

Semaine mathématiques et numérique :

Quelques défis avec Scratch (Cycle 3)

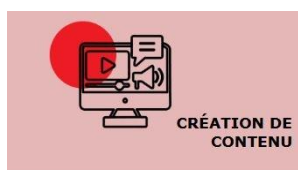
Document d'accompagnement des 5 défis 2020

Vous trouverez à la suite **une présentation de l'interface du logiciel Scratch3** (disponible à cette adresse <https://scratch.mit.edu> ou téléchargeable en [cliquant ici](#)) ainsi que **5 défis** qui vous permettront de débiter progressivement avec vos élèves et de leur faire découvrir les principales fonctions de *Scratch*. A chaque mission, sont associées des petites cartes d'aide (à découper pour pouvoir les distribuer séparément) que nous vous conseillons de ne donner aux élèves qu'en cas de « blocage » total car elles présentent tous les blocs à utiliser pour réussir la mission (et donc quasiment la solution au défi).

L'idée est de laisser les élèves tâtonner et faire des essais afin de trouver une solution (il peut y en avoir plusieurs) **qui réponde aux missions proposées.**

Ces défis permettent de travailler les compétences suivantes :

CRCN (Cadre de Référence des Compétences Numériques) :



3.4. Programmer : Écrire des programmes et des algorithmes pour répondre à un besoin (automatiser une tâche répétitive, accomplir des tâches complexes ou chronophages, résoudre un problème logique...) et pour développer un contenu riche (jeu, site web...) (avec des environnements de développement informatique simples, des logiciels de planification de

tâches...).

Création de contenu : Programmer

Niveau 1	Niveau 2	Niveau 3
- Lire et construire un algorithme qui comprend des instructions simples	- Réaliser un programme simple	- Développer un programme pour répondre à un problème à partir d'instructions simples d'un langage de programmation - Modifier un algorithme simple en faisant évoluer ses éléments de programmation - Mettre au point et exécuter un programme simple commandant un système réel ou un système numérique

Quelques compétences de mathématiques travaillées dans ces activités en cycle 3 :

(Se) repérer et (se) déplacer dans l'espace en utilisant ou en élaborant des représentations :	Reconnaitre, nommer, décrire, reproduire, représenter, construire quelques solides et figures géométriques :
Programmer les déplacements d'un robot ou ceux d'un personnage sur un écran.	Réaliser une figure simple ou une figure composée de figures simples à l'aide d'un logiciel.

Compétences transversales :

Domaine 1 : les langages pour penser et communiquer

Comprendre, s'exprimer en utilisant les langages mathématiques, scientifiques et informatiques

Manipuler le clavier pour taper du texte

Domaine 2 : Les méthodes et les outils pour apprendre

Utiliser l'ordinateur

Coopérer et réaliser des projets

Domaine 5 : Les représentations du monde et l'activité humaine

Programmer les déplacements d'un personnage sur un écran

<u>Sommaire</u> :	Eléments de programmation utilisés dans les défis.....	p3
	L'interface	p4
	Conseils	p7
	AIDE pour utiliser les capteurs	p8

Eléments de programmation utilisés dans les défis

Déplacement :

Utilisation de blocs permettant le déplacement et l'orientation d'un sprite.

Repérage :

Utilisation des coordonnées $(x ; y)$ dans le repère orthonormé.

Communication :

Un sprite communique un message écrit (bulle) ou oral (enregistrement d'un son).

Apparence :

Action sur les costumes d'un sprite, sur les arrière-plans ou utilisation des effets graphiques.

Boucle :

Utilisation d'au moins une boucle (répéter ...).

Programmation de touches :

Certaines touches du clavier sont programmées pour déclencher des actions.

Condition :

Utilisation de capteurs permettant de vérifier si une condition est remplie ou pas.

Temporisation :

Gestion des lancements des actions pour qu'elles démarrent à un moment précis ou durent un temps défini.

Boîte de dialogue :

Utilisation d'une boîte de dialogue dans laquelle l'utilisateur peut entrer des données (une « réponse »).

Test logique :

Utilisation des blocs « opérateurs » permettant de comparer deux éléments (texte ou numérique).

