FONDATION La main à la pâte

Centre pilote 54
La main à la pâte

Parcours

1, 2, 3…CODEZ !

Cycle III

Le projet « **1, 2, 3… codez !** » développé par la Fondation *La main à la pâte* avec l'appui de la communauté scientifique (notamment l'Inria) vise à initier élèves et enseignants à la science informatique, de la maternelle à la classe de 3e.
Il propose à la fois des activités branchées (nécessitant un ordinateur, une tablette ou un robot) permettant d'introduire les bases de la programmation et des activités débranchées (informatique sans ordinateur) permettant d'aborder des concepts de base de la science informatique (algorithme, langage, représentation de l'information…).

Sciences et technologie

Par l'analyse et par la conception, les élèves peuvent décrire les interactions entre les objets techniques et leur environnement, et les processus mis en œuvre. Les élèves peuvent aussi réaliser des maquettes, des prototypes, comprendre l'évolution technologique des objets et utiliser les outils numériques.

Matière, mouvement, énergie, information

- Identifier un signal et une information
- Identifier différentes formes de signaux (sonores, lumineux, radio…).
- Nature d'un signal, nature d'une information, dans une application simple de la vie courante

Matériaux et objets techniques

- Repérer et comprendre la communication et la gestion de l'information
- Environnement numérique de travail.
- Le stockage des données, notions d'algorithmes, les objets programmables.
- Usage de logiciels usuels.

Les élèves apprennent à connaitre l'organisation d'un environnement numérique. Ils décrivent un système technique par ses composants et leurs relations. Les élèves découvrent l'algorithme en utilisant des logiciels d'applications visuelles et ludiques. Ils exploitent les moyens informatiques en pratiquant le travail collaboratif. Les élèves maitrisent le fonctionnement de logiciels usuels et s'approprient leur fonctionnement.

Mathématiques

Nombres et calculs

- Organisation et gestion de données
- Représentations usuelles : tableaux (en deux ou plusieurs colonnes, à double entrée), graphiques cartésiens.

Espace et géométrie

- (Se) repérer et (se) déplacer dans l'espace en utilisant ou en élaborant des représentations
- Se repérer, décrire ou exécuter des déplacements, sur un plan ou sur une carte.
- Accomplir, décrire, coder des déplacements dans des espaces familiers.
- Programmer les déplacements d'un robot ou ceux d'un personnage sur un écran.
- Vocabulaire permettant de définir des positions et des déplacements.
- Divers modes de représentation de l'espace.
- Situations donnant lieu à des repérages dans l'espace ou à la description, au codage ou au décodage de déplacements.
- Travailler : avec de nouvelles ressources comme les […] logiciels d'initiation à la programmation…

Repères de progressivité

Une initiation à la programmation est faite à l'occasion notamment d'activités de repérage ou de déplacement (programmer les déplacements d'un robot ou ceux d'un personnage sur un écran)

Partenaires : Les Fab Labs lorrains

Ouverts aux étudiants, scientifiques, entrepreneurs ou bricoleurs, les Fab Labs (Fabrication Laboratories) offrent des moyens de prototypage rapide pour la réalisation de projets innovants. Panorama de ces ateliers de fabrication numérique originaux en Lorraine.

SOMMAIRE

Séance 1 en classe Le parcours du combattant	
Objectifs	Se familiariser avec les notions d'algorithme, de langage, et de représentation de l'information (texte et image)
Notions	« Algorithme » - Un "algorithme" est une méthode permettant de résoudre un problème - Un test dit quelle action effectuer quand une condition est vérifiée - Une condition est une expression qui est soit vraie, soit fausse
Matériel	Par élève - Fiche 12 - Fiche 13 Pour la classe - Un vidéoprojecteur ou une impression A3 de la Fiche 12 Cahiers d'expériences.
Phases de déroulement de l'activité	**Situation déclenchante** Le héros s'éveille dans un monde inconnu, en pleine nature. Un périple s'offre à lui pour descendre de la montagne où il se trouve. Le héros ne se souvient pas d'avoir gravi la montagne, et la forêt au pied de la montagne ne lui est pas familière. Il ne reconnaît pas le chant des oiseaux… il n'est pas chez lui. En contre-bas, il lui semble apercevoir une clairière : il décide de la rejoindre. Les élèves doivent le guider en lui donnant des instructions conditionnelles. **Exprimer des instructions à l'aide de conditions (par groupe ou collectivement)** L'enseignant distribue aux élèves la Fiche 12, et il la projette ensuite au tableau : il s'agit du parcours que va devoir effectuer le héros pour rejoindre la clairière au pied de la montagne. Pour l'aider, les élèves doivent décrire une succession d'instructions que le héros suivra à la lettre pour arriver sain et sauf. La formulation de ces instructions doit prendre la forme SI …. ALORS…. **Mise en commun** La mise en commun est l'occasion pour l'enseignant d'introduire un vocabulaire nouveau, utilisé en informatique. Une méthode permettant de résoudre un problème s'appelle un « algorithme ». Dans le cas présent, l'algorithme s'exprime en utilisant des « tests » : une « condition » (« *SI le héros rencontre une falaise* ») suivie d'une ou plusieurs instructions à suivre si la condition est vérifiée (« *ALORS il doit escalader* »). À chaque étape de son périple, le héros vérifie la totalité des conditions du programme, et obéit scrupuleusement à toutes les instructions applicables.

	L'enseignant demande aux élèves de comparer cet algorithme avec une autre instruction que l'on aurait pu donner au héros : « *retourne chez toi* ». Dans le second cas, on donne un problème complexe à résoudre, sans expliquer comment le faire. Si le héros ne sait pas comment le faire, notre instruction ne va pas l'aider. Un algorithme est construit à partir d'instructions « élémentaires » que le héros sait exécuter. **Exercice : inventer soi-même d'autres instructions conditionnelles** L'enseignant propose aux élèves d'inventer d'autres instructions, en suivant la même règle (il faut être le plus explicite possible) En imaginant par exemple le héros dans un autre environnement : jungle hostile, banquise, cité futuriste, etc… Il peut aussi inciter les élèves à expliciter, à l'aide d'expressions conditionnelles, des algorithmes qu'ils rencontrent au quotidien, par exemple en sport, en grammaire, ou même dans le règlement intérieur de l'école (ce qu'il faut faire dans telle ou telle situation), etc. **Conclusion et traces écrites** La classe synthétise collectivement ce qui a été appris au cours de cette séance : • *Un algorithme est une méthode permettant de résoudre un problème.* • *Un test dit quelle action effectuer quand une condition est vérifiée* • *Une condition est une expression qui est soit vraie, soit fausse* Les élèves notent ces conclusions dans leur cahier de sciences.
Remarques	**Prolongement** Dans la salle de motricité, l'enseignant peut reproduire un autre parcours du combattant, en utilisant des obstacles, des tunnels, des cerceaux, des marches, etc… Le but de l'exercice est de définir des instructions, en utilisant là encore la même syntaxe « SI… ALORS… », pour que les élèves puissent parcourir le chemin en toute sécurité.

Retour **SOMMAIRE**

Séance 2 en classe Décoder un message	
Objectifs	Se familiariser avec les notions d'algorithme, de langage, et de représentation de l'information (texte et image)
Notions	« Information » - On peut coder un texte en représentant ses lettres par des nombres choisis à l'avance.
Matériel	Par élève - Fiche 14 Pour la classe - Un vidéoprojecteur ou une impression A3 de la Fiche 14 Cahiers d'expériences.
Phases de déroulement de l'activité	**Situation déclenchante** À peine sorti d'un périlleux parcours, le héros doit résoudre une énigme gravée sur un tronc d'arbre. Les élèves comprennent qu'il s'agit d'un message codé. Pour aider le héros, ils doivent décoder ce message pour en comprendre le sens. **Expérimentation : décoder un message encodé (par groupes)** L'enseignant projette au tableau la première moitié de la Fiche 14 : il s'agit de l'inscription gravée sur le tronc d'arbre. Il demande aux élèves ce qu'ils en pensent. Les enfants ne peuvent pas lire ce qui est écrit, pourtant cela ressemble à un texte inscrit dans un langage inconnu. Chaque symbole ressemble beaucoup à des nombres écrits en chiffres arabes, ce qui permettra de nommer plus facilement chacun d'eux. Il suffit peut-être, pour comprendre le message, de trouver une correspondance entre ces symboles et les lettres de notre alphabet ? L'enseignant introduit alors les termes « encoder » et « décoder ».  L'enseignant affiche alors la totalité de la fiche documentaire au tableau. Les élèves doivent trouver des indices permettant de décoder ce message Au tableau, l'enseignant complète au fur et à mesure la table de correspondance en fonction des trouvailles des différents groupes. Si les élèves ont du mal à comprendre comment décoder ce message, l'enseignant peut les aiguiller progressivement de plusieurs façons :

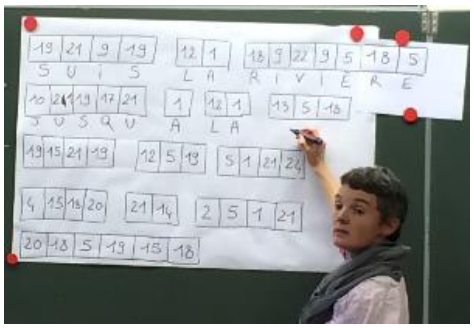
Quels sont les mots les plus courts ? À quoi peuvent-ils correspondre en français ?
Les mots les plus courts de la langue française sont « à », « y », mais on peut également retrouver des formes contractées « l' », « d' », etc. Les mots de 2 lettres sont également peu nombreux (le, la, on …)
Quelle est la lettre la plus courante dans un texte rédigé en français ? (réponse : la lettre E)
Qu'en est-il ici ? Dans le texte codé ici, c'est le symbole 5. On peut donc supposer que « 5 » encode systématiquement toutes les lettres « E » du message initial. Et la lettre E est justement la cinquième lettre de l'alphabet.

De façon plus intuitive, on peut aussi essayer, puisque les symboles ressemblent à des nombres, de les remplacer par les lettres de l'alphabet du même rang (intuitivement, on a « envie » de remplacer 1 par A, 2 par B, 3 par C…)

Mise en commun

Ensemble, la classe réussit à décoder le message :
SUIS LA RIVIERE
JUSQU'A LA MER
SOUS LES EAUX DORT
UN BEAU TRESOR
Dans ce codage, la lettre A était codée par « 1 », la lettre B, par « 2 », la lettre E, par « 5 » et ainsi de suite jusqu'à Z codé par « 26 ».

Les élèves sont alors encouragés à encoder/décoder d'autres messages de leur choix, pour se les transmettre (attention, pour cette phase également, penser à placer les symboles dans des cases).



Classe de CE2 d'Emmanuelle Wilgenbus (Antony)

Conclusion et traces écrites
La classe synthétise collectivement ce qui a été appris au cours de cette séance :
On peut coder un texte en représentant ses lettres par des nombres choisis à l'avance (par exemple, 1 peut coder « A », 2 peut coder « B »…)
Les élèves notent ces conclusions dans leur cahier de sciences.

Remarques	Pour simplifier le décodage et nous focaliser sur la méthode plus que sur le résultat, nous n'encodons pas (à dessein) la ponctuation. En français, les termes « coder, chiffrer, crypter » sont souvent confondus. Ici, Nous parlons ici de codage car nous nous intéressons à la représentation des caractères alphabétiques par des nombres, qui est utilisée en informatique même lorsque des informations ne sont pas confidentielles. Alors que le « chiffrement » désigne la déformation d'un message pour le rendre incompréhensible aux personnes non concernées. Dans un codage « classique », il est obligatoire d'encoder « A » en « 01 », car tous les symboles du codage doivent avoir la même longueur. Cependant, à cet âge, les enfants apprennent la numération et on leur explique qu'un nombre ne commence jamais par un zéro à gauche (sauf si le nombre est zéro, bien entendu). Voilà pourquoi l'encodage ici repose sur des cases qui permettent de délimiter efficacement les symboles.

