans certains cas.

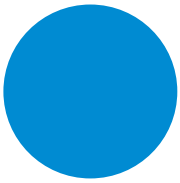
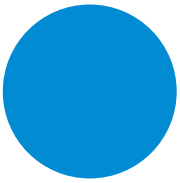
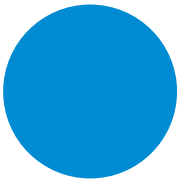
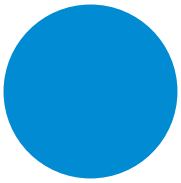
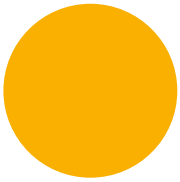
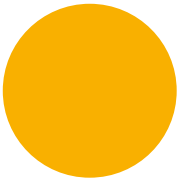
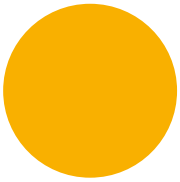
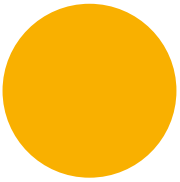
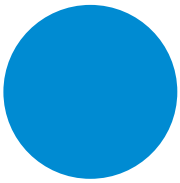
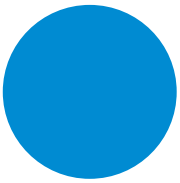
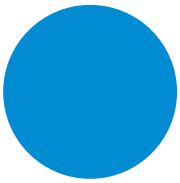
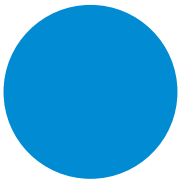
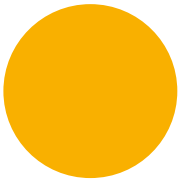
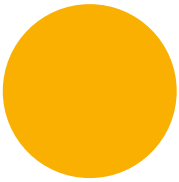
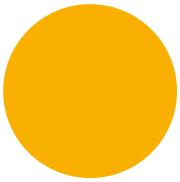
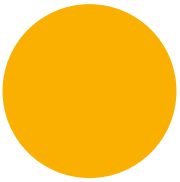
Fiche 1.1

Cartes critères (formes)

 Rond	 Rond	 Rond	 Rond
 Carré	 Carré	 Carré	 Carré
 Triangle	 Triangle	 Triangle	 Triangle
 Rond	 Rond	 Rond	 Rond
 Carré	 Carré	 Carré	 Carré
 Triangle	 Triangle	 Triangle	 Triangle

Fiche 1.2

Cartes critères (couleur)

 Bleu	 Bleu	 Bleu	 Bleu
 Rouge	 Rouge	 Rouge	 Rouge
 Jaune	 Jaune	 Jaune	 Jaune
 Bleu	 Bleu	 Bleu	 Bleu
 Rouge	 Rouge	 Rouge	 Rouge
 Jaune	 Jaune	 Jaune	 Jaune

Fiche 1.3

Cartes critères (nombres)

1	1	1	1
2	2	2	2
3	3	3	3
1	1	1	1
2	2	2	2
3	3	3	3

Fiche 1.4

Cartes connecteurs

ET	ET	ET	ET
ET	ET	ET	ET
ET	ET	ET	ET
OU	OU	OU	OU
OU	OU	OU	OU
OU	OU	OU	OU

Fiche 1.5

Messages

Message A

Chers élèves, voilà bien longtemps que vous n'êtes pas venus nous rendre visite!

Nous nous sommes rencontrés grâce à vos Blobs la dernière fois. Saurez-vous reconnaître nos maisons?

Message B

Chers élèves, vous semblez bien vous repérer dans notre Square CT! Toutes nos félicitations! Nous aimerions mieux vous connaître...

On peut commencer par se saluer! Chez nous c'est difficile, chaque maison a sa propre manière de dire bonjour.

Par exemple, les habitants des maisons bleues et 2 saluent en levant les 2 bras... Saurez-vous à votre tour nous saluer?

Fiche 1.6

Cartes saluts

SI  ET 2 **ALORS** lever 2 bras

SI  OU 3 **ALORS** se mettre à genoux

SI  ET 1 **ALORS** lever 1 jambe

SI  OU  **ALORS** lever 1 bras

Fiche 1.7

Prénom : .....

