

1, 2, 3, codez ! - Activités cycle 3 - Etape 2.1: Découverte de Scratch

Résumé	Les élèves découvrent <i>Scratch</i> , un environnement de programmation adapté à l'école primaire. Ils apprennent à lancer le programme et enchaîner quelques instructions simples.
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 <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 événement : 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 la classe</p> <ul style="list-style-type: none"> Un vidéo projecteur Version agrandie de la Fiche 32 <p>Pour chaque binôme</p> <ul style="list-style-type: none"> 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>Pour chaque élève</p> <ul style="list-style-type: none"> Fiche 32
Lexique	Programme, script, lutin, instruction, événement
Durée :	1 heure

Rappel :

Il est absolument indispensable que l'enseignant fasse lui-même le projet avant de le proposer aux élèves ! Il suffit de suivre les étapes décrites dans la séquence. Un enseignant débutant en *Scratch* prendra environ 3 heures pour faire l'intégralité du projet.

Notes pédagogiques

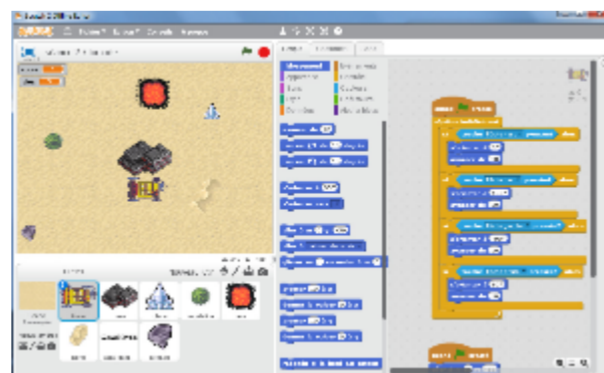
- Apprendre à programmer se fait en programmant, pas en regardant quelqu'un programmer ! Il est intéressant de réfléchir à 2 sur un même problème (probablement plus que de programmer seul), mais il est important d'être actif. Nous conseillons donc de mettre les élèves par petits groupes devant les machines (idéalement, 2 élèves par machine) et de leur demander de « passer la main » (donner le clavier et la souris à son voisin) toutes les 10 ou 15 minutes.
- Nous conseillons, si possible, d'organiser la classe en demi-groupes de façon à ne pas avoir trop de binômes à gérer. Pendant que la moitié de la classe travaille sur *Scratch*, l'autre moitié fait autre chose, en autonomie.
- Si possible, faire 2 séances de *Scratch* par semaine, surtout au début du projet.
- Cette étape de découverte de *Scratch* est, volontairement, très directive (les tâches 0 et 1 sont même une démonstration de l'enseignant !). C'est la seule qui se présente sous cette forme. Tous les binômes vont devoir accomplir une série de tâches élémentaires. À la fin de chaque tâche, une mise en commun permet de s'assurer que chacun a compris et sait faire. Les autres étapes seront moins dirigées, les élèves devenant plus autonomes et avançant chacun à leur rythme.
- Pour gagner du temps, allumer les ordinateurs avant le début de la séance.

Tâche 0 : démonstration, par l'enseignant, du jeu final (5 minutes)

L'enseignant explique que l'on va simuler la mission d'exploration (à défaut de pouvoir la vivre) à travers un jeu vidéo que l'on va programmer soi-même. Depuis son poste informatique, il lance *Scratch* et montre le jeu vidéo « final » (celui qu'il a lui-même réalisé à l'avance) sans expliquer comment fonctionne le programme, mais en faisant simplement une démonstration du jeu.