Retour **SOMMAIRE**

Séance 3 en classe Programmer un parcours	
Objectifs	Se familiariser avec les notions d'algorithme, de langage, et de représentation de l'information (texte et image)
Notions	« Machines » - Les machines qui nous entourent ne font qu'exécuter des "ordres" (instructions) - En combinant plusieurs instructions simples on peut effectuer une tâche complexe « Langages » - On peut donner des instructions à une machine en utilisant un langage spécial, appelé langage de programmation, compréhensible par l'homme et la machine. - Un programme est un algorithme exprimé dans un langage de programmation. - Un bug est une erreur dans un programme. - Un tout petit bug peut parfois avoir des conséquences énormes.
Matériel	Par élève - Fiche 15 - Un pion (jouet, figurine) représentant le sous-marin Pour la classe - Un vidéoprojecteur ou une impression A3 de la Fiche 15 - Une silhouette aimantée (ou punaisée) représentant le sous-marin Cahiers d'expériences.
Phases de déroulement de l'activité	**Situation déclenchante** Après avoir suivi la rivière, le héros arrive à la mer. Sur la plage, il repère un ponton, et s'en approche. En regardant au fond de l'eau, il voit le trésor ! Mais celui-ci est hors de portée. En revanche, il voit un petit sous-marin qu'on peut commander à la voix. Il va falloir lui expliquer comment aller chercher le trésor. **Expérimentation :** inventer un langage pour guider le sous-marin (par binômes) L'enseignant projette au tableau la Fiche 15 : on y voit le fond de l'eau avec un dédale de coraux que le sous-marin doit parcourir afin de parvenir jusqu'au trésor. Par binômes, les élèves doivent proposer une série d'instructions qui pourront décrire le parcours à suivre. L'enseignant introduit alors le terme « programme » pour décrire l'ensemble d'instructions simples qui peuvent être exécutées par une machine. Les contraintes sont : le sous-marin ne peut se déplacer que d'une case à la fois ; il ne bouge pas en diagonale. Les binômes peuvent essayer de reproduire leur parcours en bougeant le pion qui leur est fourni, en le faisant respecter scrupuleusement les instructions.

	Mise en commun L'enseignant demande à un des groupes de présenter son programme à la classe. Pour vérifier ce que donne l'exécution du programme, les élèves déplacent au tableau la silhouette représentant le sous-marin, en suivant rigoureusement les instructions. Si la méthode est concluante, l'enseignant la reprend au tableau, et demande si d'autres binômes ont eu d'autres idées. On remarque qu'il existe (au moins) deux langages pour diriger le sous-marin. On peut lui donner des directions « absolues » (va vers la surface, va à l'Ouest/vers le ponton…) ou, au contraire, des directions relatives, c'est-à-dire qui dépendent de l'orientation du sous-marin (tourne vers la droite, avance, tourne vers la gauche, recule…). *Note : il est préférable de découper l'instruction « avance d'une case vers la droite » en 2 instructions bien distinctes : 1/ tourne vers la droite (sous-entendu : en restant sur place), puis 2/ avance d'une case.* La classe remarque que le sous-marin n'a besoin que d'un langage très simple pour être commandé (en particulier, très peu de mots différents sont nécessaires). L'enseignant explique que les machines comme les ordinateurs, les robots, etc., peuvent être programmés à l'aide de langages particuliers, appelés « langages de programmation », qui sont beaucoup plus simples que les langues naturelles comme le français, l'anglais, etc. Un second apport de cette mise en commun permet d'appréhender la notion de « bug ». Au cours des présentations des différents programmes, il viendra certainement une occasion où une instruction manquera, ou sera erronée. À ce moment, même si la classe sait que le résultat ne sera pas bon, l'enseignant peut décider d'amener tout de même le programme jusqu'à sa fin, pour voir où le sous-marin finira par arriver. Une seule erreur peut avoir des conséquences très importantes. On remarquera qu'une erreur dans un langage autocentré peut conduire plus loin du trésor qu'une erreur dans un langage allocentré. Cependant, dans les deux cas, il s'agit d'un bug et on notera deux choses. Premièrement, l'objectif n'est pas atteint, donc c'est un échec aussi important dans un cas que dans l'autre. Deuxièmement, si le pirate qui a laissé le trésor au fond de l'eau a également placé des pièges autour, alors on ne veut pas se tromper... même pas un tout petit peu. **Conclusion et traces écrites** La classe synthétise collectivement ce qui a été appris au cours de cette séance : - Un programme est une suite d'instructions exprimées dans un langage particulier compréhensible par l'homme et la machine. - Un bug est une erreur dans un programme. Un tout petit bug peut parfois avoir des conséquences énormes. Les élèves notent ces conclusions dans leur cahier de sciences.

Remarques	La première méthode (Nord, Ouest…) est dite « allocentrée » tandis que la seconde (droite, gauche…) est dite « autocentrée ». Les élèves n'ont pas besoin de connaître ces termes, qui ne seront plus utilisés par la suite. Le fait même de distinguer ces deux méthodes n'est pas l'objectif de cette séance : par ailleurs, les enfants mélangeront souvent des termes issus des deux méthodes. Une troisième méthode peut (plus rarement) être proposée : il s'agit de donner des coordonnées aux cases (A1, A2, B1…) et, comme dans un jeu de bataille navale, coder les déplacements en donnant le nom de la case de départ et d'arrivée. Exemple, « va de A1 vers A2 ». À noter : le chemin « A1 vers A2 » n'est pas ambigu car les cases sont adjacentes. En revanche, le chemin « A1 vers B7 » est ambigu (et, donc, non satisfaisant) : il y a plusieurs façons d'aller de la première case à la seconde. Nous ne détaillons pas cette méthode dans ce qui suit.

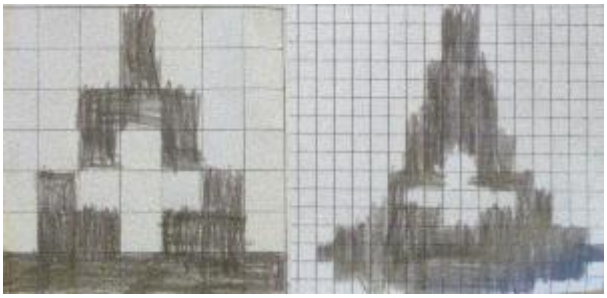
Retour **SOMMAIRE**

Séance 4 en classe Appeler le magicien	
Objectifs	Se familiariser avec les notions d'algorithme, de langage, et de représentation de l'information (texte et image)
Notions	« Information » - On peut représenter une image par une grille de pixels noirs ou blancs.
Matériel	Par élève - Fiches 16 et 17 (sur calque si possible) Pour la classe - Un vidéoprojecteur ou une impression A3 de la Fiche 158 Cahiers d'expériences.
Phases de déroulement de l'activité	**Situation déclenchante** Dans le coffre à trésor récupéré au fond de l'océan (séance précédente), le héros trouve un parchemin qui décrit une recette magique : cette recette permettra au héros de retourner chez lui ! Mais sans les ingrédients ou les ustensiles, le héros ne peut pas la réaliser. Le gardien lui parle alors d'un magicien qui pourrait lui venir en aide. Afin de le contacter, le héros doit envoyer un message aux oiseaux, qui sauront localiser le magicien. **Expérimentation** : pixelliser une image (par binômes) Parce que les oiseaux de ce pays ne savent pas parler ou comprendre la langue du héros, il n'y a qu'une solution pour s'adresser à eux : dessiner sur le sol quelque chose qui attirera leur regard. À sa disposition, le héros voit de gros galets : des blancs et des noirs. Il pourrait les utiliser pour réaliser une fresque au sol, que les oiseaux pourront voir d'en haut. L'enseignant distribue alors la Fiche 16 et la moitié supérieure de la Fiche 17 (grilles de 7x7). À l'aide des galets blancs ou des galets noirs, alignés en une grille de 7x7 cases (appelés « pixels »), les élèves doivent reproduire grossièrement la forme du chapeau du magicien. Chaque case ne peut être qu'entièrement noire, ou entièrement blanche, ce qui correspond à l'utilisation d'un galet noir ou d'un galet blanc. Dans tous les cas, il faudra rappeler sans cesse la consigne : il n'est pas question de simplement décalquer l'image du chapeau : les cases ne peuvent être que totalement noires ou totalement blanches, et il est interdit de subdiviser les cases en y traçant des traits supplémentaires pour mieux coller au dessin original. Un exercice en ligne propose une démonstration numérique de cette consigne.

Mise en commun
Les élèves comparent les images pixellisées qu'ils ont réalisées. Si les images sont difficilement reconnaissables de près, brandies d'un bout à l'autre de la classe, le chapeau de magicien y devient tout à fait identifiable.

Expérimentation : améliorer l'image pixellisée
Les élèves les plus âgés vont trouver que l'image pixellisée n'est vraiment pas assez précise. L'enseignant peut alors leur demander de trouver des astuces permettant d'améliorer ce résultat. Deux pistes peuvent surgir : soit on utilise des galets (pixels) d'autres couleurs, soit on utilise plus de galets.
Cette seconde option, conforme au contexte de l'histoire, permet de toucher à la notion de « résolution » : en augmentant le nombre de cases, on peut affiner le dessin de l'image et la rendre encore plus facilement identifiable (mais le nombre de pixels augmente très vite : une grille de 7x7 contient 49 pixels, et en doublant chaque dimension on doit dessiner 196 pixels, et ainsi de suite…)

Les élèves se voient alors attribuer une seconde grille, plus fine, de 14x14 pixels (bas de la Fiche 17). Ils doivent répéter l'opération : pixelliser l'image initiale sur cette nouvelle grille. Encore une fois, il faudra veiller à ce que les élèves ne décalquent pas simplement le dessin fourni.



A gauche, le chapeau de magicien pixellisé sur une grille 7x7. A droite, le même chapeau pixellisé sur une grille plus fine.

Conclusion et traces écrites
La classe synthétise collectivement ce qui a été appris au cours de cette séance :
O*n peut représenter une image par une grille de pixels noirs ou blancs.*
Les élèves notent ces conclusions dans leur cahier de sciences.

Retour **SOMMAIRE**

Séance 5 au Centre Pilote Découverte et programmation Scratch Atelier 1 : 2h00	
Prérequis	- Utiliser le clavier et la souris - Savoir lancer un programme en double-cliquant sur son icône ; - Savoir enregistrer son travail dans un fichier, et ce fichier dans un dossier (ou répertoire) ; - Savoir récupérer le travail que l'on a enregistré auparavant.
Objectifs	Découvrir et utiliser *Scratch*, un environnement de programmation graphique simple d'utilisation.
Notions	« Machines » et « Langages » - On peut donner des instructions à une machine en utilisant un langage spécial, appelé langage de programmation, compréhensible par l'homme et la machine. - Un "algorithme" est une méthode permettant de résoudre un problème - Un programme est un algorithme exprimé dans un langage de programmation - Des instructions peuvent démarrer en même temps, si leur exécution est déclenchée par un même événement. - Certaines boucles sont répétées indéfiniment. - Certaines boucles sont répétées jusqu'à ce qu'une condition soit remplie.
Matériel	- Travailler en demi-groupe, un ordinateur pour 2 élèves - Scratch doit être installé sur toutes les machines - Fiche 32 par élève - Annexe EXO SCRATCH
Phases de déroulement de l'activité	L'enseignant explique aux enfants qu'ils vont utiliser un ordinateur pour raconter les principaux épisodes de l'histoire de leur héros. Pour cela, ils vont devoir programmer l'ordinateur, c'est-à-dire lui dire quoi faire. Il faudra utiliser un langage spécial, un langage de programmation, compréhensible à la fois par les enfants et par l'ordinateur. Le langage qu'ils vont utiliser s'appelle *Scratch.* **Phase de découverte** Après une présentation rapide de *Scratch*, les élèves explorent librement le logiciel, puis exécutent des exercices simples, en particulier les tâches suivantes : - **Tâche 1 :** lancer *Scratch* et découvrir son interface. L'enseignant réalise une démonstration rapide de *Scratch* : présentation des différents éléments (scènes, lutins, arrière-plans, onglet « script ») et d'actions élémentaires (glisser des instructions dans la zone de programmation, combiner des instructions en sous-programmes, supprimer des instructions), L'enseignant distribue alors la Fiche 32, que les élèves colorieront au fur et à mesure de la séquence.