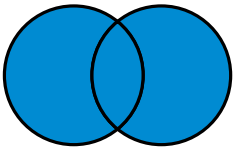
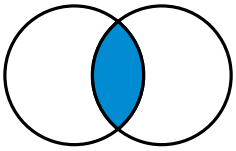
Instructions et connecteurs

Une → _____ permet d'exécuter certaines actions uniquement dans certains cas.

Elle se compose d'une ou plusieurs → _____ qui sont soit vraies, soit fausses.

Pour le vérifier, on effectue un → _____.

On peut également utiliser des → _____ comme **ET**, **OU**.

	
2 critères avec → _____	2 critères avec → _____

Avec le connecteur logique **OU**, l'ensemble des possibles est plus → _____ qu'avec le connecteur **ET**.

Fiche 1.7

Corrigé

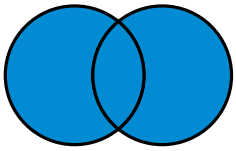
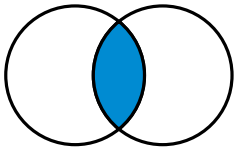
Instructions et connecteurs

Une → *instruction conditionnelle* permet d'exécuter certaines actions uniquement dans certains cas.

Elle se compose d'une ou plusieurs → *conditions* qui sont soit vraies, soit fausses.

Pour le vérifier, on effectue un → *test*.

On peut également utiliser des → *connecteurs logiques* comme **ET**, **OU**.

	
2 critères avec → *OU*	2 critères avec → *ET*

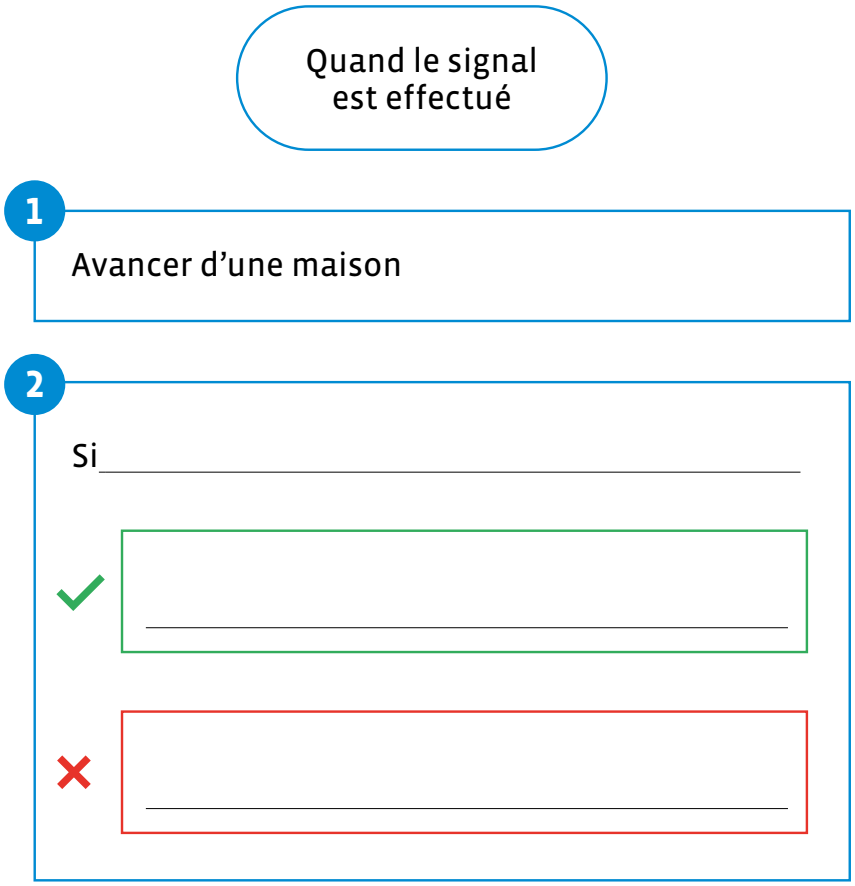
Avec le connecteur logique **OU**, l'ensemble des possibles est plus → *grand* qu'avec le connecteur **ET**.

Fiche 2.1

Prénom : .....

Exercice 1

Écris ton programme sous forme de blocs.



Exercice 2

Complète le texte.

Le logigramme est un outil utilisé par les informaticiennes et les informaticiens pour présenter leurs → _____ .

Les instructions sont écrites dans des → _____ , les conditions dans des → _____ . En réalisant des tests, on peut savoir si la condition est → _____ ou → _____.

