1, 2, 3, codez ! - Activités cycle 3 - Etape 2.7: Game Over

Résumé	Les élèves complètent leur programme en introduisant un test sur le nombre de vies restantes : le message « Game over » apparait et le programme s'arrête quand toutes les vies sont épuisées.
Notions	« Machines » : • 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 « Algorithmes » • 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. « Langages » : • 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 évènementielle. • 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	Pour chaque binôme • Un ordinateur avec *Scratch* et le programme enregistré à la séance précédente

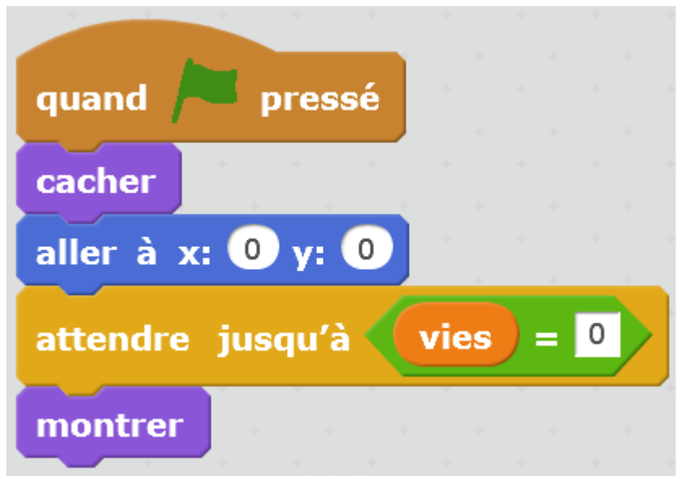
Les élèves doivent maintenant faire en sorte que le jeu s'arrête lorsque le nombre de vies restantes vaut zéro. Cela suppose plusieurs choses :

- Faire apparaître « game over »
- Faire en sorte qu'on ne puisse plus jouer (sauf en relançant le programme)

Tâche 1 : faire apparaître « game over » quand le nombre de vies vaut 0 (15 minutes)

Faire apparaître « game over » peut se faire de plusieurs façons. La plus simple consiste à importer un lutin dont l'apparence est « game over » (lutin fourni). Ce lutin doit apparaître lorsque le nombre de vies vaut zéro.
Cela se fait très facilement, à l'aide de l'instruction «**montrer** » dans la catégorie « apparence ». Note : il ne faut pas oublier, au lancement du programme, de cacher le lutin « game over » !

Le programme de ce lutin « game over » est :



Programme du lutin « game over »

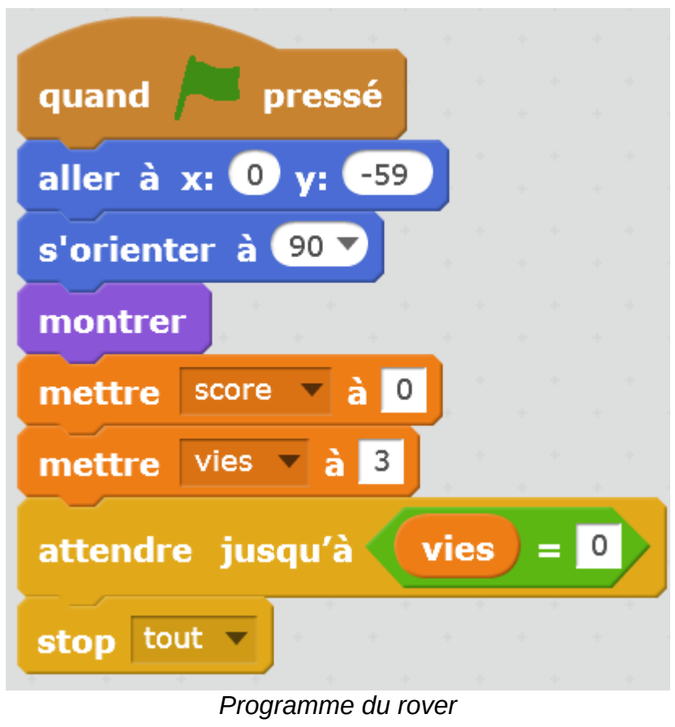
Note pédagogique :
Il est aussi possible d'importer un lutin « texte » depuis la bibliothèque « Scratch ». Ce lutin s'appelle « awesome! » : en cliquant sur l'onglet « costume », on peut changer la couleur, la police de caractère… et éditer le texte.

Tâche 2 : arrêter le jeu quand apparaît « game over » (15 minutes)

En l'état actuel, on peut continuer à jouer lorsque « game over » s'affiche, ce qui n'est pas satisfaisant. Pour mettre fin au jeu, il y a plusieurs stratégies possibles :

Méthode 1

Déclencher l'arrêt de tous les programmes quand le nombre de vies vaut zéro. Cela se fait via la commande «**Stop tout** » de la catégorie «**contrôle** ».