- **Tâche 2 :** explorer librement *Scratch*. Les élèves découvrent les différentes catégories d'instructions (« mouvement », « apparence », « évènement », « contrôle »…)

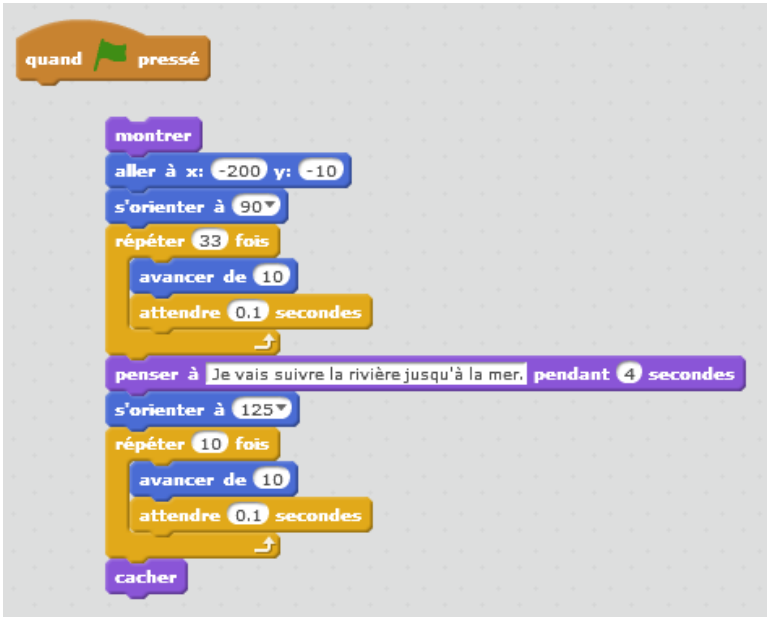
- **Tâche 3 :** s'approprier l'interface en résolvant des petits exercices progressifs (annexe EXO SCRATCH)

Prise en main et réalisation du récit
Les élèves explorent les moyens de contrôler les mouvements d'un personnage.
Puis, ils racontent un premier passage de l'aventure de leur héros, dans lequel, tout juste sorti de la forêt, il longe la rivière jusqu'à la mer. Ce faisant, ils réinvestissent des notions vues à la séance précédente (séquence d'instructions et événement), découvrent la notion d'initialisation et utilisent des boucles prédéfinies « répéter … fois ».

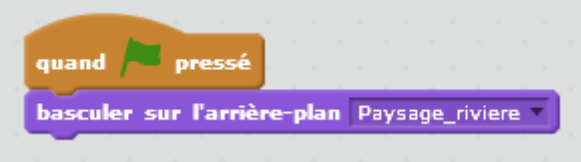
Pour aller plus loin… Changer de scène et ajouter un lutin
Tout d'abord, l'enseignant montre à la classe, à l'aide de l'ordinateur de démonstration, comment charger une nouvelle scène. Il faut cliquer sur la scène actuelle (la rivière), cliquer sur l'onglet « arrière-plans », et charger un arrière-plan à partir d'un fichier. Les élèves reproduisent le tout sur leur propre ordinateur.

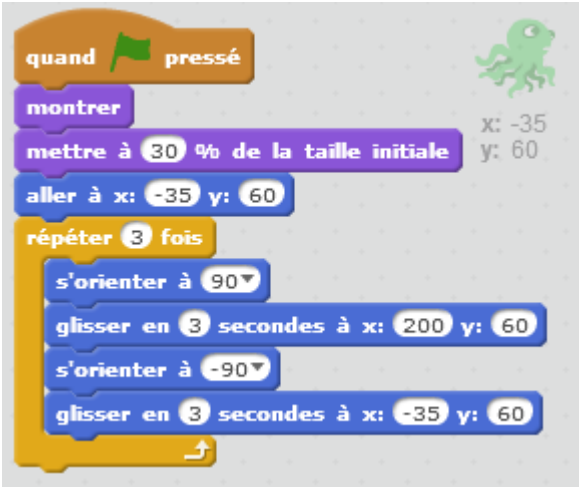
L'enseignant explique enfin que pour le moment, on va inactiver le script du héros correspondant à l'épisode de la rivière : pour cela, il suffit de séparer tout le bloc d'instructions de l'événement déclencheur « Quand drapeau vert pressé », comme ceci :

Exemple :

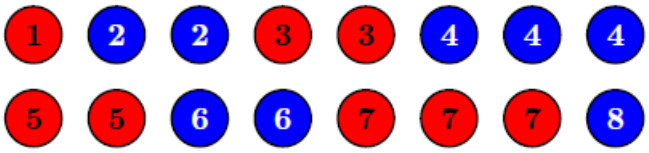


Il est temps pour les élèves de programmer le nouvel épisode : le héros doit se rendre tout au bout du ponton, où il rencontre la pieuvre faisant des allers-retours de droite à gauche. Lorsque le héros arrive au bout du ponton, la pieuvre s'arrête à proximité de lui. Suite à un court dialogue, la pieuvre disparait (elle part chercher le trésor).

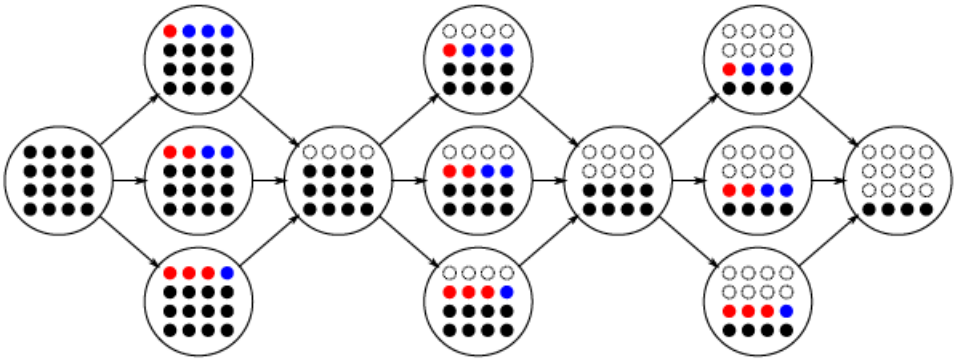
	Note pédagogique Nous proposons ici de laisser les élèves chercher par eux-mêmes. Toutefois, en cas de trop grande difficulté, ne pas hésiter à rappeler les instructions utiles. **Coordonner les épisodes** L'enseignant rappelle que pour le moment, le programme du héros correspondant à l'épisode de la rivière est désactivé (aucun événement ne le déclenche). On voudrait pourtant que les deux épisodes du récit s'enchainent. Solution attendue : Dans la zone de programmation de la scène on initialise la scène pour le début du récit :  **Conclusions et traces** La classe synthétise collectivement ce qui a été appris au cours de cette séance : - On peut donner des instructions à une machine en utilisant un langage spécial, appelé langage de programmation, compréhensible par l'homme et la machine. - Un « algorithme » est une méthode permettant de résoudre un problème - Un programme est un algorithme exprimé dans un langage de programmation. - *Les sous-programmes sont déclenchés par des événements. Dans un même programme, un seul événement peut déclencher plusieurs sous-programmes à la fois.*
Remarques	De nombreuses séances sont nécessaires pour utiliser et comprendre le logiciel Scratch. D'autres activités complémentaires sont annexées à ce parcours, permettant de poursuivre le travail sur ce logiciel, en classe. Les exercices sont progressifs, à réaliser en autonomie ou de façon accompagnée. Par ailleurs, d'autres jeux sont proposés en ligne : - http://castor-informatique.fr - http://code.org

Remarques Pour aller plus loin…	Peu d'élèves se préoccupent d'initialiser la position du lutin au démarrage au programme. Faire ré-émerger la notion si besoin : « Comment avons-nous fait pour que le héros soit toujours visible au début de l'épisode de la rivière ? », « Comment avons-nous fait pour que le héros soit toujours à la lisière de la forêt au début de l'épisode de la rivière ? ». Il faut tout simplement préciser les valeurs des coordonnées X et Y souhaitées, à l'aide de l'instruction « aller à x :… y:.. »). Peu d'élèves pensent à choisir le style de rotation de la pieuvre. Elle se retrouve alors la tête en bas lorsqu'elle se déplace vers la gauche. L'enseignant peut introduire l'instruction « glisser en … secondes à x : … y : … » pour simplifier le programme de la pieuvre, dont la version ci-dessus comporte des boucles imbriquées. Cela donne par exemple : 

Retour **SOMMAIRE**

Séance 6 au Centre Pilote Algorithme débranché Atelier 2 : 1h00	
Objectifs	Découvrir, définir l'algorithme. Résoudre des problèmes
Matériel	- 16 petits objets (clous, allumettes…) par binôme
Phases de déroulement de l'activité	**Le jeu de Nim** **Présentation du jeu et ses règles :** Voici un premier petit jeu simple, pour rentrer dans le sujet. On dispose sur une table 16 objets. Chacun leur tour, les deux joueurs ramassent un, deux ou trois objets sur la table. Le joueur qui ramasse le dernier objet remporte la partie.  Le joueur bleu gagne **Phase d'appropriation** Les élèves jouent par binômes. Qui a gagné le plus souvent ? **Phase de recherche** L'enseignant joue plusieurs parties et bien sûr, gagne ! *Quelle est la stratégie qui permet de gagner à coup sûr ?* **Mise en commun** Les élèves exposent leur solution. **Correction Stratégie gagnante** Le jeu de Nim est sans suspense : le premier à jouer perd, car il existe une astuce pour que le deuxième joueur gagne à tous les coups. La stratégie gagnante est de laisser 4, 8, 12 ou 16 objets à l'adversaire (un multiple de 4). Pour se convaincre de l'efficacité de la stratégie gagnante, prenons le dernier tour comme exemple. Il reste 4 objets, et J1 joue : • si J1 prend 1 objet, J2 en prend 3 (dont le dernier) ; • si J1 prend 2 objets, J2 en prend 2 (dont le dernier) ; • si J1 prend 3 objets, J2 en prend 1 (le dernier). Dans ce cas, si J2 sait jouer, J1 perd à tous les coups. En appliquant la même méthode, J2 peut guider le jeu de manière à passer de 16 objets à 12, puis 8 et enfin 4. Donc, si J2 sait jouer, J1 a perdu la partie avant même de commencer.

Pour amener les participants à découvrir la stratégie gagnante, on peut grouper les objets par 4, rendant ainsi l'astuce plus visible.

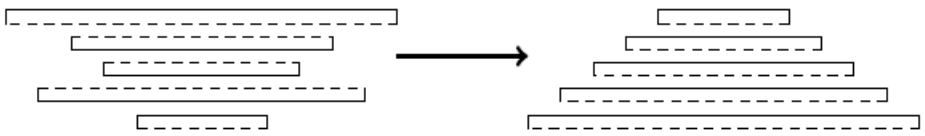


Les élèves rejouent en essayant …d'atteindre le plus possible la victoire !