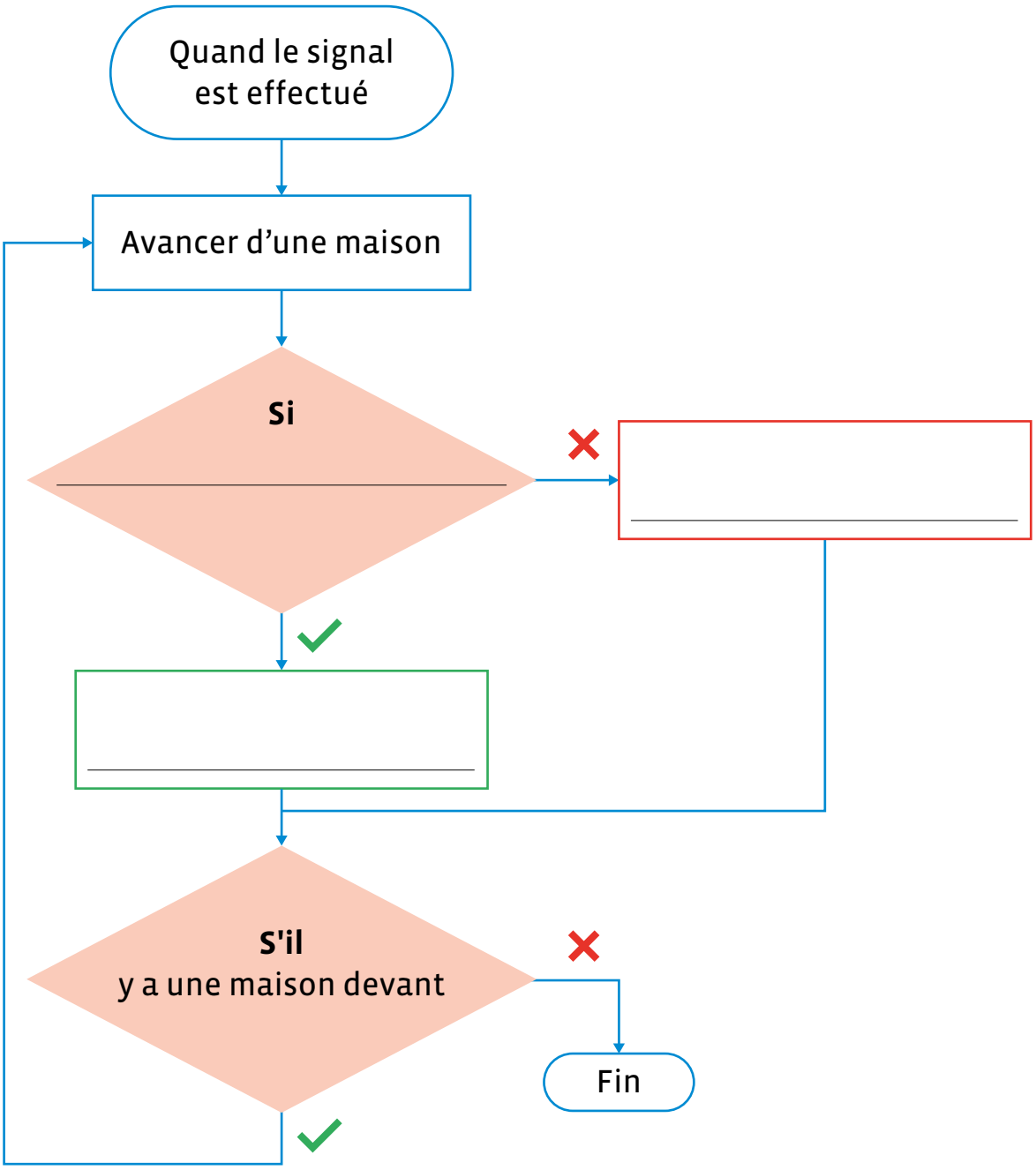
Les → _____ permettent de visualiser l'ordre, le sens d'exécution des instructions de l'algorithme.

Fiche 2.2

Prénom : .....

Exercice 3

Complète le logigramme suivant pour représenter ce même programme.



Fiche 3.1

Message

Message C

Chers élèves,

Pas de spectacle aujourd'hui, car nous subissons actuellement un pic de pollution. Pouvez-vous nous aider en nous livrant des vivres?

