

## 1, 2, 3, codez ! - Activités cycle 3 - Séquence 2 : Simuler la mission dans Scratch

Cette séquence est consacrée à la **programmation** d'un jeu vidéo qui va simuler notre mission d'exploration spatiale, dans la continuité de ce qui a été fait lors de la [séquence III-1](#).

Cette séquence 2 se passe essentiellement devant l'ordinateur. Néanmoins, on y trouve quelques **séances débranchées** (sans ordinateur) destinées à mieux comprendre certains concepts, comme les variables ou les opérateurs logiques. Ces séances débranchées, idéalement, doivent être mises en œuvre sur un autre temps que le temps dédié à la programmation (en séance de mathématiques ou français, selon les cas), afin de ne pas interrompre le projet.

### Avant de se lancer dans une activité de programmation : quelques conseils

Programmer un jeu vidéo est extrêmement motivant pour les élèves... mais pour que cela se passe bien, il faut prendre quelques précautions.

#### Scratch, un environnement idéal pour apprendre à programmer

Il existe de très nombreux outils d'apprentissage de la programmation. Nous avons fait le choix d'utiliser le logiciel [Scratch](#), en raison de sa qualité remarquable, sa simplicité d'utilisation, sa gratuité, et sa communauté très active, y compris dans le champ de l'éducation (primaire et collège).

Ce choix est cependant arbitraire et, bien entendu, l'enseignant peut suivre la même progression en utilisant d'autres environnements (comme *snape*, très similaire à *Scratch*, et lui aussi gratuit, ou *kodu* et *tangara*, qui sont payants...).

Nous proposons également une alternative à Scratch « seul » dans le cas où la classe possède un robot, dont la programmation reprend les mêmes concepts, mais appliqués à un objet physique. Le robot Thymio, par exemple, se programme à l'aide du langage VPL et/ou Scratch (voir, à ce sujet, la séquence 3 de la progression pour le cycle 2, pages 178 et suivantes).

#### S'assurer de la maîtrise de quelques compétences TICE de base

Programmer nécessite d'interagir avec un ordinateur, ce qui suppose de maîtriser quelques compétences élémentaires :

- Utiliser le **clavier et la souris** (c'est très souvent le cas en cycle 3, mais pas systématique) ;
- 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.

Si certains élèves ne maîtrisent pas ces compétences de base, ils pourront les acquérir à travers ce projet de programmation, mais ils risquent de perdre beaucoup de temps lors des premières séances.

#### Travailler en demi-groupe

L'idéal est de disposer d'un **ordinateur pour 2 élèves** (3 maximum). Afin de permettre une telle répartition, et aussi afin de faciliter la gestion de classe pendant les activités de programmation (au cours desquelles l'enseignant est très sollicité), nous conseillons de travailler en demi-groupe : la moitié de la classe programme pendant que l'autre moitié travaille à autre chose, en autonomie.

#### Préparer l'environnement de travail

Afin de gagner du temps, il est utile d'avoir préparé l'environnement de travail :

- *Scratch* doit être [installé](#) sur toutes les machines (ou accessible [en ligne](#)) ;
- Un raccourci vers *Scratch* doit être présent sur le bureau ;
- De même, un répertoire dédié au projet (et à la classe) doit être facilement accessible, depuis le bureau ou sur une clé USB dédiée au groupe. Ce répertoire contiendra les fichiers nécessaires au projet (images à importer, sauvegarde des programmes...). Nous mettons sur ce site Web dédié au projet, tous les fichiers utiles, téléchargeables au fil des séances. Les utilisateurs les plus avancés pourront mettre les fichiers utiles directement dans les sous-répertoires du logiciel *Scratch* (par exemple : <Scratch>/Medias/Costumes/MissionMars et <Scratch>/Medias/Background/MissionMars, où <Scratch> désigne le répertoire d'installation du logiciel)

#### Faire soi-même le projet avant !

C'est sans doute le conseil le plus important, bien qu'il paraisse aller de soi. Il est primordial que l'enseignant ait pris 2 à 3 heures sur son temps personnel AVANT la première séance de classe, pour se familiariser avec *Scratch* et pour réaliser les tâches que les élèves auront à réaliser au fil du projet.

Sinon, il risque fort de ne pas être capable d'aider les élèves quand ceux-ci en auront besoin.

Ce n'est pas difficile (il suffit de suivre le déroulement proposé dans cette séquence), et c'est même assez amusant ! Le tableau récapitulatif ci-après liste les différentes étapes et, pour chaque étape, les différentes tâches à réaliser pour construire un jeu vidéo. Si l'enseignant choisit un autre scénario pour le jeu, il faudra bien entendu adapter cette progression.

Note : vous pouvez **tester le jeu** en ligne (ou en faire la démonstration aux élèves), en cliquant sur l'image ci-dessous. Récoltez de l'eau et de la nourriture (végétaux) tout en évitant les pièges !



À noter : dans la plupart des classes de cycle 3, réaliser l'ensemble du projet **nécessitera 6 ou 7 séances d'1 heure**. Certains binômes auront produit un jeu plus complet et complexe que d'autres, mais tout le monde aura produit un jeu satisfaisant et valorisant.