Conclusion

Comme pour le jeu de Nim, un algorithme est une stratégie gagnante permettant de trouver la solution à un problème donné. Dans l'exemple précédent, le problème était « comment gagner au jeu de Nim ? »

Retour **SOMMAIRE**

Séance 7 au Centre Pilote Algorithme débranché Atelier 3 : 1h00	
Objectifs	Découvrir, définir l'algorithme. Résoudre des problèmes
Matériel	- Un lot de disques de feutrine, colorés, de tailles différentes, par binôme - Une pelle à tarte ou spatule par binôme
Phases de déroulement de l'activité	**Le jeu du crêpier** **Présentation du jeu et ses règles :** A la fin de sa journée, un crêpier dispose d'une pile de crêpes désordonnée. Le crêpier étant un peu psycho-rigide, il décide de ranger sa pile de crêpes, de la plus grande (en bas) à la plus petite (en haut), avec le coté brûlé caché.  Pour cette tâche, le crêpier peut faire une seule action : glisser sa spatule entre deux crêpes et retourner le haut de la pile. Comment doit-il procéder pour trier toute la pile ?  **Phase d'appropriation** Les élèves jouent par binômes. *Quel groupe est parvenu à replacer le plus grand nombre de crêpes ?* **Phase de recherche** *Quelle est la meilleure stratégie ?* Si certains bloquent, conseillez-les. Par exemple : « *essaye d'abord de mettre la grande crêpe en bas* », ou encore « *où doit se trouver la grande crêpe pour pouvoir l'amener en bas ?* » **Mise en commun** Les élèves exposent leur solution. **Correction Stratégie gagnante** **Description d'un algorithme** L'algorithme permettant de résoudre le problème du crêpier est le suivant :

	1. amener la plus grande crêpe en haut de la pile 2. mettre la face brûlée vers le haut 3. retourner toute la pile - la crêpe est rangée 4. recommencer en ignorant les crêpes rangées Cet algorithme assez simple nous apprend deux choses. Premièrement, un algorithme n'a d'intérêt que si on peut l'expliquer - pire encore, que si on peut l'expliquer à un ordinateur. Il doit donc être écrit sans ambiguïté. Deuxièmement, un algorithme décompose le problème en une série de tâches simples. On appelle ce principe « **Diviser pour mieux régner** ».Les élèves rejouent en essayant …d'atteindre le plus possible la victoire ! Demandez ensuite de calculer le nombre de coups nécessaires pour ranger la pile de crêpes. Le nombre de coups dépendant de l'état initial, faites-les généraliser en trouvant le nombre de coups maximal pour ranger une crêpe, puis n crêpes… **Conclusion** Le premier objectif de l'écriture d'un algorithme est qu'il résolve le problème. Le second objectif est qu'il le résolve le plus vite possible. De plus, s'il existe plusieurs algorithmes résolvant le même problème, l'évaluation de la performance nous donne un critère objectif pour savoir lequel est le plus efficace. https://www.youtube.com/watch?v=o6-4g4I6mOg **Autre problème s'il reste un peu de temps :** Une grenouille tombe au fond d'un puits de 50 mètres de profondeur. Pour s'en sortir chaque jour elle monte de 3 mètres, mais chaque nuit elle glisse de 2 mètres... Au bout de combien de jours la grenouille va-t-elle sortir du puits ?

Retour **SOMMAIRE**

Séances 8 et 9 en classe Découverte du robot Thymio	
Objectifs	Se familiariser avec le robot, observer, analyser ses déplacements.
Notions	« Machines » - Les machines qui nous entourent ne font qu'exécuter des "ordres" (instructions). « Robot » - Un robot est une machine qui peut interagir avec son environnement. - Un robot possède des capteurs qui lui permettent de percevoir son environnement. - Un robot peut effectuer des actions : bouger, produire un son, émettre de la lumière… - Un robot possède un ordinateur qui décide quelles actions faire dans quelles situations. - Si on compare un robot à un animal, on peut dire que : o ses capteurs sont ses organes sensoriels o ses moteurs sont comme ses muscles o son ordinateur est comme son cerveau o l'assemblage de ses pièces est comme son corps « Algorithmes » - Dans un programme, des tests disent quelle instruction effectuer quand une condition est vérifiée.
Matériel	Par élève - Fiche 23 - Cahiers d'expériences. Pour la classe - Un robot Thymio - Des objets pouvant servir d'obstacles - Travail en groupes…
	Situation déclenchante L'enseignant présente Thymio éteint aux élèves. Les élèves découvrent comme allumer, éteindre, faire changer de couleur le Thymio. Puis, l'enseignant distribue la Fiche 23 aux élèves répartis en groupes de 4 voire 5. Chaque groupe va devoir étudier une couleur de Thymio (vert, rouge, violet, jaune), en décrire le comportement, et compléter la Fiche 23 en reliant les paires évènements/actions.

Phases de déroulement de l'activité	**Mise en commun** La mise en commun permet d'introduire le vocabulaire suivant : un test est composé d'une condition (« **SI** *Thymio vert détecte un objet devant lui* ») et d'une instruction à effectuer uniquement si la condition est vérifiée (« **ALORS** *il avance* »). La Fiche 23 est volontairement synthétique. De ce fait, elle ignore certains comportements de Thymio, qui peuvent être explorés et décrits oralement : - En mode « rouge » : Thymio se comporte différemment selon qu'un objet est placé « derrière lui, à droite » ou « derrière lui, à gauche ». À vous de trouver comment ! - En mode « violet » : le comportement des flèches n'est pas aussi simple que cela. En réalité, le déplacement est plus complexe que simplement avancer, être immobile, ou reculer. Thymio possède 3 vitesses dans chaque sens (vers l'avant ou vers l'arrière). Appuyer sur une flèche permet d'augmenter (flèche « avant ») ou de diminuer la vitesse (flèche « arrière »), de la même manière que sur une boîte de vitesse de moto. Si, par exemple, on avance en vitesse 3 (la plus grande), appuyer sur la flèche « arrière » ne fera pas reculer, mais ralentir d'un cran. Thymio avancera donc en vitesse 2. Ces vitesses peuvent être combinées avec les instructions « tourne à droite » ou « tourne à gauche » pour faire tourner Thymio plus ou moins vite. - En mode « jaune », le comportement de Thymio est moins caricatural que sur la fiche documentaire. Devant un obstacle placé juste sous son nez, Thymio fait parfois plusieurs essais avant, finalement, de décider de contourner l'obstacle. **Bilan : définition de ce qu'est un robot** Vocabulaire : capteur, programme, consignes… *Que peut-on faire avec les robots ?* *A quoi servent-ils ?* *Y en a-t-il chez nous ?*
Remarques	L'utilisation du robot Thymio peut permettre d'approfondir une nuance de langage, qui va revenir souvent dans les différentes séances : quelle est la différence entre « bouger » et « se déplacer » ? On utilise souvent le premier pour signifier le second, mais jamais l'inverse. Un bras mécanique vissé au sol ne pourra pas, par définition, se déplacer ; par contre, ses multiples articulations lui permettront de bouger : changer, ciller, pivoter, se déformer, s'agiter… Thymio, en faisant bouger ses roues (on dira « tourner » ses roues), peut en revanche se déplacer, voyager, se mouvoir, partir, aller, (re)venir…

Retour **SOMMAIRE**

Séance 10 en classe Programmer Thymio	
Objectifs	Comprendre les capteurs du robot, commencer à le programmer. Prendre conscience des langages.
Notions	« Machines » - Les machines qui nous entourent ne font qu'exécuter des "ordres" (instructions) « Langages» - On peut donner des instructions à une machine en utilisant un langage spécial, appelé langage de programmation, compréhensible par l'homme et la machine. Si on lance le même programme plusieurs fois, il donne toujours le même résultat. « Robot » - Un robot est une machine qui peut interagir avec son environnement - Un robot possède un ordinateur qui décide quelles actions faire dans quelles situations « Algorithmes » - Un test dit quelle action effectuer quand une condition est vérifiée.
Matériel	Par élève - Fiches 24 et 25 - Cahiers d'expériences. Pour le groupe - Un robot Thymio - Un ordinateur disposant du logiciel VPL (logiciel gratuit à télécharger à partir de la page https://www.thymio.org/fr:start (qui permet de choisir son environnement de travail…) - A la fin de l'installation, renommer le raccourci « Aseba » en « Thymio ». - Travail en groupes…
Phases de déroulement de l'activité	**Situation déclenchante** Lors des séances précédentes, les élèves ont manipulé le Thymio, puis découvert qu'un ordinateur dirigeait les actions du robot en fonction des détections de ses capteurs. Désormais, les élèves vont créer leurs propres programmes pour ordonner à Thymio de faire d'autres actions. Pour qu'élèves et Thymio se comprennent, l'enseignant présente le langage de programmation VPL. **Programmation** Pour apprivoiser le langage VPL, l'enseignant distribue aux élèves la Fiche

	24. Les élèves doivent commencer par s'approprier l'interface et la façon de créer des programmes avec les cartes fournies. En particulier, ils repèrent que les cartes de la colonne de gauche correspondent à divers évènements que les capteurs peuvent déclencher, tandis que les cartes de la colonne de droite correspondent à des actions. La classe s'intéresse ensuite à des programmes préexistants. L'enseignant distribue la Fiche 25. Les élèves doivent tester successivement les 4 programmes proposés et expliciter à l'écrit sous forme de test ce qu'ils font. Concrètement, pour tester l'effet d'un programme sur le Thymio, les élèves doivent : Supprimer les cartes déjà placées dans l'espace central de l'interface graphique en appuyant sur les croix de suppression correspondantes (voir Fiche 24); Positionner la carte événement et la carte action du programme à tester dans cet espace central ; Modifier ces cartes si nécessaire en sélectionnant des boutons et/ou en déplaçant des curseurs ; Lancer le programme en appuyant sur la flèche de lecture ; Poser le Thymio sur une surface plane, si possible sans le débrancher, pour observer/tester les effets du programme, en manipulant Thymio autant que nécessaire. Si les élèves débranchent Thymio, ils devront le rebrancher avant de tester le programme suivant. **Mise en commun** Au tableau, l'enseignant rassemble les descriptifs trouvés par la classe pour les différents programmes : - Programme 1 : SI on appuie sur le bouton central, ALORS Thymio avance - Programme 2 : SI Thymio détecte un objet devant lui, ALORS son capot devient vert - Programme 3 : SI Thymio détecte un objet en-dessous de lui, ALORS son châssis devient bleu - Programme 4 : SI on tapote sur le capot de Thymio, ALORS il joue de la musique **Conclusions** La classe synthétise collectivement ce qui a été appris au cours de cette séance : - Les machines qui nous entourent ne font qu'exécuter des ordres (instructions). - On donne les instructions à une machine en créant un programme, qui utilise un langage de programmation. - L'exécution d'un programme est reproductible (si les instructions ne changent pas, ni les données à manipuler, le programme donne toujours le même résultat).

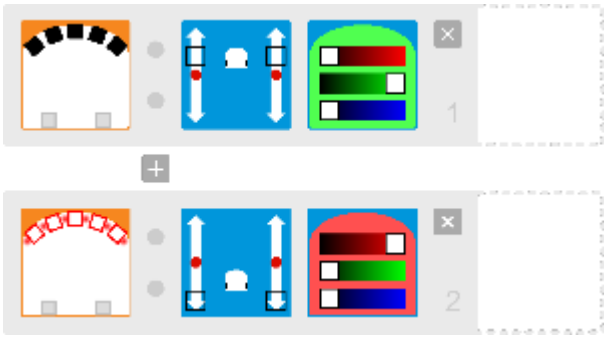
Remarques	Le Programme 3 fait appel aux capteurs de châssis de Thymio. Ce sont ces mêmes capteurs qui sont impliqués dans le mode cyan « pisteur ». Si on approche Thymio du bord de la table, il ne détecte rien. De même si on le pose sur une feuille noire! Par contre il détecte toute surface claire. Dans le cas des programmes 1, 2 et 3, Thymio ne revient jamais tout seul à son état d'origine (il ne s'arrête pas tant qu'on ne lui fournit pas d'instruction en ce sens, il ne revient pas à sa couleur d'origine). C'est tout à fait normal, car ce serait un autre comportement que l'on n'a pas encore programmé ici : quand il détecte un évènement, Thymio change de couleur ou se met en mouvement, et il s'y tient jusqu'à ce qu'un autre programme lui demande autre chose. Le retour à l'état d'origine est l'objet de la séance suivante.