Notes pédagogiques :

- Cette démonstration est très importante car elle motive énormément les élèves et les rassure : ils vont vraiment programmer un « vrai » jeu vidéo ! Elle permet par ailleurs de faire le lien entre cette activité de programmation et les activités débranchées de la [séquence 1](#).
Voici notre rover, que l'on va apprendre à diriger. Il va récolter des ressources pour permettre aux humains de vivre à la base : de l'eau, de la nourriture... (pour chaque ressource récoltée, le score augmente). Mais attention aux pièges ! S'il tombe dans un piège, il perd une vie. S'il n'a plus de vie, la partie est terminée.
- Il est important d'expliquer aux élèves que programmer un tel jeu ne se fait pas en une séance mais nécessitera plusieurs séances (typiquement, 6 ou 7 séances en fonction de leur niveau et de leurs exigences).



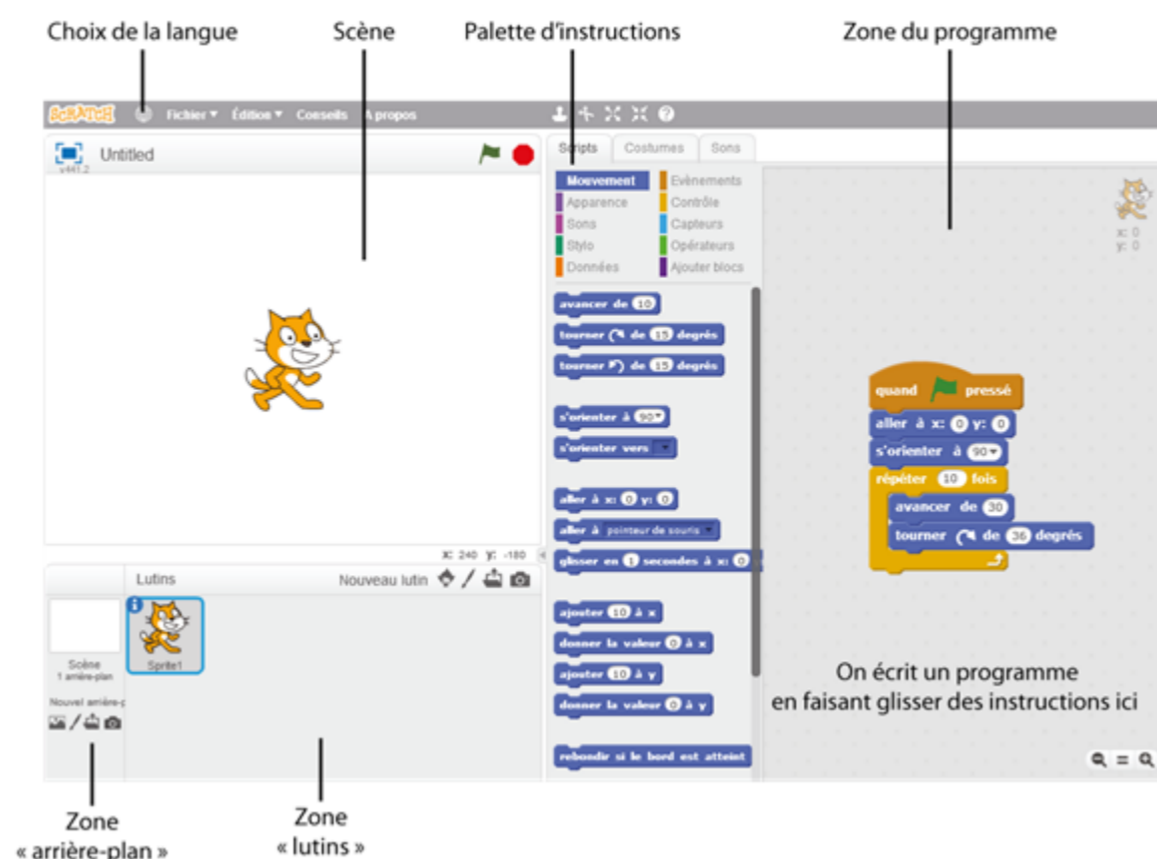
Capture d'écran : version finale du jeu vidéo réalisé par l'enseignant

(NB : un programme corrigé est disponible [ici](#), mais encore une fois, il est important que l'enseignant tente lui-même de le réaliser !)

