



1, 2, 3, codez ! - Activités cycle 3 - Etape 2.3: Piloter le rover

Résumé	Les élèves réalisent leur premier programme, leur permettant de piloter le rover à l'aide des flèches. Ils se familiarisent avec le système de coordonnées.
Notions	<p>« Machines » :</p> <p>Les machines qui nous entourent ne font qu'exécuter des ordres (instructions) En combinant des instructions élémentaires, nous pouvons leur faire exécuter des tâches complexes</p> <p>« Algorithmes »</p> <p>Un algorithme est une méthode permettant de résoudre un problème. Une boucle permet de répéter plusieurs fois la même action Certaines boucles, dites « infinies », ne s'arrêtent jamais. Certaines boucles, dites « itératives » sont répétées un nombre prédéfini de fois.</p> <p>« Langages » :</p> <p>Pour donner des instructions à une machine, on utilise un langage de programmation, compréhensible à la fois par la machine et par l'être humain <i>Scratch</i> est un environnement de programmation graphique, qui utilise un langage simple. Un programme est l'expression d'un algorithme dans un langage de programmation Certaines instructions ne s'exécutent qu'au déclenchement d'un évènement : on parle de programmation événementielle. Certaines instructions s'exécutent à la suite les unes des autres : on parle de programmation séquentielle. 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)</p>
Matériel	<p>Pour la classe</p> <p>Un vidéo projecteur Version agrandie de la Fiche 32</p> <p>Pour chaque binôme</p> <p>Un ordinateur connecté à Internet ou, en l'absence de connexion Internet de bonne qualité, un ordinateur sur lequel le logiciel <i>Scratch</i> a été préalablement installé .</p> <p>Pour chaque élève</p> <p>Fiche 32 une photocopie de la Fiche 33</p>

Une fois que chaque binôme a réussi l'importation de son programme (ne contenant, pour l'instant, que le rover et l'arrière-plan), la classe revient sur la liste des étapes à réaliser pour programmer le jeu vidéo. La première chose à faire, c'est de piloter le rover. Le plus simple est de piloter le rover à l'aide des touches « flèches » du clavier.

Note pédagogique :

Pour cette étape, les élèves auront encore besoin d'être guidés. Ensuite, ils auront acquis les automatismes leur permettant d'être bien plus autonomes, et chaque binôme avancera à son rythme.



Tâche 1 : faire avancer le rover vers la gauche (10 minutes)

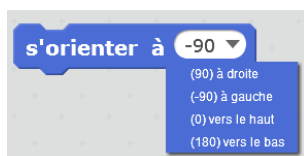
Les élèves savent déjà comment faire avancer le rover vers la droite... il suffit de lui dire d'avancer, puisque, par défaut, il est déjà orienté vers la droite. Le faire avancer vers la gauche est une tâche un peu plus difficile, car les élèves doivent d'abord demander au rover de s'orienter vers la gauche, avant d'avancer. Ils travaillent en autonomie et tâtonnent, l'enseignant passant régulièrement dans les groupes pour s'assurer que personne n'est bloqué. Il peut les guider en les incitant à chercher une instruction « s'orienter ».

Notes pédagogiques :

Deux instructions de ce type sont disponibles :

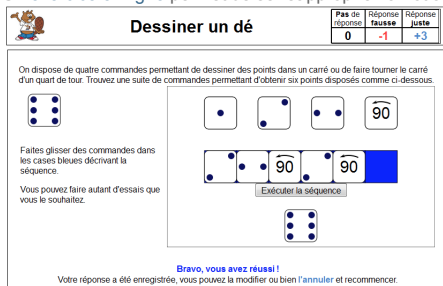
« s'orienter vers... » qui ne nous intéresse pas car la seule option disponible, accessible en cliquant sur la petite flèche, est « pointeur de souris » (le lutin, dans ce cas, s'oriente vers la position du pointeur de la souris).

« s'orienter à... » qui est celle qui nous intéresse. Lorsque l'on clique sur le nombre présent dans l'instruction (en général, ce nombre par défaut est « 90 »), une bulle d'aide nous explique que l'angle 0° désigne le haut de l'écran ; 90° désigne la droite, etc.

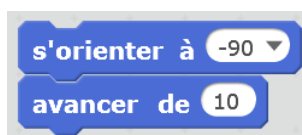


Il faut donc choisir ici « s'orienter à -90 »

Un [exercice en ligne](#) permet de se réapproprier la notion d'orientation. Attention, cet exercice n'utilise pas Scratch.