Retour **SOMMAIRE**

Séance 11 au Centre Pilote Les capteurs de Thymio Atelier 1 : 1h00	
Objectifs	La programmation par VPL de Thymio est évènementielle : les élèves découvrent comment utiliser les états des capteurs de Thymio pour déclencher des actions précises
Notions	« Robot » - Un robot possède des capteurs qui lui permettent de percevoir son environnement. « Algorithmes » - Un test dit quelle action effectuer quand une condition est vérifiée.
Matériel	Par groupe de 3/4 élèves - Un Thymio - Un ordinateur disposant du logiciel VPL - Scratch doit être installé sur toutes les machines Par élève - Fiche 26
Phases de déroulement de l'activité	Lors de la séance précédente, les élèves ont pu programmer quelques comportements simples de Thymio : avancer, changer de couleur. Mais ils ont également repéré que Thymio ne revenait jamais dans son état initial : s'il commençait à avancer, rien dans son programme ne lui disait comment ou à quelle condition s'arrêter. **Expérimentation : détections et non-détections (par groupes)** L'enseignant distribue alors la Fiche 26. Chaque groupe va devoir tester les programmes proposés, toujours en effaçant les programmes précédents, et répondre aux questions posées. ***Note scientifique*** Pour la première fois, le Programme 5 contient plus qu'un test (il en contient 2). Il faut reproduire les deux tests l'un en dessous de l'autre pour que le programme soit complet. **Mise en commun** La classe réalise que VPL permet d'écrire des tests très précis, selon que les capteurs détectent quelque chose (icône rouge), détectent une absence (icône blanche), ou qu'on ne se préoccupe pas de leur état (icône grise). **Conclusion et traces écrites** La classe synthétise collectivement ce qui a été appris au cours de cette séance : *Quand un capteur détecte quelque chose, on dit qu'il y a un évènement.* *Une condition peut être « un évènement est arrivé » ou « un évènement n'est pas arrivé »*

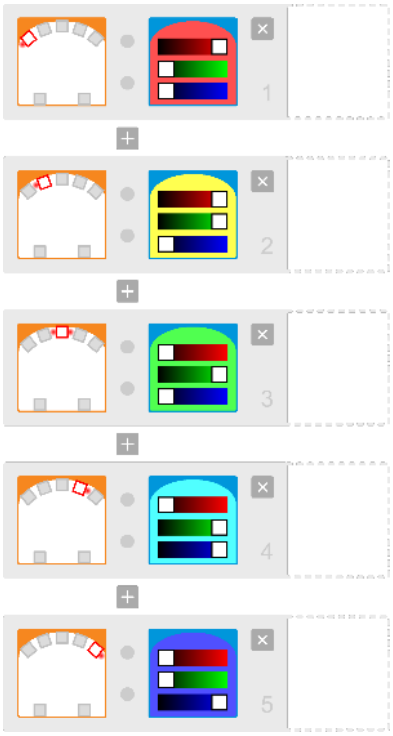
	Prolongement Les élèves les plus rapides peuvent appliquer cette découverte pour compléter les programmes de la Fiche 25 (séance précédente) : - Programme 1 : ajouter un test pour que Thymio s'arrête (quand on appuie sur une autre touche, par exemple). - Programme 2 : ajouter un test pour que Thymio ne soit plus vert (par exemple, qu'il devienne jaune) s'il ne détecte plus rien devant lui. - Programme 3 : ajouter un test pour que Thymio ne soit plus bleu s'il ne détecte rien sous lui.

Retour **SOMMAIRE**

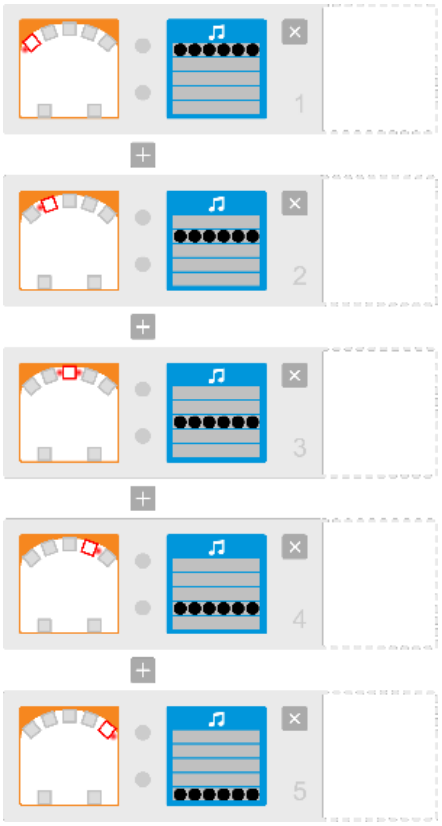
Séance 12 au Centre Pilote Programmons Thymio Atelier 2 : 1h00	
Objectifs	Les élèves relèvent de petits défis pour rédiger leurs premiers programmes VPL pour Thymio.
Notions	« Machines » et « Langages » - Les machines qui nous entourent ne font qu'exécuter des "ordres" (instructions) - On peut donner des instructions à une machine en utilisant un langage spécial, appelé langage de programmation, compréhensible par l'homme et la machine. « Robot » - Un robot est une machine qui peut interagir avec son environnement - Un robot possède un ordinateur qui décide quelles actions faire dans quelles situations - « Algorithmes » - Un test dit quelle action effectuer quand une condition est vérifiée
Matériel	Par groupe de 3/4 élèves - Un Thymio - Un ordinateur disposant du logiciel VPL
Phases de déroulement de l'activité	**Défis : réaliser de nouveaux programmes pour Thymio (par groupes)** Dans cette séance d'évaluation formative, les élèves réinvestissent les concepts découverts précédemment. L'enseignant va proposer trois défis successifs à la classe. Les élèves auront 20 minutes pour proposer un programme qui répond à chaque problème. **Défi 1 : Faire avancer Thymio** si son capteur avant ne détecte rien, et reculer si ce capteur détecte quelque chose. Associer une couleur à chacun de ces deux déplacements 

Défi 2 : Créer un sélecteur de couleur. À chaque capteur de l'avant de Thymio associer une couleur.

Dans le cadre de ce défi, les élèves peuvent réaliser qu'il y a une certaine priorité des tests. Par exemple, si deux conditions sont réalisées simultanément, avec des exécutions concurrentes, laquelle s'exécute ? Ici l'exécution affecte la couleur du Thymio : il ne peut pas avoir deux couleurs en même temps, laquelle choisit-il donc ? Réponse : VPL applique l'instruction de numéro la plus élevée. Imaginons qu'on active simultanément les capteurs « centre-droite » (instruction #3 : colorie Thymio en cyan) et « droite » (instruction #4 : colorie Thymio en bleu), alors Thymio se colorie en bleu. (On a ici concurrence des instructions #3 et #4 : c'est la #4 qui l'emporte.)



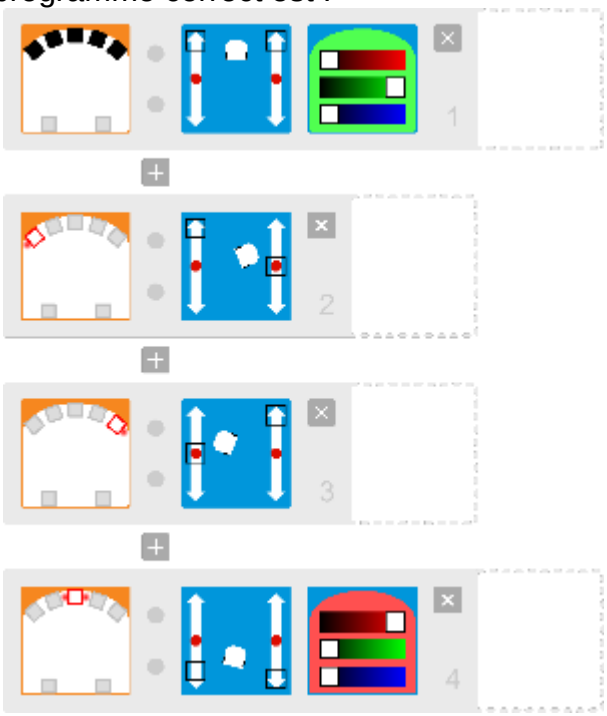
Défi 3 : Créer un instrument de musique. À chaque capteur associer un son.



Retour **SOMMAIRE**

Séance 13 au Centre Pilote Parcours d'obstacles Atelier 3 : 2h00	
Objectifs	Les élèves doivent reproduire le mode « explorateur » de Thymio jaune. Au cours de cette séance, ils écrivent le programme et testent leur programme sur un labyrinthe réel.
Notions	« Machines » et « Langages » - Les machines qui nous entourent ne font qu'exécuter des "ordres" (instructions) - En combinant plusieurs instructions simples on peut effectuer une tâche complexe - On peut donner des instructions à une machine en utilisant un langage spécial, appelé langage de programmation, compréhensible par l'homme et la machine. - Un bug est une erreur dans un programme. « Robot » - Un robot est une machine qui peut interagir avec son environnement - Un robot possède un ordinateur qui décide quelles actions faire dans quelles situations « Algorithmes » - Un test dit quelle action effectuer quand une condition est vérifiée
Matériel	Par groupe de 3/4 élèves - Un Thymio - Un ordinateur disposant du logiciel VPL - Fiche 27
Phases de déroulement de l'activité	L'enseignant rappelle aux élèves que Thymio était livré avec des modes pré-programmés. Il leur propose un défi : réussir à reprogrammer par eux-mêmes un équivalent (simplifié) du mode « jaune » du Thymio. Les élèves se souviennent qu'il s'agit du mode explorateur, où Thymio avance en esquivant les obstacles. **Défi : reproduire un Thymio explorateur (par groupes)** Selon l'aisance de la classe et l'âge des élèves, ce défi peut prendre plusieurs formes. Pour les élèves les plus autonomes, l'enseignant peut garder pour lui la Fiche 27 et l'utiliser comme aide-mémoire. Inversement, la programmation peut être plus guidée si l'enseignant distribue aux élèves cette même fiche. En groupes ou en classe entière, il faut d'abord arriver à conceptualiser les diverses étapes à programmer: *par défaut, que fait Thymio ? S'il détecte un obstacle à sa droite, que doit-il faire ? Et à gauche ? Devant lui ?* Puis il faut utiliser VPL pour programmer le robot, et tester si le programme marche en jouant avec Thymio sur la table.

Un exemple de programme correct est :



Expérimentation : un vrai test pour notre Thymio (par groupes)
La classe est maintenant face à un grand labyrinthe constitué d'obstacles dont la hauteur est d'au minimum de 6cm. Tous les groupes vont tester leur programme simultanément : les robots vont interagir avec le labyrinthe, et interagir entre eux.

Si cela s'y prête, le sol du labyrinthe peut être muni d'une surface où on peut dessiner : chaque groupe pourra insérer un feutre dans le trou prévu à cet effet sur le capot de Thymio. Cela permettra après coup de visualiser les chemins parcourus par les différents robots au cours de l'expérience.

Les groupes chargent le programme qu'ils ont conçu et laissent leur robot parcourir le labyrinthe. Ils peuvent améliorer leur programme au fur et à mesure des défauts observés : l'enseignant introduit alors le terme « bug » pour décrire ces dysfonctionnements.

Conclusion et traces écrites
La classe synthétise collectivement ce qui a été appris au cours de cette séance :

- *En combinant plusieurs instructions simples on peut effectuer une tâche complexe, comme parcourir un labyrinthe*
- *Un bug est une erreur dans un programme.*