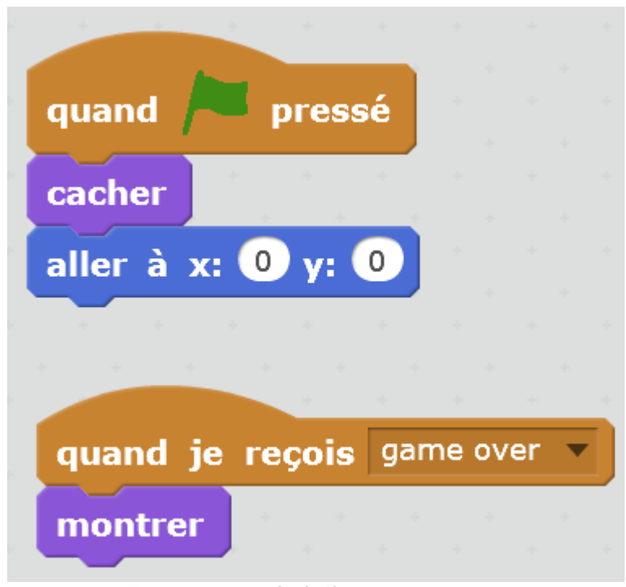
Programme du rover

En théorie, cela suffit à stopper le jeu. Malheureusement, en pratique, un bug de *Scratch* (présent à l'heure où nous écrivons ces lignes) fait que, malgré l'instruction « stop tout », certains programmes s'arrêtent et d'autres pas (il est encore possible de bouger le rover). Pour cette raison (et aussi car elle est visuellement plus satisfaisante), on préfère la seconde méthode, ci-après.

Méthode 2

Faire en sorte que tous les autres lutins disparaissent lorsque le nombre de vies vaut zéro. Ainsi, le seul lutin encore à l'écran est « game over », et le jeu est bel et bien terminé. Cette méthode suppose qu'un des programmes (par exemple celui du rover) **envoie un message**. Ce message s'intitule tout simplement « game over ».
Tous les lutins (à l'exception du lutin « game over », bien sûr), se cachent lorsqu'ils reçoivent ce message. Puisqu'on leur demande de se cacher à la fin… il faut penser à leur demander de se montrer au lancement du programme !

Programme du rover.

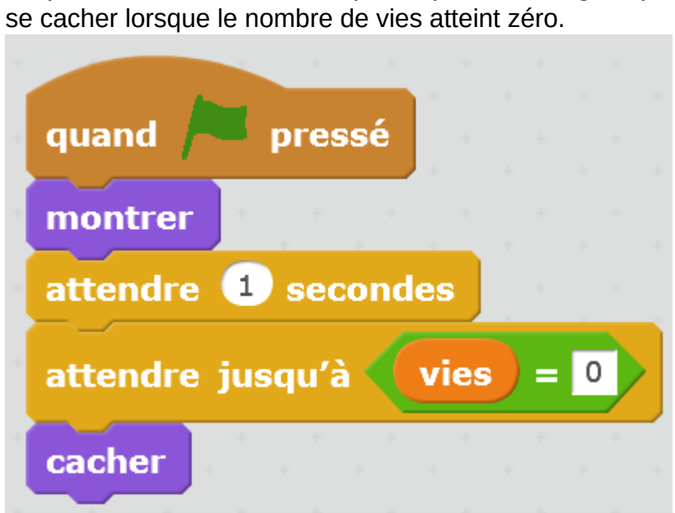


Programme du lutin « game over »

La dernière instruction du programme du rover (se cacher quand on reçoit « game over ») est également présente dans les programmes des autres lutins (sauf le lutin « game over » qui, lui, se montre à ce moment-là).

Notes pédagogiques

- On peut faire une variante simplifiée (et moins élégante) de la méthode 2 si l'on ne souhaite pas utiliser l'envoi ou la réception de messages. Cette variante consiste demander à chaque lutin de se cacher lorsque le nombre de vies atteint zéro.



  La pause d'1 seconde permet de s'assurer que la variable « vies » a eu le temps, dans le programme du rover, d'être initialisée à 3. Sinon, comme elle vaut 0 au lancement du programme, les lutins se cachent immédiatement.
- Introduire une pause, même d'une fraction de seconde, peut permettre de résoudre des bugs assez subtils liés à des problèmes de synchronisation entre les différents programmes. *Scratch* donne l'illusion qu'il exécute tous les programmes en parallèle, mais en réalité il les exécute les uns après les autres (très rapidement… d'où l'illusion de la simultanéité). Introduire une pause ou envoyer/recevoir un message est une manière de forcer l'exécution d'un programme avant un autre.

Conclusion et trace écrite

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

<< Etape III-2.6 Séquence III-2 Etape III-2.8 >>