

1, 2, 3, codez ! - Activités cycle 3 - Etape 2.6: Eviter des obstacles, gérer son nombre de vies

Résumé	Les élèves complètent leur programme en introduisant des obstacles à éviter (nouveaux lutins) et en créant une variable pour leur nombre de « vies ». Ils réinvestissent les notions de test, de boucle et de variable abordées précédemment, et approfondissent la notion d'évènement.
Notions	<p>« Machines » :</p> <ul style="list-style-type: none"> 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>« Algorithmes »</p> <ul style="list-style-type: none"> 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>« Langages » :</p> <ul style="list-style-type: none"> 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 Scratch 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)
Matériel	<p>Pour chaque binôme</p> <ul style="list-style-type: none"> Un ordinateur avec Scratch et le programme enregistré à la séance précédente Les nouveaux lutins qui seront utilisés à cette séance, téléchargeables ici.

Note pédagogique :

Il peut être utile, de temps en temps, de montrer le jeu « final » (version de l'enseignant) afin de relancer la motivation et de faire le point sur les étapes à venir. Il ne s'agit pas de leur donner la solution en leur faisant lire le programme, mais simplement de faire une démonstration du jeu.

Pour pimenter le jeu simulant la mission spatiale, l'enseignant explique qu'il va falloir introduire des obstacles (un lac de lave, une dune de sable), et une nouvelle variable : le nombre de « vies ». Le rover démarre le jeu avec 3 vies et perd une vie, en revenant à la base, chaque fois qu'il touche un obstacle. Le jeu s'arrête lorsque le nombre de vies atteint 0.

Tâche 1 : ajout de nouveaux lutins (5 minutes)

Se référant à ce qu'ils ont appris lors des précédentes séances, les élèves importent sans difficulté les 3 nouveaux lutins (la base, la dune et la lave) et les initialisent en fixant leur position en un endroit précis de l'écran. Ils peuvent décider, par exemple, de placer la base au centre, ce qui nécessite de décaler la position initiale du rover (puisqu'on l'avait placé au centre).

Cela n'est pas indispensable, mais on peut vouloir préciser la « profondeur » des différents lutins (rover, base, lave, dune, végétation, glace). Décider quel lutin sera au premier plan se fait via l'instruction « envoyer au premier plan » disponible dans la catégorie « apparence » des instructions.

envoyer au premier plan

Tâche 2 : création et initialisation d'une variable « nombre de vies » (5 minutes)

De même, ils créent facilement une nouvelle variable « vies » qu'ils initialisent à la valeur 3 (au même endroit que l'initialisation du score, par exemple dans le programme du rover). Le programme du rover contient donc, désormais (en plus des sous-programmes permettant de le diriger) :



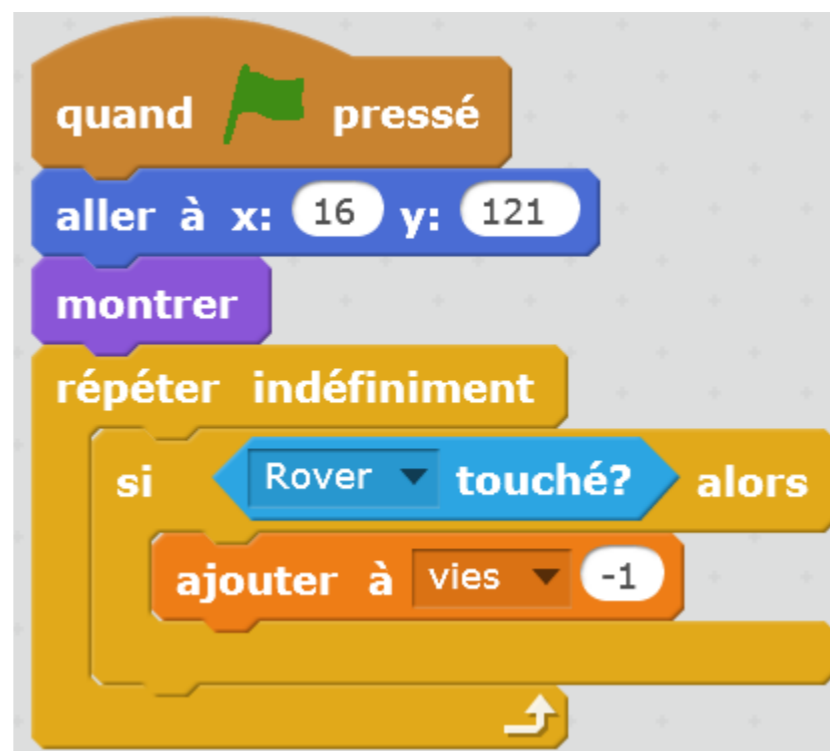
Tâche 3 : perdre une vie quand le rover touche la lave (30 minutes)