Retour **SOMMAIRE**

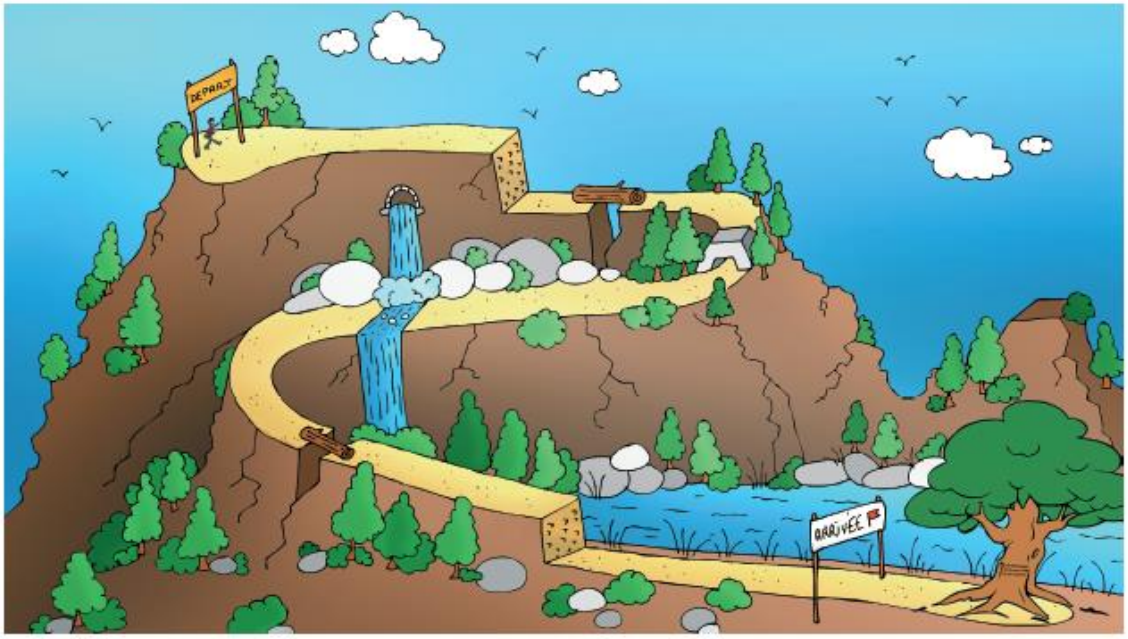
Facultatif…

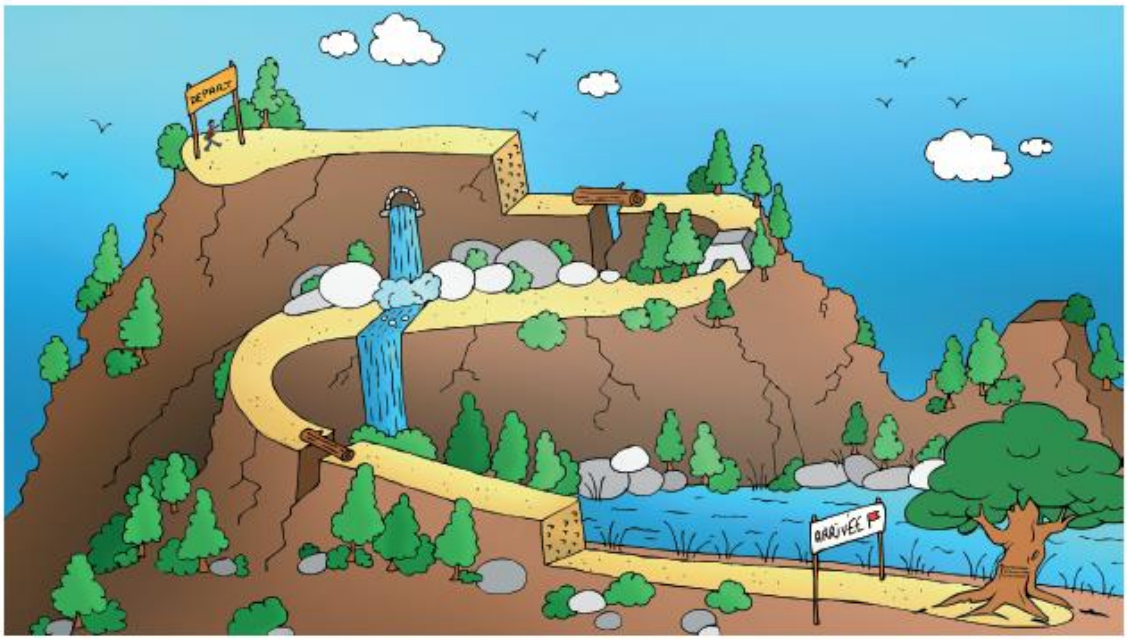
Séances 14 et… en classe Séances suivantes en classe : l'Histoire des sciences, histoire de l'informatique	
Objectifs	Comprendre l'histoire des sciences dans le domaine de l'informatique, le numérique. Placer les étapes sur un axe chronologique.
Matériel	- Fiches 47 à 57 - Cahiers d'expériences.
Phases de déroulement de l'activité	- Observation - Lecture de documents - Tri - Mise en ordre chronologique - Reconstitution de l'histoire des sciences…

Retour **SOMMAIRE**

ANNEXES

FICHE 12
Le parcours du héros





FICHE 13
Les instructions du héros

Consigne : Complète les cases de gauche en indiquant les obstacles que le héros peut rencontrer. Ensuite, écris dans les cases de droite des instructions qui lui permettront de passer ces obstacles.

SI le héros rencontre	*une crevasse*	ALORS il doit	*passer en équilibre sur le tronc d'arbre.*
SI le héros rencontre		ALORS il doit	
SI le héros rencontre		ALORS il doit	
SI le héros rencontre		ALORS il doit	
SI le héros rencontre		ALORS il doit	

FICHE 14
Une énigme à décoder

Consigne : Décode ce message en utilisant et en complétant la table de correspondance affichée au tableau.

19	21	9	19		12	1		18	9	22	9	5	18	5

10	21	19	17	21		1		12	1		13	5	18		

19	15	21	19		12	5	19		5	1	21	24		4	15	18	20

21	14		2	5	1	21		20	18	5	19	15	18

Le message gravé sur le tronc d'arbre

✂ -

1	2	3	4	5	6	7	8	9

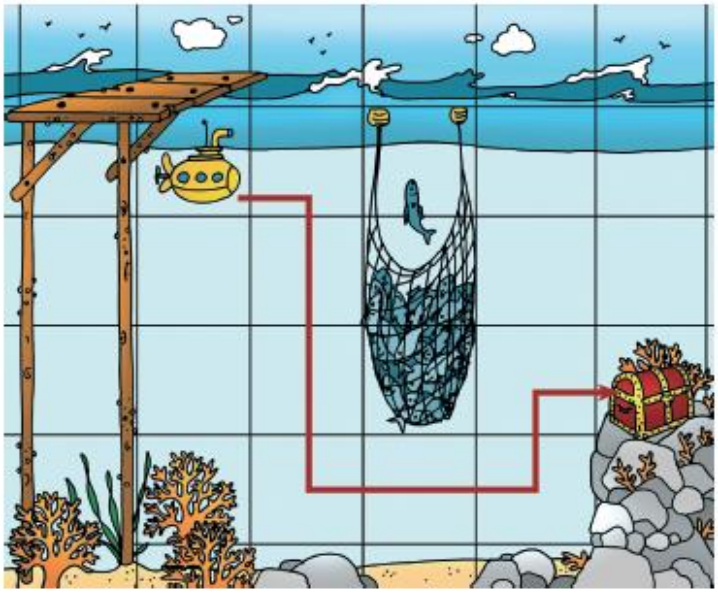
10	11	12	13	14	15	16	17	18

19	20	21	22	23	24	25	26

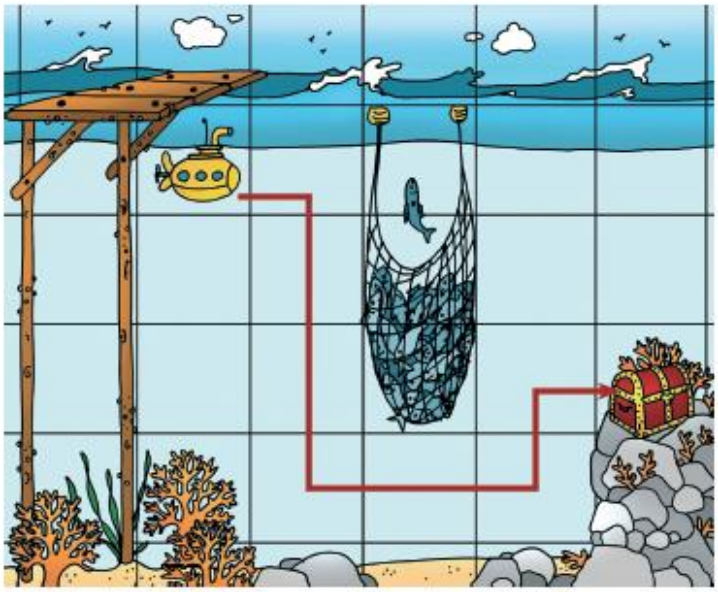
Table de correspondance pour déchiffrer le message du tronc d'arbre

FICHE 15
Le dédale des coraux

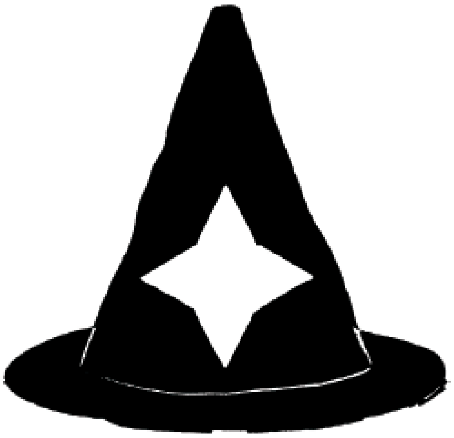
Consigne : Écris un programme (suite d'instructions) permettant au sous-marin de suivre ce parcours pour rejoindre le trésor. Attention, il ne peut se déplacer que d'une case à la fois et pas en diagonale.



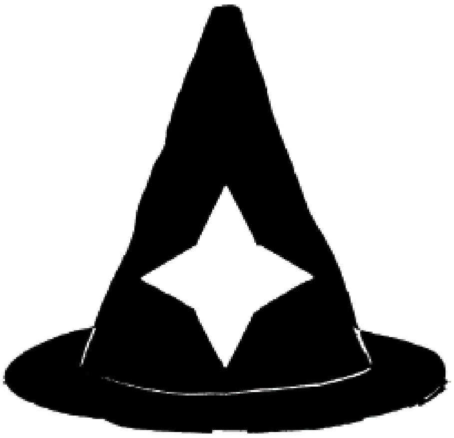
Consigne : Écris un programme (suite d'instructions) permettant au sous-marin de suivre ce parcours pour rejoindre le trésor. Attention, il ne peut se déplacer que d'une case à la fois et pas en diagonale.



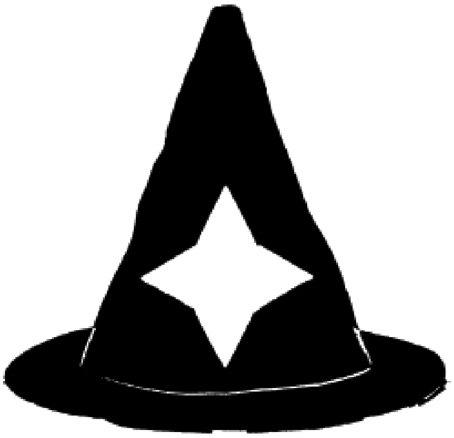
FICHE 16
Appelons le magicien : le chapeau à dessiner



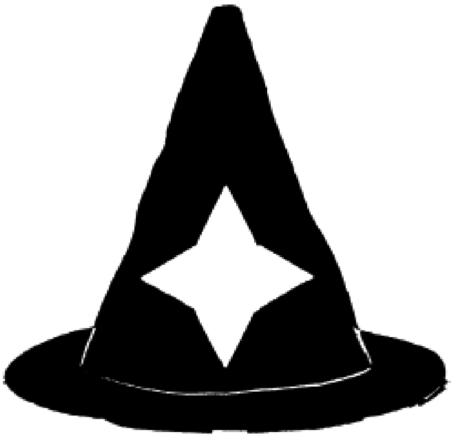
Consigne : Remplis certaines cases de la grille avec du noir pour imiter l'image du chapeau du magicien. Chaque case doit être entièrement blanche ou entièrement noire.



Consigne : Remplis certaines cases de la grille avec du noir pour imiter l'image du chapeau du magicien. Chaque case doit être entièrement blanche ou entièrement noire.

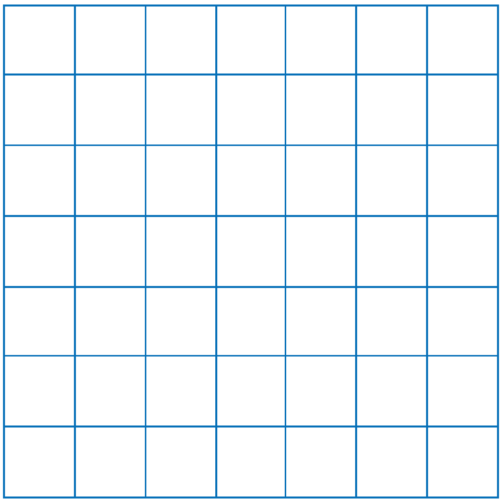


Consigne : Remplis certaines cases de la grille avec du noir pour imiter l'image du chapeau du magicien. Chaque case doit être entièrement blanche ou entièrement noire.