Tâche 1 : lancer Scratch et découvrir son interface (10 minutes)

Chaque binôme lance *Scratch* en double-cliquant sur son icône.

Si c'est la première fois que *Scratch* est utilisé sur cette machine, il se peut que le logiciel soit en anglais. Dans ce cas, les élèves peuvent très facilement le passer en français en cliquant sur l'icône représentant un globe terrestre (en haut à gauche, à côté du logo «Scratch»).



L'enseignant explique aux élèves que *Scratch* est un langage de programmation conçu spécifiquement pour apprendre à programmer. Lorsque l'on ouvre le logiciel, il y a un lutin à l'écran (un chat). On peut lui donner des instructions simples.

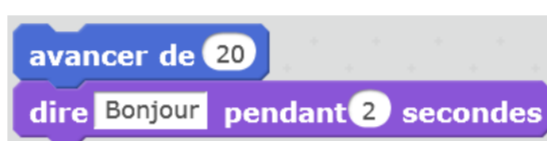
Il réalise une petite **démonstration** (les élèves devront, en fin de séance, refaire ces petits exercices).

Par exemple, pour demander au chat de **se déplacer de 10 pas**, il suffit de faire glisser l'instruction « avancer de 10 » depuis la palette d'instructions vers la zone du programme. Si l'on clique ensuite sur cette inscription, on remarque que le chat avance bien de 10 pas (1 pas = 1 pixel de l'écran).



Si l'on souhaite avancer de 20 pas, il suffit de changer « 10 » en « 20 » en cliquant dans la zone dédiée.

Si maintenant on souhaite que le chat avance de 20 pas, puis dise « Bonjour », il suffit de coller la nouvelle instruction à la fin du programme. L'instruction « dire Bonjour » n'existe pas, mais il y a une instruction « dire Hello » dans la catégorie « apparence » de la palette d'instructions. Il suffit de prendre cette instruction puis de remplacer le texte « Hello » en « Bonjour » en cliquant sur ce texte. Écrire un programme se fait simplement en emboîtant des instructions entre elles.



Si maintenant on veut que le chat fasse cela à chaque fois que l'on clique sur le drapeau vert (en haut à droite de la scène, le drapeau vert permet de lancer le programme), alors il faut rajouter l'instruction « Quand drapeau vert pressé » à chercher dans la catégorie « événements » des instructions. Cela donne :



Finalement, l'enseignant montre comment **supprimer une instruction** (ou tout un bloc d'instructions) : il suffit de faire glisser cette instruction (ou ce bloc) depuis la zone du programme vers la palette des instructions.

L'enseignant présente très rapidement l'interface de *Scratch*, qui comprend :

- Une **scène** : c'est là que se déroule le « jeu » (ou, plus généralement, le programme... on peut faire autre chose que des jeux dans *Scratch* !).
- Une **zone « lutins »** les lutins sont les personnages ou les objets qui seront manipulés dans le programme (ils peuvent se déplacer, changer de forme, parler, interagir avec les autres lutins...). Lorsqu'on lance *Scratch*, il n'y a qu'un seul lutin d'affiché à l'écran : un chat (plus tard, on ajoutera d'autres lutins et on supprimera le chat).
- Une **zone « arrière-plan »** juste à côté des lutins. L'arrière-plan est fixe, contrairement aux lutins qui peuvent bouger. Par défaut, dans *Scratch*, l'arrière-plan est un fond blanc uni (plus tard, on le modifiera).
- Un **onglet « script »** qui permet d'accéder à :
 - Une **palette d'instructions** (colonne centrale, à droite de la scène). C'est ici que l'on va trouver les instructions (ou « blocs ») que l'on va pouvoir utiliser dans notre programme. Il y a de nombreuses instructions, qui sont regroupées par couleur (exemple : tout ce qui concerne le mouvement du lutin est dans un onglet bleu foncé, tout ce qui concerne son apparence est dans un onglet violet, etc.).
 - Une **zone « programme »**, à droite de la palette d'instructions. C'est ici que l'on va écrire le programme, tout simplement en prenant des instructions depuis la palette et en les faisant glisser dans cette zone.
- Les autres onglets (costumes, sons) sont inutiles pour le moment.