Nous conseillons, au moins au début du projet, de faire **2 séances par semaine**, afin d'éviter que les élèves aient oublié, d'une séance à l'autre, ce qu'ils ont appris, car la programmation est une activité véritablement nouvelle pour eux.















































Sauf indication contraire, toutes les étapes sont branchées ; les durées indiquées sont des durées moyennes.



#### Des repères de difficulté

Pour se repérer dans ces différentes étapes et tâches, nous utiliserons un symbole inspiré des pistes de ski :

	<b>Piste verte : facile</b> <i>Tous les élèves y arrivent sans problème.</i>
	<b>Piste bleue : moyenne</b> <i>La plupart des élèves y arrivent seuls, certains ont besoin d'être un peu guidés.</i>
	<b>Piste rouge : difficile</b> <i>La plupart des élèves ont besoin d'être guidés (plus ou moins selon leur aisance).</i>
	<b>Piste noire : très difficile</b> <i>Tous les élèves ont besoin d'être guidés. Ces tâches sont facultatives.</i>

# Listing des tâches

	Etape	Titre	Tâche	
	Etape 1	Découverte de l'environnement de programmation Scratch	Tâche 0 : démonstration, par l'enseignant, du jeu final (5 minutes)	
			Tâche 1 : lancer Scratch et découvrir son interface (10 minutes)	
			Tâche 2 : explorer librement Scratch (15 minutes)	
			Tâche 3 : petits exercices (20 minutes)	
	Etape 2	Planter le décor, et sauvegarder son travail	Tâche 1 : changer le lutin (5 minutes)	
			Tâche 2 : changer l'arrière-plan (5 minutes)	
			Tâche 3 : enregistrer son programme Scratch (5 minutes)	
	Etape 3	Piloter le rover	Tâche 1 : faire avancer le rover vers la gauche (10 minutes)	
			Tâche 2 : faire avancer le rover dans n'importe quelle direction (5 minutes)	
			Tâche 3 : piloter le rover à l'aide des flèches (15 minutes)	
			Tâche 4 : rebondir sur les bords (5 minutes)	
			Tâche 5 : initialiser la position du rover (5 minutes)	
			Tâche 6 : comprendre les coordonnées X et Y du rover (20 minutes)	
	Etape 4	Récolter des ressources, gérer son score	Tâche 1 : importer une ressource (la glace) sous la forme d'un nouveau lutin	
			Tâche 2 : faire dire « bravo » à la ressource lorsqu'elle est touchée par le rover (20 minutes)	
			Tâche 3 : faire disparaître la ressource quand elle est touchée (10 minutes)	
			Tâche 4 : créer une variable « score » (5 minutes)	
			Tâche 5 : augmenter le score lorsqu'on récolte une ressource (10 minutes)	
			Tâche 6 : initialiser le score à zéro (10 minutes)	
			Tâche 7 : faire réapparaître une ressource à une position aléatoire (15 minutes)	
			Tâche 8 : importer une nouvelle ressource (la végétation) et refaire le même travail que pour la glace (20 minutes)	
 	Etape 5	Activités branchées et débranchées pour mieux s'appropriier certains concepts algorithmiques	Activité 1 : évaluation formative sur le concept de boucle (branchée, 10 à 20 minutes)	
			Activité 2 : un jeu de cartes pour s'appropriier la notion de variable (débranchée, 1 heure)	
			Activité 3 : un jeu de cartes pour travailler les opérateurs logiques (débranchée, 1 heure)	
			Activité 4 : comprendre qu'un algorithme n'est pas toujours parfait : le jeu du voyageur de commerce (débranchée, 1 heure)	
	Etape 6	Éviter des obstacles, gérer son nombre de vies	Tâche 1 : ajout de nouveaux lutins (5 minutes)	
			Tâche 2 : création et initialisation d'une variable « nombre de vies » (5 minutes)	
			Tâche 3 : perdre une vie quand le rover touche la lave (30 minutes)	
			Tâche 4 : refaire la même chose avec la dune (10 minutes)	
	Etape 7	Mettre fin au jeu : « game over »	Tâche 1 : faire apparaître « game over » quand le nombre de vies vaut 0 (15 minutes)	
			Tâche 2 : arrêter le jeu quand apparaît « game over » (15 minutes)	
	Etape 8	Pimenter un peu le jeu	Tâche 1 : faire apparaître un compte à rebours au lancement du jeu (15 minutes)	
			Tâche 2 : limiter la durée du jeu (15 minutes)	
			Tâche 3 : ajouter une tornade qui se déplace aléatoirement (15 minutes)	
			Tâche 4 : faire grossir la tornade (15 minutes)	
			Tâche 5 : faire accélérer la tornade au fur et à mesure (20 minutes)	
			Tâche 6 : simuler un monde torique (joindre les côtés de la scène) (20 minutes)	

<a href="#">Tâche 7</a> : éviter que les ressources et les pièges ne se superposent (20 minutes)			
	<a href="#">Etape 9</a>	Prolongements possibles en <i>Scratch</i>	À ce stade, le projet est terminé. Nous proposons ici quelques pistes pour explorer d'autres fonctionnalités de <i>Scratch</i> , par exemple pour alimenter de futurs projets personnels d'élèves.

---

<< [Séquence III-1](#)

[Activités cycle 3](#)

[Séquence III-3](#) >>

---

Source URL: <https://www.fondation-lamap.org/fr/page/34518/1-2-3-codez-activites-cycle-3-sequence-2-simuler-la-mission-dans-scratch>