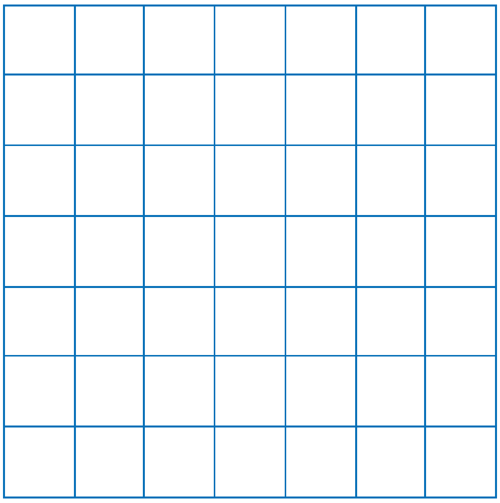


Consigne : Remplis certaines cases de la grille avec du noir pour imiter l'image du chapeau du magicien. Chaque case doit être entièrement blanche ou entièrement noire.

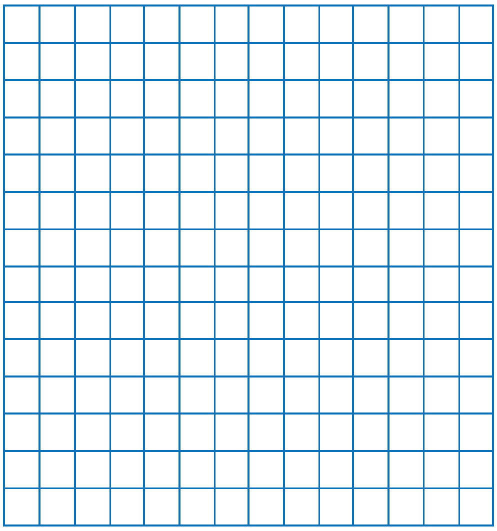
FICHE 17
Appelons le magicien : différentes grilles



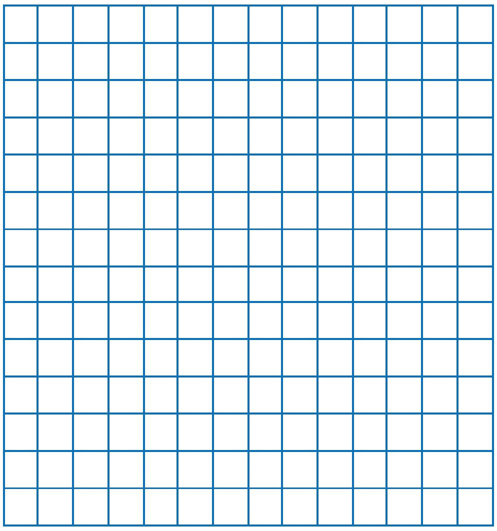
Consigne : Remplis certaines cases de la grille avec du noir pour imiter l'image du chapeau du magicien. Chaque case doit être entièrement blanche ou entièrement noire.



Consigne : Remplis certaines cases de la grille avec du noir pour imiter l'image du chapeau du magicien. Chaque case doit être entièrement blanche ou entièrement noire.

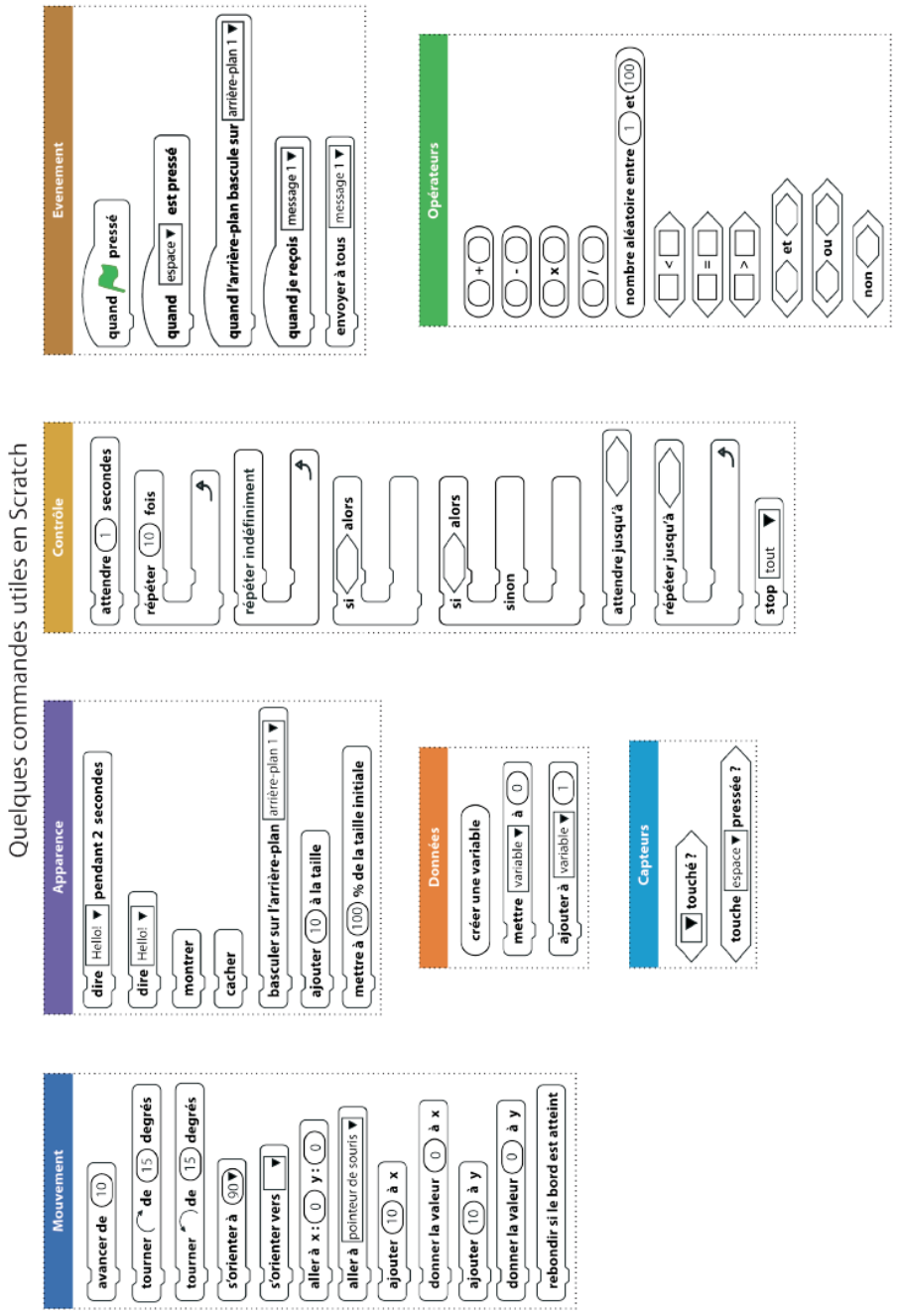


Consigne : Remplis certaines cases de la grille avec du noir pour imiter l'image du chapeau du magicien. Chaque case doit être entièrement blanche ou entièrement noire.



Consigne : Remplis certaines cases de la grille avec du noir pour imiter l'image du chapeau du magicien. Chaque case doit être entièrement blanche ou entièrement noire.

FICHE 32
Quelques commandes utiles en *Scratch*



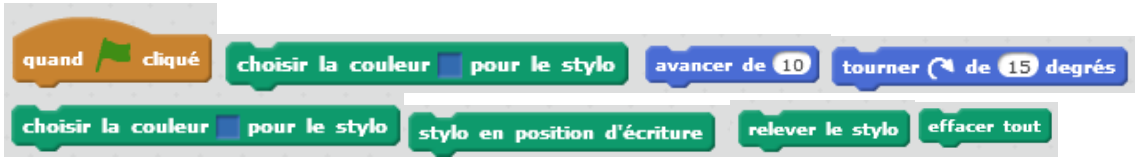
	Exercices Prise en main de SCRATCH

1. Écrire un script afin que le chat dise "Salut la foule" quand on presse sur la touche "ESPACE".

2. Ecrire un script respectant les contraintes suivantes :
   - Quand on presse sur le drapeau vert, le chat demande "Quel est le mot de passe ?".
   - Si l'utilisateur répond 1234 alors le chat affiche "BRAVO, tu as trouvé le bon code" et le lutin change de costume.
   - Sinon le chat affiche "DESOLE, ce n'est pas le bon code" et le lutin garde son costume.

3. Créer un nouveau lutin, programmer ce lutin pour qu'il dessine un carré rouge,

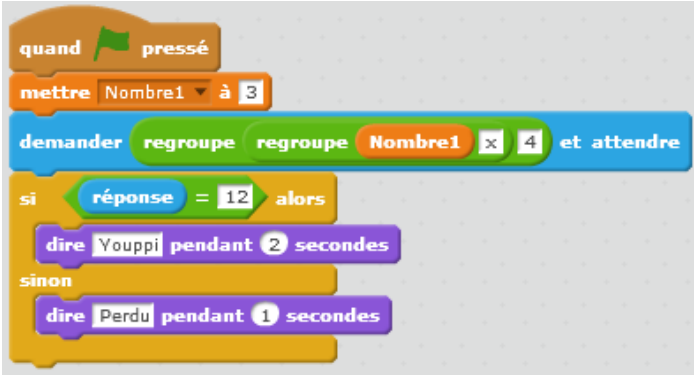
Utilise les blocs suivants :



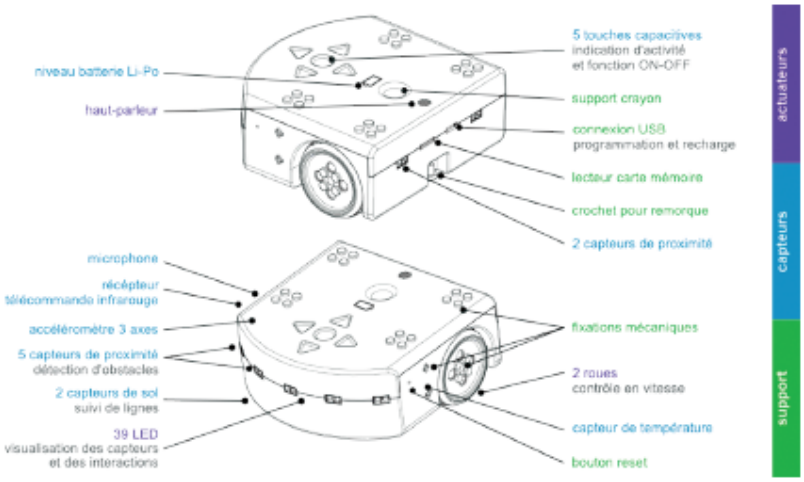
4. Faire un programme d'entrainement aux tables de multiplication pour les classes de 6èmes

On pourra commencer par la table de 4.

Voici le début d'un programme qui peut vous aider.