Finalement, le programme qui permet au lutin de se déplacer vers la gauche est :



Tâche 2 : faire avancer le rover dans n'importe quelle direction (5 minutes)

Les élèves doivent maintenant être capables de faire avancer le rover dans n'importe quelle direction (droite, gauche, haut et bas), en reprenant exactement la même méthode que précédemment.

NB : désormais, on a besoin de l'instruction « s'orienter à 90 » pour lui dire d'aller à droite... car le lutin n'est plus orienté, par défaut, dans cette direction.



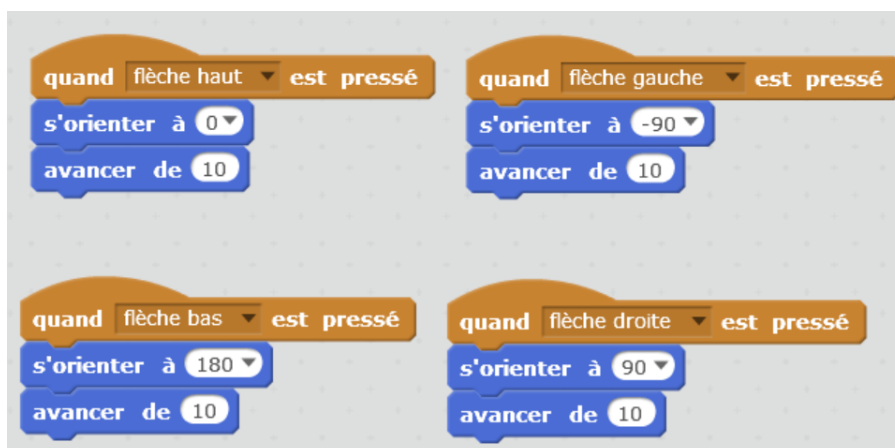
Tâche 3 : piloter le rover à l'aide des flèches (15 minutes)

Les élèves doivent maintenant faire en sorte que le rover se déplace quand ils pressent les flèches du clavier. Ils cherchent en autonomie comment faire. Certains se rappellent la commande « quand drapeau vert pressé » vue lors de la toute première séance Scratch. C'est un événement qui permet de déclencher une action. Ici aussi, on cherche un événement : l'action se déclenche quand une touche du clavier est pressée.

La commande « **quand (espace) est pressé** » nous intéresse, sauf qu'il faut changer « espace » par une des flèches (flèche droite pour aller à droite). Cela se fait de la même manière que précédemment :



Finalement, la zone du programme du rover comporte 4 sous-programmes, chacun décrivant le déplacement dans une direction particulière. Voici à quoi peut ressembler le programme :



Notes pédagogiques :

On remarque ici qu'on peut faire co-exister plusieurs sous-programmes dans le même programme. Chacun s'exécute lorsque l'événement déclencheur (ici, presser une touche du clavier) est détecté.

Certains élèves paniquent parfois en pensant que tout leur programme a disparu suite à une fausse manœuvre. En général, cela n'est pas le cas (le programme n'est pas effacé). Ils ont simplement cliqué sur la scène (qui a sa propre zone de programme... mais vide puisqu'on n'y a rien mis pour le moment) au lieu du lutin. Parfois, ils ont bien sélectionné le lutin, mais ont cliqué sur l'onglet « costumes » au lieu de l'onglet « scripts ». Il suffit de retourner sur le lutin, et sur l'onglet « scripts » pour voir réafficher le fameux programme !



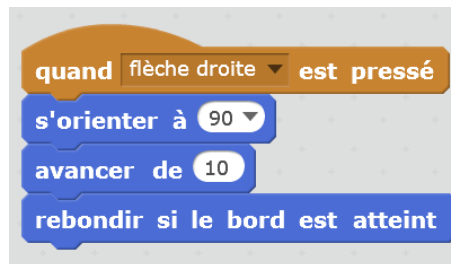
Classe de CM1 de Caroline Vinel (Paris)



Tâche 4 : rebondir sur les bords (5 minutes)

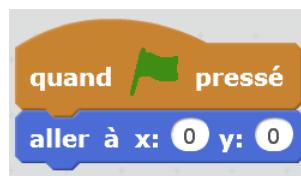
Les élèves cherchent comment faire en sorte que le rover rebondisse sur les bords. Par exemple, si on le dirige vers la droite et qu'il atteint l'extrémité droite de l'écran, le rover doit rebondir afin de ne pas sortir de l'écran.

Cela se fait très simplement en ajoutant l'instruction « **rebondir si le bord est atteint** » en bas de chacun des sous-programmes faits précédemment. Par exemple :



Tâche 5 : initialiser la position du rover (5 minutes)

L'enseignant rappelle que, lorsqu'on lance le programme (drapeau vert), le rover doit se situer au centre de l'écran. Les élèves reprennent sans difficulté les instructions qu'ils avaient vues lors de la première séance *Scratch*.