Nous vous conseillons d'utiliser vos Blobs machines car l'air est irrespirable!

Nous avons créé des programmes qu'elles devront exécuter...

Fiche 3.2

Cartes programmes livraison

1 Avancer d'une maison

Si maison avec fenêtre vide
alors: livrer un colis

Si 2 colis ont été déposés en tout
alors: retourner à la station (fin du programme)
sinon: recommencer le programme

2 Avancer de 2 maisons

Si maison avec fenêtre vide
alors: livrer un colis
sinon: reculer d'une maison

Si 2 colis ont été déposés en tout
alors: retourner à la station (fin du programme)
sinon: recommencer le programme

3 Avancer de 2 maisons

Si maison avec fenêtre vide
alors: livrer un colis

Reculer d'une maison

Si 2 colis ont été déposés en tout
alors: retourner à la station (fin du programme)
sinon: recommencer le programme

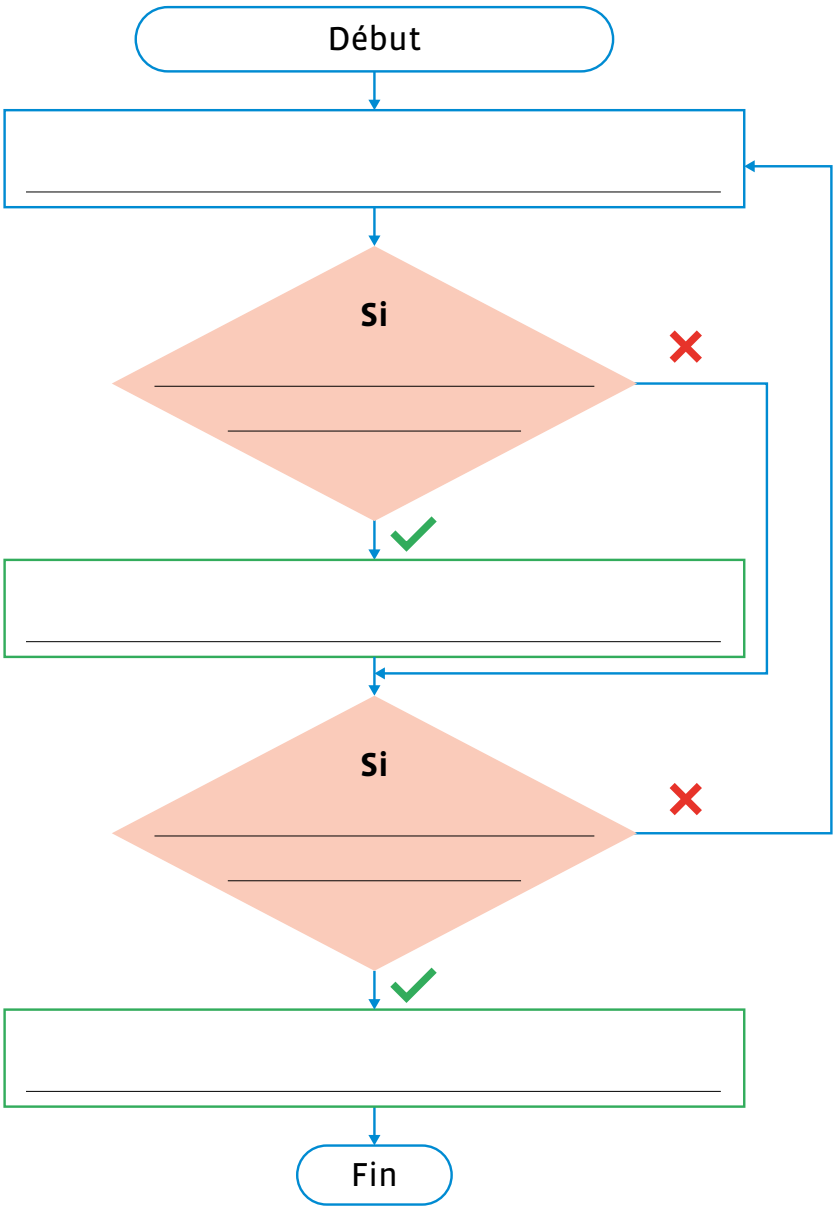
Fiche 4.1

Prénom : .....