Nombre aléatoire :

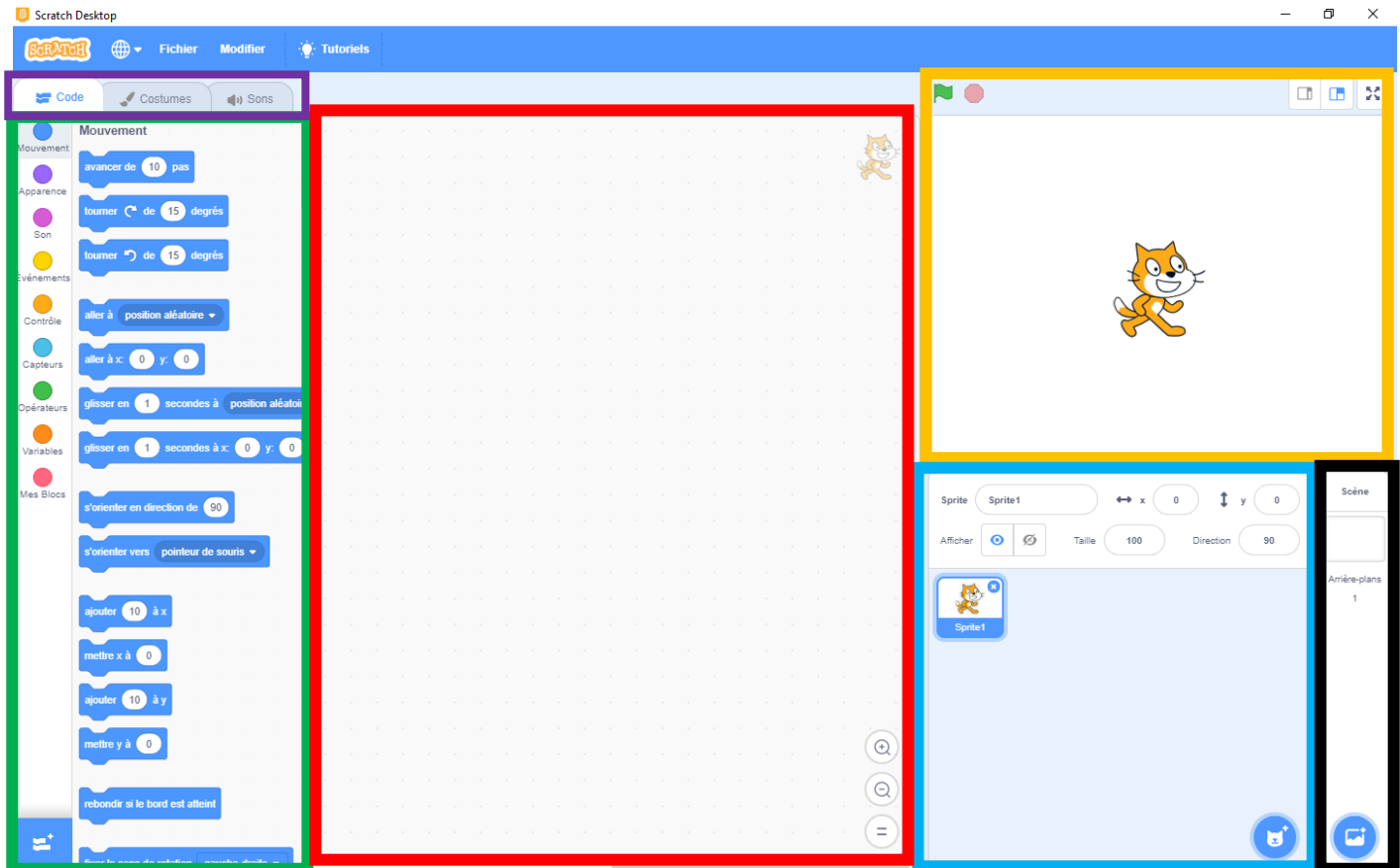
Utilisation de nombres aléatoires

Variable :

Utilisation d'une variable (valeur que l'on fait varier au cours du programme ; par exemple un score).

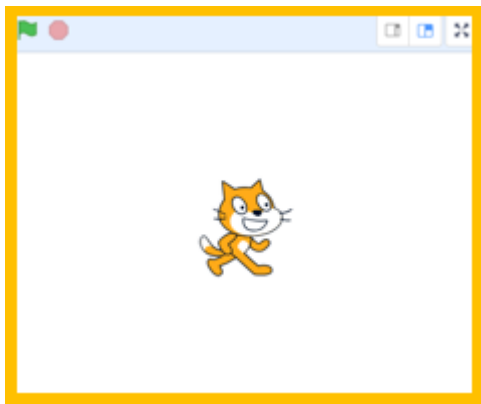
[illegible]