À la séance précédente, toucher une ressource permettait d'augmenter de 1 la valeur de la variable « score ». De même, ici, toucher la lave ou la dune doit faire diminuer de 1 la valeur de la variable « vie ». Nous conseillons de résoudre d'abord la tâche pour un des obstacles (la lave), puis de recommencer pour l'autre (la dune).

Comment soustraire, dans Scratch ?

Les élèves cherchent comment contrôler / programmer cette diminution. Il n'y a pas de commande « soustraire », mais seulement « ajouter ». Au besoin, discuter collectivement du fait qu'il faut ajouter la valeur -1 pour soustraire 1.

Pour la lave, le programme est :



Programme provisoire pour la lave

Ce programme a cependant un défaut : lorsque le rover entre en contact avec un obstacle, il le touche pendant un certain temps (quelques dixièmes de secondes, quelques secondes si on ne bouge plus) : pendant tout ce temps, la variable « vie » décroît. Au bout de quelques secondes, on se retrouve avec des valeurs négatives très grandes (-4000...). La seule façon d'éviter cela est de rompre immédiatement le contact entre le rover et l'obstacle.

Puisque l'obstacle est fixe, c'est donc le rover qui doit se déplacer. Or, cela ne peut pas être fait dans le programme de la lave ou de la dune : il faut être dans le programme du rover pour commander la position du rover.

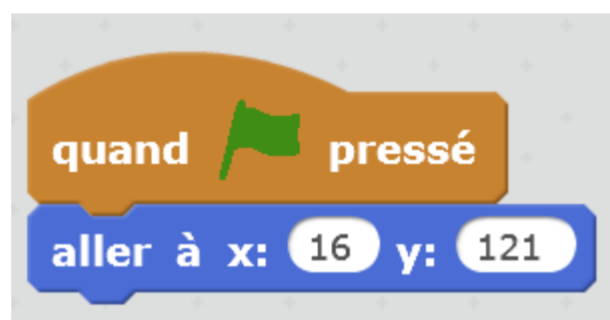
Comment déplacer le rover ?

Il y a (au moins) deux solutions possibles à ce problème :