Tâche 2 : explorer librement *Scratch* (15 minutes)

Les élèves disposent de 15 minutes pour explorer librement *Scratch*.

Pour le moment, ils ne doivent pas chercher à modifier la scène ou le lutin (ce sera fait lors de l'**étape suivante**) mais simplement manipuler des instructions simples et les agencer pour observer ce qui se passe. L'enseignant les encourage à explorer les différentes catégories d'instructions, en particulier :

- Catégorie « mouvement » (bleu foncé)
- Catégorie « apparence » (violet)
- Catégorie « événement » (marron)
- Catégorie « contrôle » (orange)



Classe de CE2 d'Emmanuelle Wilgenbus (Antony)






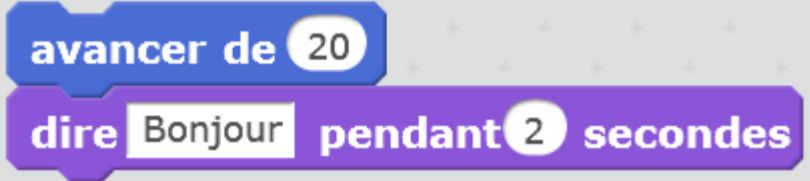





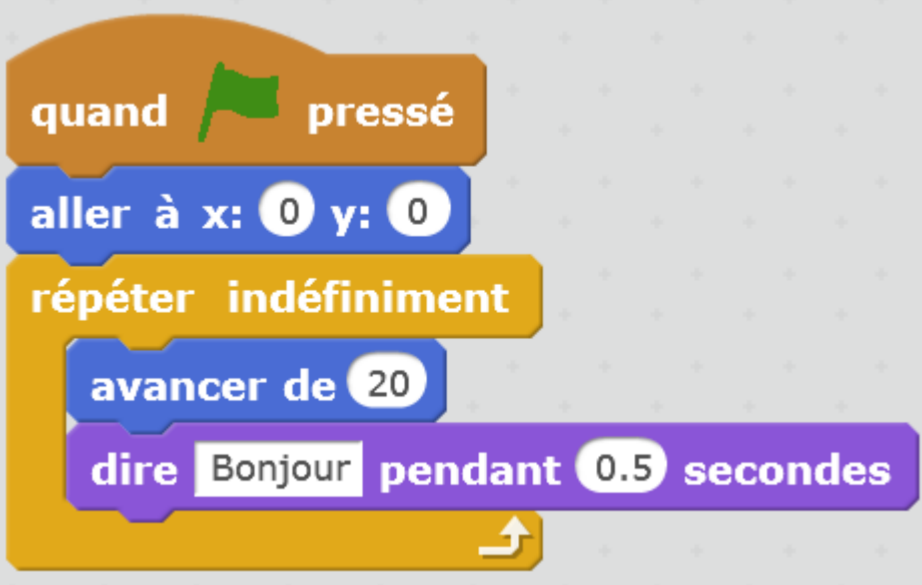
Note pédagogique :

Bien penser, dès maintenant, à instaurer une règle d'alternance afin que ce ne soit pas toujours le même élève qui contrôle le clavier et la souris.

Tâche 3 : faire de petits exercices (20 minutes)

L'enseignant donne une série de petits exercices (qui reprennent, pour la plupart, ce qui a été fait avant sous forme de démonstration), que les élèves exécutent. Après chaque exercice, une rapide mise en commun permet de s'assurer que chacun sait faire l'exercice.