Le programme des colis (avec structure)

Voici le programme des Steams pour déposer leurs colis de vivres.
Complète le logigramme pour représenter ce programme.

Avancer d'une maison
Si maison avec fenêtre vide
    alors: livrer un colis
Si 2 colis ont été déposés en tout
    alors: retourner à la station (fin du programme)
    sinon: recommencer le programme



Fiche 4.2

Prénom : .....

Le programme des colis (recherche)

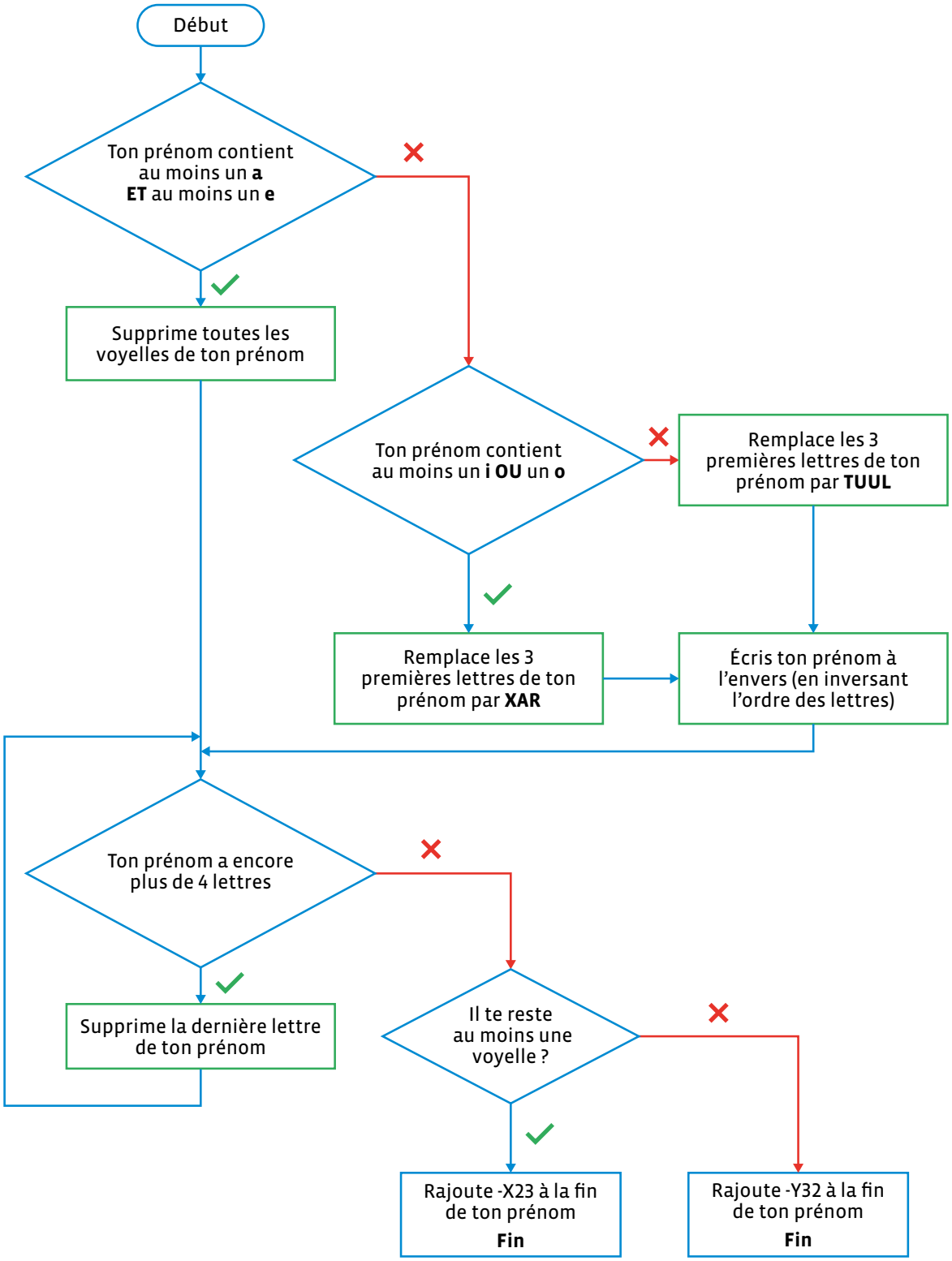
Voici le programme des Steams pour déposer leurs colis de vivres.
Crée le logigramme pour représenter ce programme.