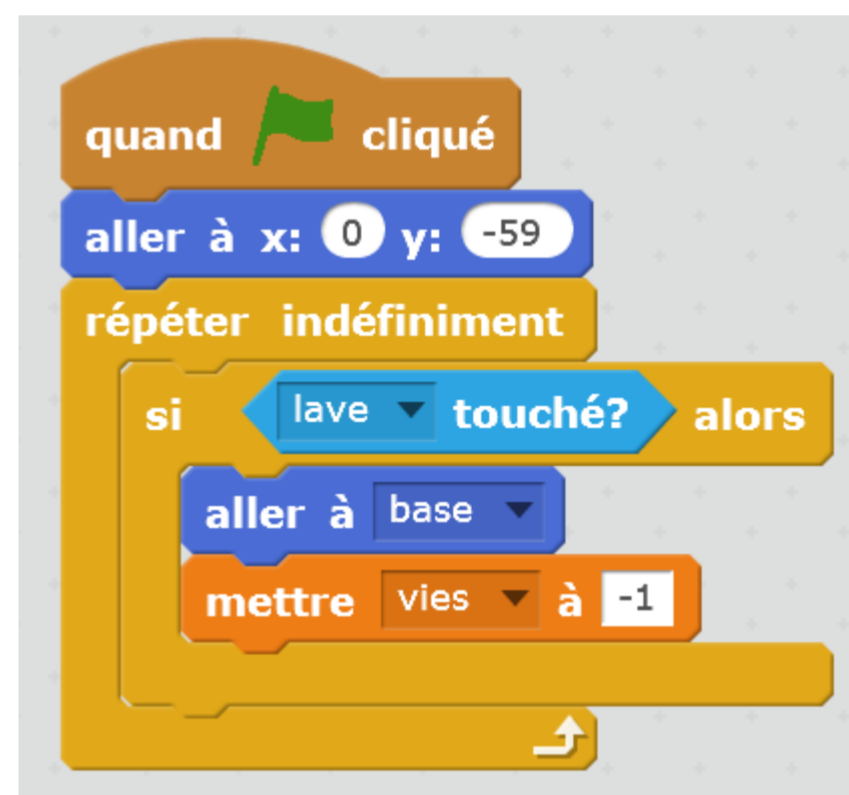
- **Solution 1**

Supprimer le programme que l'on a fait pour la lave (sauf l'initialisation de sa position) et en faire un similaire dans le programme du rover. Dans le programme du rover, on ajoute une commande disant, par exemple, de rentrer à la base.

Notons, en passant, que pour rentrer à la base, on peut soit dire au rover d'aller en (X=0, Y=0), soit dire au rover d'aller à la position du lutin « base ». Cette dernière solution est meilleure car elle marchera même si on décide de placer la base ailleurs.