Exercice 1 : faire avancer le chat de 10 pas

	<p>Exercice 2 : faire avancer le chat de 20 pas</p> <p>Plusieurs solutions possibles :</p>  <p>ou</p> <p>On préférera la seconde solution, plus élégante et facile à lire.</p>
	<p>Exercice 3 : remettre le chat au centre de la scène</p>  <p>Certains élèves vont sans doute trouver l'astuce... mais pour la plupart, il faudra la leur montrer. Malgré tout, il est indispensable pour eux de voir cette instruction dès maintenant, car, à force de déplacer le chat, ils vont le faire sortir de l'écran et ne sauront pas comment le récupérer.</p>
	<p>Exercice 4 : faire avancer le chat de 20 pas et lui faire dire « Bonjour »</p>  <p>Bien préciser que « dire » bonjour signifie pour nous « écrire » bonjour : on veut faire afficher une bulle à l'écran avec le texte « bonjour » dedans, on ne veut pas faire parler le chat ! NB : il est possible de faire parler le chat (lui faire émettre des notes de musique ou jouer un fichier son que l'on aura importé ou enregistré dans <i>Scratch</i>) ; nous le déconseillons fortement en classe.</p>
	<p>Exercice 5 : répéter 3 fois : faire avancer le chat de 20 et lui faire dire « bonjour »</p>  <p>Inciter les élèves qui ne trouvent pas à chercher dans la catégorie « contrôle » (orange). Ils y trouveront une instruction qui ressemble (« répéter 10 fois ») et qu'il est facile de modifier en remplaçant 10 par 3. Cette boucle enferme les autres instructions. Tout ce qui se trouve à l'intérieur de la boucle est exécuté 3 fois.</p> <p>On voit le chat qui s'arrête 2 secondes entre chaque mouvement... pour raccourcir cette pause, il suffit de raccourcir la durée pendant laquelle il dit « bonjour » (si on écrit « 0.5 » au lieu de 2, il ne s'arrêtera qu'1/2 seconde à chaque fois).</p>
	<p>Exercice 6 : répéter indéfiniment : faire avancer le chat de 20 et lui faire dire « bonjour »</p>  <p>Cela se fait très simplement, sur le même modèle que l'exercice précédent, mais avec un autre type de boucle.</p>
	<p>Exercice 7 : même chose quand on clique sur le drapeau vert</p>  <p>Il suffit a priori de rajouter l'instruction « quand drapeau vert pressé » (issue de la catégorie événement), mais cela est encore mieux si on demande au chat de repartir du centre de la scène. Expliquer, à ce moment, le rôle du bouton rouge (situé à côté du drapeau vert). Un clic sur ce bouton rouge met fin à l'exécution du programme (qui, sinon, ne s'arrête jamais dans le cas présent).</p>

Bilan et conclusion

La classe synthétise collectivement ce qui a été appris au cours de cette séance, notamment en listant les commandes scratch qui sont désormais connues de tous :

- Avancer de 10
- Aller à (0,0)
- Dire « bonjour » pendant 2 secondes
- Répéter 10 fois

- *Répéter indéfiniment*
- *Quand drapeau vert est pressé*

Il peut être très utile aux élèves de colorier les instructions Scratch au fur et à mesure qu'elles sont découvertes et comprises. Ainsi, après chaque nouvelle étape, on pourra constater les progrès du binôme ou de la classe entière.

La [Fiche 32 « Quelques commandes utiles dans Scratch »](#) peut être photocopiée pour chaque élève et agrandie pour la classe entière. Cette fiche sera enrichie plus tard, quand les élèves auront manipulé des [tests](#), des [capteurs](#), des [variables](#), et des [opérateurs](#).

[Séquence III-2](#)

[Etape III-2.2 >>](#)

Source URL: <https://www.fondation-lamap.org/fr/page/34519/1-2-3-codez-activites-cycle-3-etape-21-decouverte-de-scratch>