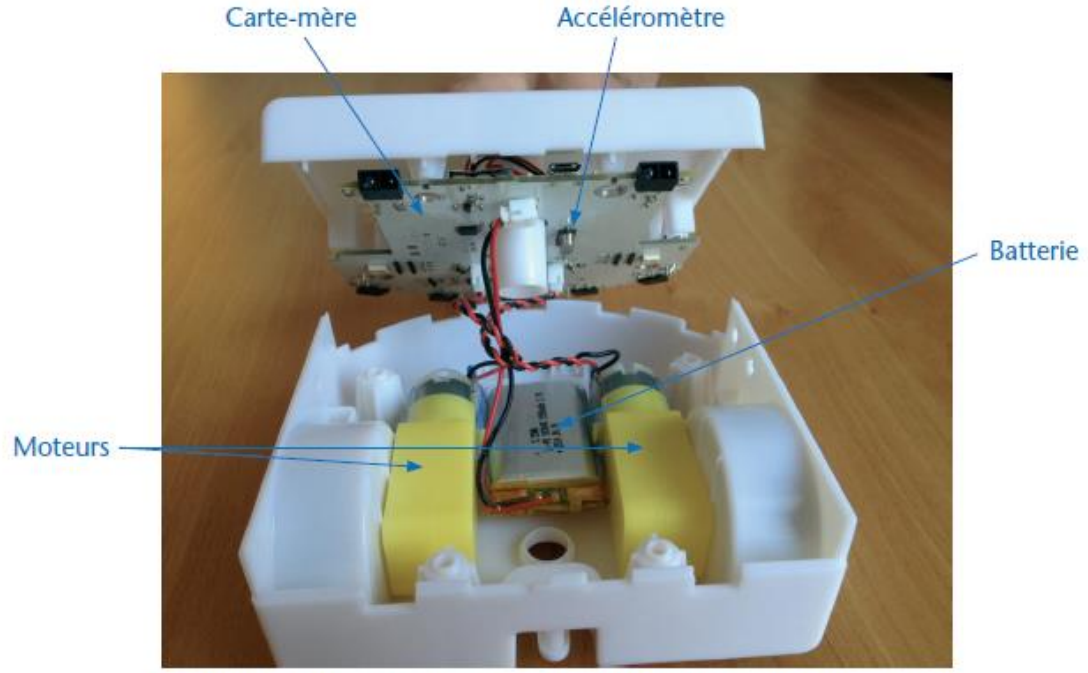
FICHE 8
Présentation rapide de Thymio



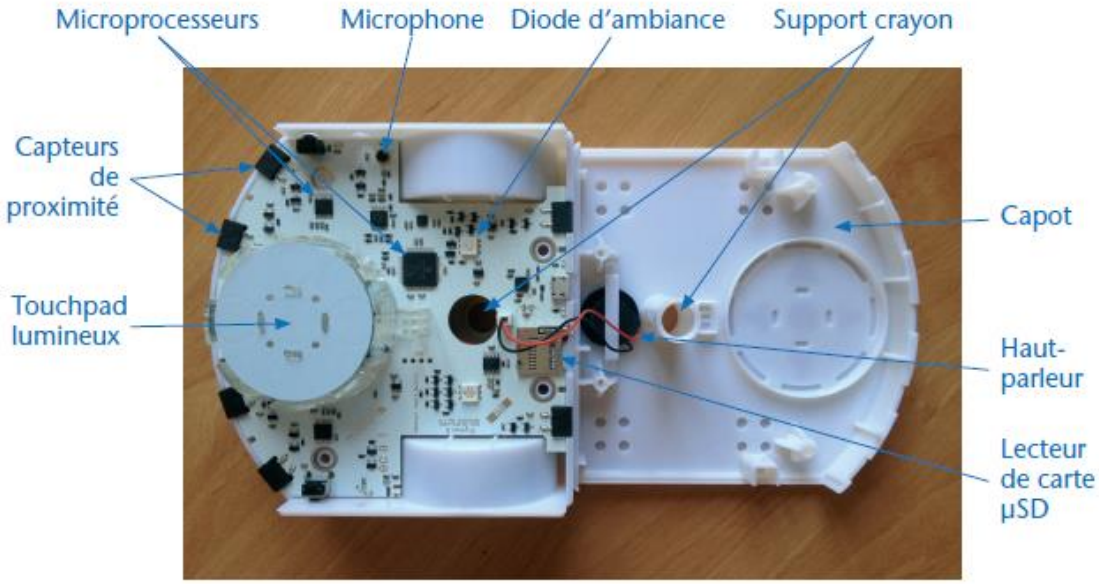
« Pour allumer le robot, il suffit d'appuyer et de maintenir le doigt sur le rond qui se trouve au centre des flèches jusqu'à ce que le robot émette un son et devienne vert. Cela prend quelques secondes.
Pour éteindre le robot, il suffit de maintenir le doigt sur le rond central quelques secondes jusqu'à ce que le robot joue une mélodie et s'éteigne complètement. »
(Citation : https://www.thymio.org/fr:thymiostarting)
Thymio est pré-programmé avec six comportements. Ces comportements sont toujours présents dans le robot. Pour choisir le comportement qu'adopte le Thymio, il suffit de démarrer le robot et de sélectionner une couleur grâce aux boutons flèches, le bouton central permettant de démarrer le comportement. Lorsque le comportement est actif, le bouton central permet de revenir au menu de sélection des comportements.

Mode	Couleur	Comportement
Amical	Vert	Thymio suit les obstacles qui bougent devant lui.
Explorateur	Jaune	Thymio explore au hasard et évite les obstacles.
Craintif	Rouge	Thymio fuit les obstacles situés devant ou derrière lui.
Pisteur	Cyan	Thymio suit une piste sombre sur fond clair dessinée au sol.
Obéissant	Violet	Thymio est dirigé manuellement grâce aux flèches situées sur son capot.
Attentif	Bleu	Thymio réagit aux sons : en fonction du nombre de clappements de mains qu'il entend, il peut tourner, avancer, s'arrêter, faire un cercle.

FICHE 9
Dissection d'un Thymio



Le châssis du Thymio: la batterie (au centre) alimente les deux moteurs (en jaune) qui permettent de faire tourner les roues



La carte-mère du Thymio, qui porte les capteurs infrarouges, le touchpad central lumineux, les microprocesseurs, les diodes

FICHE 23
Je découvre Thymio

Consigne : Allume ton Thymio et teste les différents modes qu'il propose. Trouve un surnom pour chaque mode. Relie ensuite les paires d'événements et d'actions en fonction de ce que tu observes.

Thymio vert

Surnom :

SI Thymio détecte un objet devant lui	●	● **ALORS** il tourne à gauche
SI Thymio détecte un objet à droite	●	● **ALORS** il tourneà droite
SI Thymio détecte un objet à gauche	●	● **ALORS** il avance

Thymio rouge

Surnom :

SI Thymio détecte un objet devant lui	●	● **ALORS** il recule
SI Thymio détecte un objet à droite	●	● **ALORS** il recule en tournant à droite
SI Thymio détecte un objet à gauche	●	● **ALORS** il recule en tournant à gauche
SI Thymio détecte un objet derrière lui	●	● **ALORS** il avance

Thymio violet (départ arrêté)

Surnom :

SI on appuie sur la flèche avant	●	● **ALORS** il avance
SI on appuie sur la flèche arrière	●	● **ALORS** il recule
SI on appuie sur la flèche de droite	●	● **ALORS** il tourne à gauche
SI on appuie sur la flèche de gauche	●	● **ALORS** il tourne à droite

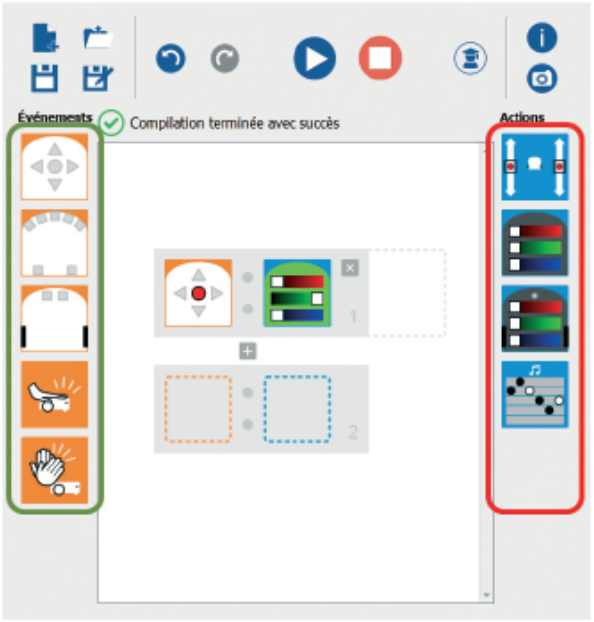
Thymio jaune

Surnom :

SI Thymio détecte un objet devant lui	●	● **ALORS** il tourne à gauche
SI Thymio détecte un objet à droite	●	● **ALORS** il tourne à droite
SI Thymio détecte un objet à gauche	●	● **ALORS** il recule
SI Thymio ne détecte rien	●	● **ALORS** il avance

FICHE 24
Programmer Thymio : découvrir l'interface VPL

Consigne : Place deux cartes au centre et modifie-les pour reproduire le programme ci-dessous. Entoure ensuite les bonnes réponses dans les phrases proposées.



Le bouton sert à : *Démarrer le programme* *Arrêter le programme*

Le bouton sert à : *Démarrer le programme* *Arrêter le programme*

Les images dans le cadre en vert concernent les : *Actions* *Capteurs*

Les images dans le cadre en rouge concernent les : *Actions* *Capteurs*

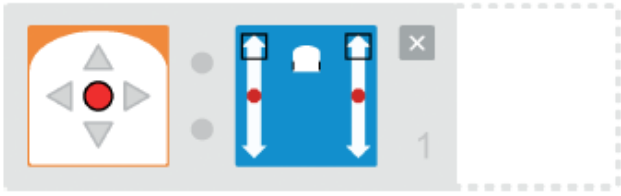
Le bouton « + » encadré en vert sert à : *Supprimer un ordre* *Ajouter un ordre*

Le bouton « × » encadré en rouge sert à : *Supprimer un ordre* *Ajouter un ordre*

FICHE 25
Mes premiers programmes pour Thymio

Consigne : Voici 4 programmes différents, chacun formé avec une carte événement et une carte action. Teste-les sur ton Thymio, puis complète les phrases associées pour décrire ce que fait chaque programme.

Programme 1 :



SI ALORS

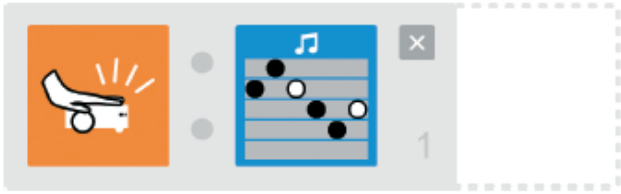
Programme 2 :

SI ALORS

Programme 3 :

SI ALORS

Programme 4 :

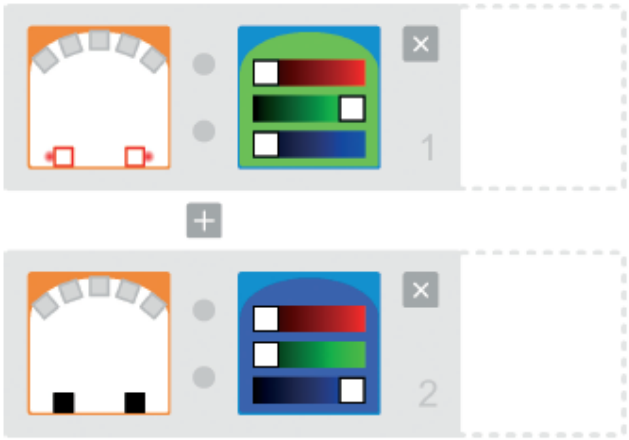


SI ALORS

FICHE 26
Testons les capteurs de Thymio

Consigne : Voici 2 programmes différents, le programme 5, constitué de 2 tests et le programme 6, constitué d'un seul test. Essaie-les sur ton Thymio, puis réponds aux questions posées.

Programme 5 :



Entoure la bonne réponse :

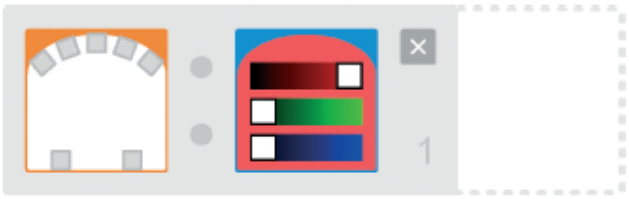
De quelle couleur est le Thymio lorsque ta main est devant les deux capteurs de l'arrière ?

VERT / BLEU

De quelle couleur est le Thymio lorsque ta main n'est pas devant les capteurs de l'arrière ?

VERT / BLEU

Programme 6 :



Réponds aux questions :

De quelle couleur est le Thymio lorsque ta main est devant les capteurs de l'arrière ?

.....

De quelle couleur est le Thymio lorsque ta main n'est pas devant les capteurs de l'arrière ?

.....

Relie les icônes à ce qu'elles veulent dire.

L'icône  signifie ● ● « SI le capteur détecte ou ne détecte pas… »

L'icône  signifie ● ● « SI le capteur ne détecte pas… »

L'icône  signifie ● ● « SI le capteur détecte… »

FICHE 27
Programmer un Thymio « explorateur »

1. 

Créer une instruction pour que Thymio avance s'il ne détecte rien avec ses capteurs de devant

2. 

Ajouter une instruction pour que Thymio tourne à droite lorsqu'il détecte quelque chose à gauche

3. 

Ajouter une instruction pour que Thymio tourne à gauche lorsqu'il détecte quelque chose à droite

4. 

Ajouter une instruction pour que Thymio recule légèrement tout en tournant un peu s'il détecte quelque chose devant lui

5. (FACULTATIF)

Ajouter des instructions pour que Thymio s'allume en rouge s'il détecte un obstacle, et en vert sinon