Programme de la lave

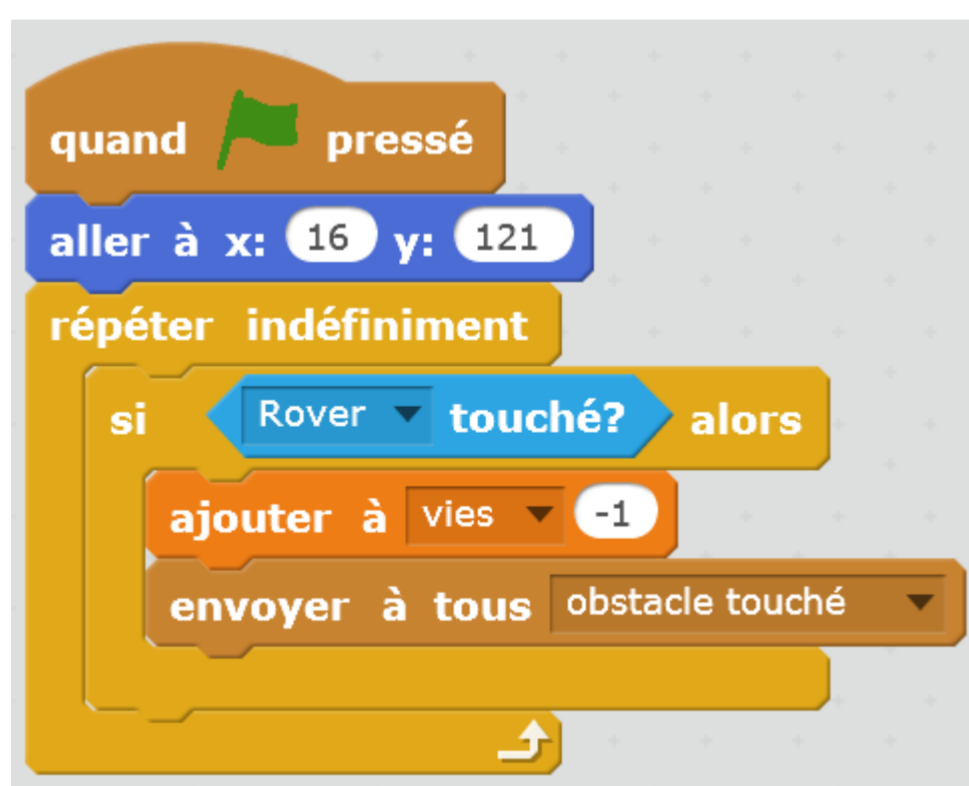


Programme du rover

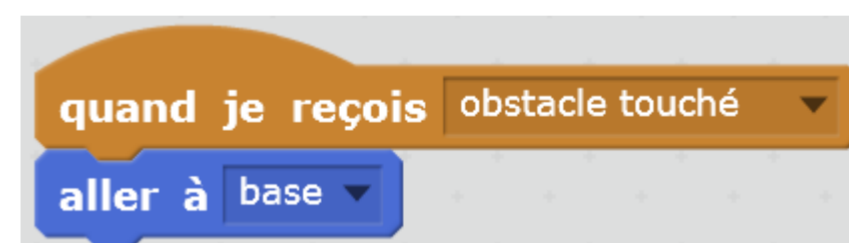
Solution 1

- **Solution 2 (plus élégante... et plus pratique pour la suite)**

Une solution plus élaborée consiste à garder le programme que l'on a fait pour la lave, en y ajoutant une nouvelle instruction qui va envoyer un message aux autres programmes (celui du rover en particulier). Ce message, dans le programme du rover, déclenchera une action (retourner à la base). Tout comme les noms de variables, les intitulés des messages doivent être explicites. Ici, notre message est par exemple « obstacle touché ».



Programme de la lave



Programme du rover

Solution 2

Le programme de la lave envoie un message aux autres programmes (celui du rover en particulier).

Dans le programme du rover, la réception du message est un événement qui déclenche une action (aller à la base).

Les commandes permettant d'envoyer un message ou de déclencher une action lorsqu'un message est reçu se situent dans la catégorie « événements ».

Notes pédagogiques :

- On peut aussi imaginer une autre solution, par exemple programmer le rover pour qu'il rebondisse s'il touche un obstacle (ainsi, le contact ne dure pas).
- La seconde solution peut être considérée comme plus élégante si l'on souhaite insister sur le caractère « événementiel » de ce programme. Chaque fois qu'un événement se produit (par exemple, toucher un obstacle), le programme peut envoyer un message qui sera utilisé par d'autres programmes. La notion d'événement a déjà été introduite lors des séances précédentes (« quand la flèche haute est touchée », « quand le drapeau vert est touché »...), mais c'est la première fois que l'on utilise ce nouveau type d'événement : la réception d'un message diffusé par un programme.
- Nous laissons le choix à l'enseignant d'utiliser l'une ou l'autre méthode selon qu'il préfère consacrer cette séance à des révisions de notions déjà vues ou s'il souhaite introduire ici la notion, nouvelle, de message. Bien sûr, il a tout intérêt à faire ce choix en tenant compte des idées proposées par les élèves et de leur maîtrise des notions précédentes !
- À noter : l'envoi et la réception de message seront (à nouveau) utilisés, plus tard, pour gérer la fin du jeu (voir « game over »).



Tâche 4 : refaire la même chose avec la dune (10 minutes)

Maintenant que les élèves ont réussi à gérer un des obstacles (la lave), ils doivent refaire la même chose pour l'autre (la dune). Ce petit exercice permet de mieux s'appropriier ce qui a été fait précédemment, en particulier l'envoi et la réception de message.

Conclusion et trace écrite

Les élèves mettent à jour la liste des instructions *Scratch* qu'ils connaissent.

<< [Etape III-2.5](#)

[Séquence III-2](#)

[Etape III-2.7](#) >>

Source URL: <https://www.fondation-lamap.org/fr/page/34524/1-2-3-codez-activites-cycle-3-etape-26-eviter-des-obstacles-gerer-son-nombre-de-vies>