Note pédagogique :

Le programme peut maintenant être exécuté en cliquant sur le drapeau vert. Si on le souhaite, on peut faire disparaître les programmes pendant l'exécution, en cliquant sur le bouton « plein écran » en haut à gauche de la scène.

Ne pas oublier, à chaque fois, de **sauvegarder** le travail réalisé !



Tâche 6 : comprendre les coordonnées X et Y du rover (20 minutes)

La tâche précédente a mis en évidence les coordonnées X et Y du rover, à travers l'instruction « aller à (X = 0, Y = 0) ». Les étapes suivantes (ressources, pièges...) nécessiteront de manipuler ces coordonnées ; il importe donc de comprendre comment cela fonctionne.

L'enseignant demande aux élèves de visualiser les coordonnées X et Y qui s'affiche en bas à droite de la scène. Ceux-ci remarquent que les coordonnées affichées changent en fonction de la position de la souris.



Que valent X et Y quand la souris est au centre de la scène ? (réponse : X = 0, Y = 0)

Et quand la souris est tout à droite ? (réponse : X = 240, Y peut prendre n'importe quelle valeur selon la position de la souris)

Et quand la souris est tout à gauche ? (X = -240)

Et quand la souris est tout en haut ? (Y = 180) ou tout en bas ? (Y = -180)

La classe conclut collectivement que X indique la position selon l'axe horizontal (axe imaginaire, non tracé) et Y indique la position selon l'axe vertical (lui aussi imaginaire). Les élèves peuvent remarquer que, dans la catégorie « mouvement » de la palette d'instructions, de nombreuses instructions font intervenir les variables X et Y. Ici, il ne s'agit plus de la position de la souris, mais de celle du lutin sélectionné. Le rover possède son propre jeu de variables X et Y.

L'enseignant peut distribuer la [Fiche 33](#) à chaque élève et lui proposer des petits exercices :

Place le lutin sur la scène, aux coordonnées X = -100, Y = 100

Que se passe-t-il si on ajoute 50 à X ? Où est désormais le lutin ?

Que se passe-t-il si maintenant on met Y à 0 ? Où est le lutin ?

Notes pédagogiques :

Pour faciliter la compréhension de ces coordonnées, l'enseignant peut faire un parallèle avec ce que les élèves ont déjà vu en géographie : latitude / longitude. Ici, l'unité n'est pas le degré (on ne raisonne pas en angle), mais le pixel. De même, dans un jeu de bataille navale, on repère la position des

bateaux par 2 coordonnées (une lettre et un chiffre). Au choix de l'unité ou du symbole près, il s'agit exactement de la même chose : repérer la position d'un point sur une surface, ce qui nécessite 2 coordonnées car une surface est un espace à 2 dimensions.

De même, il peut être utile à certains élèves de se raccrocher à des situations concrètes au sujet des nombres négatifs. Les exemples ne manquent pas, qu'il s'agisse des dates ou de la mesure de la température (que signifie « -10°C » : est-ce plus chaud ou plus froid que « 0°C ». Et « -20°C », est-ce plus chaud ou plus froid que « -10°C » ?).

Conclusion et traces écrites

À la fin de cette séance, il importe de faire le point sur les nouvelles commandes *Scratch* que les élèves ont appris à utiliser :

S'orienter à (90)

Quand (espace) est pressé

Quand (drapeau vert) est pressé

Aller à (X = ..., Y = ...)

Les élèves colorient ces commandes sur la [Fiche 32](#) qu'ils avaient déjà utilisée.

Par ailleurs, cette séance offre l'occasion de prendre du recul par rapport aux activités de programmation et de revenir sur certains concepts :

Un programme est un algorithme exprimé dans un langage particulier, appelé langage de programmation, compréhensible à la fois par la machine et par l'être humain.

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).

L'ordinateur ne fait qu'exécuter les instructions qu'on lui donne : ni plus, ni moins.

La position d'un élément à l'écran est repérée grâce à 2 coordonnées. Dans Scratch, on les appelle X et Y. X varie entre -240 et 240 ; Y varie entre -180 et 180.

Les élèves notent ces conclusions dans leur cahier de sciences. L'enseignant, quant-à-lui, met à jour l'affiche « qu'est-ce que l'informatique ? ».

<< Etape III-2.2

Séquence III-2

Etape III-2.4 >>

Extrait de "1, 2, 3... codez !", Editions Le Pommier, 2016. Publié sous licence CC by-nc-nd 3.0.

PARTENAIRES DU PROJET