Avancer d'une maison
Si maison avec fenêtre vide
    alors: livrer un colis
Si 2 colis ont été déposés en tout
    alors: retourner à la station (fin du programme)
    sinon: recommencer le programme

Fiche 4.3

Prénom : .....

Traducteur de prénom



Ton prénom **Steam** : _____

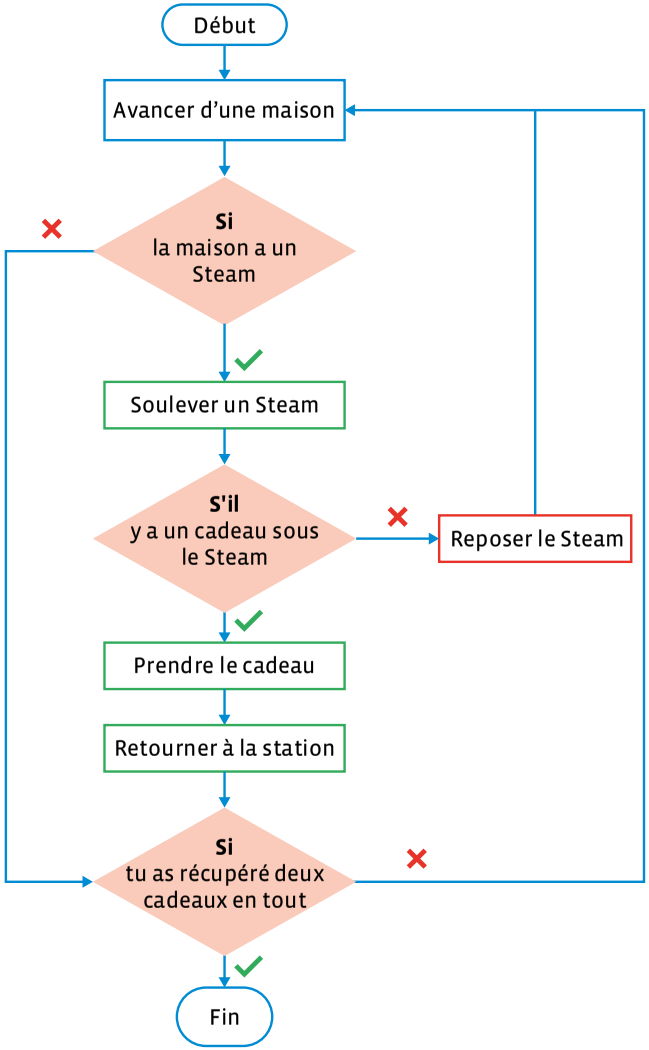
Chasse au trésor : programme A

Prénom : .....

Programme A

Avancer d'une maison
Si maison avec Steam(s)
  alors :
  Soulever un Steam
  Si cadeau sous le Steam
    alors :
    Prendre le trésor
    Retourner à la station
    sinon :
    Reposer le Steam
    Recommencer le programme
Si récupéré 2 cadeaux en tout
    alors :
    Fin du programme
    sinon :
    Recommencer le programme

Logigramme A



Fiche 6

Prénom : .....

Chasse au trésor : programme B

Logigramme B

Programme B

Avancer d'une maison
Si maison avec Steam(s)
  alors :
  Soulever un Steam
  Si cadeau sous le Steam(s)
    alors :
    Prendre le cadeau
    Retourner à la station
  Si récupéré 2 cadeaux en tout
  alors :
  Fin du programme
  sinon :
  Recommencer le programme
sinon :
Reposer le Steam
Recommencer le programme
sinon :
Recommencer le programme

Fiche 7

Messages chiffrés des Steams

Clé 10

KMBYCC DRO EXSFOBCO Clé 10	LSOXFOXEO Clé 10	XYEC FOXYXC OX ZKSH Clé 10
CDKB DBOU Clé 10	ROVVY GYBVN Clé 10	VO ZBYQBKWWO KZYVVY Clé 10

Clé 21

CPWWGZ Clé 21	GV APNZZ VMDVIZ Clé 21	GZ NVWMZ GVNZM YZ GPFZ Clé 21
WPUU G ZXGVDM Clé 21	GVDFV Clé 21	WGJWN Clé 21

Fiche 8

Cartes maison