L'interface



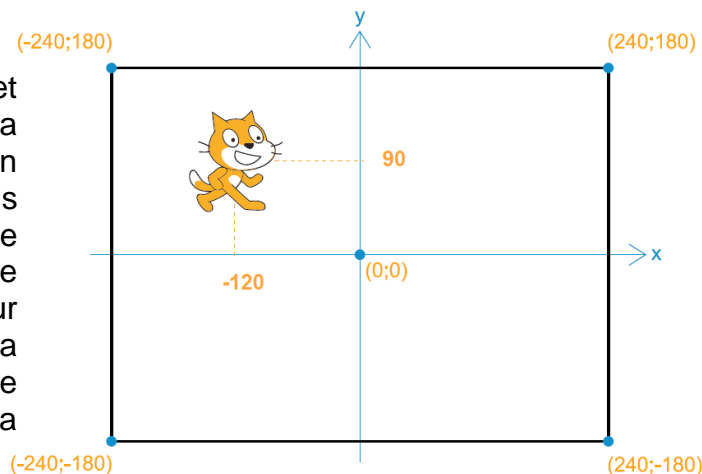
La scène :

C'est la zone où l'on peut voir le résultat de la programmation. On peut y afficher des arrière-plans, des objets ou des personnages à programmer.



Se repérer sur la scène :

Pour faciliter le positionnement et le déplacement des sprites sur la scène, celle-ci est munie d'un quadrillage invisible (plus précisément un repère orthonormé) pour lequel chaque nœud a des coordonnées (x pour la position horizontale et y pour la position verticale). Au démarrage Scratch est situé au centre de la scène, au point (x=0 ; y=0).



Taille de la scène

480 x 360 pixels


Repère orthonormé

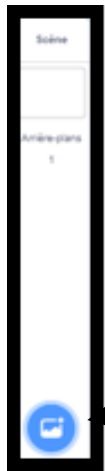
Unité = 1 pixel

Centre = centre de la scène




La zone des sprites :

Les sprites sont les objets et les personnages que l'on peut programmer. Lorsqu'on lance le logiciel, le Sprite1 (le chat *Scratch*) est automatiquement ajouté. Pour en ajouter d'autres, il suffit de passer la souris sur cette icône  pour ouvrir le menu et choisir son sprite.



La zone des arrière-plans :

Dans cette zone s'affiche l'arrière-plan actif (visible dans la scène) et en dessous le nombre d'arrière-plans qui ont été « chargé ». Au lancement du logiciel, c'est l'*arrière plan 1* (tout blanc) qui est actif.

Pour charger d'autres arrière-plans, il suffit de passer la souris sur cette icône  pour ouvrir le menu et choisir un arrière-plan.



La colonne des commandes :

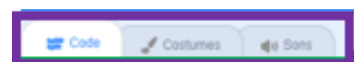
C'est ici que se trouvent tous les blocs de programmation. Chaque bloc correspond à un ordre (une commande). Les blocs sont classés dans 9 catégories qui ont chacune une couleur spécifique. Pour essayer un bloc, il suffit de cliquer une fois dessus.

La zone de codage :

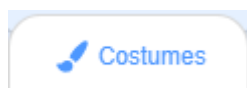
C'est ici que l'on écrit les programmes en faisant glisser des blocs de la colonne des commandes et en les assemblant. Pour commencer un programme, il faudra toujours un évènement déclencheur (dans la catégorie Evénements). L'arrière-plan ou le sprite qui est programmé est représenté en filigrane.



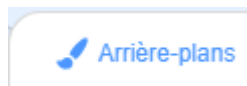
Les onglets :



Ils permettent de passer de la fenêtre de code (avec la colonne des commandes et la zone de codage) aux fenêtres de costumes, d'arrière-plan et de son.



L'onglet Costumes permet de voir les différents « costumes » (images correspondantes au sprite sélectionné), de les modifier ou d'en créer de nouveaux.



L'onglet Arrière-plans permet de voir les différents arrière-plans ayant été « chargés », de les modifier et d'en créer de nouveaux.



L'onglet Sons permet d'ajouter des sons (pour qu'ils apparaissent dans la liste déroulante des blocs « jouer le son ... ») issus de l'application ou de votre ordinateur. Si vous êtes équipé d'un micro (les ordinateurs portables en ont souvent un intégré), vous pouvez aussi vous enregistrer et utiliser ce son dans votre programme. Dans cette fenêtre, vous trouverez également des outils vous permettant de modifier les sons (en les accélérant, les ralentissant, ou en appliquant d'autres effets).

Les menus :



Regarder des tutoriels vidéos (en anglais sous-titrés en français)

Modifier la langue de l'interface

- Faire réapparaître le dernier sprite ou arrière-plan supprimé
- Activer le mode Turbo

- Créer un nouveau projet
- Ouvrir un projet enregistré sur l'ordinateur
- Enregistrer le projet en cours

Conseils :

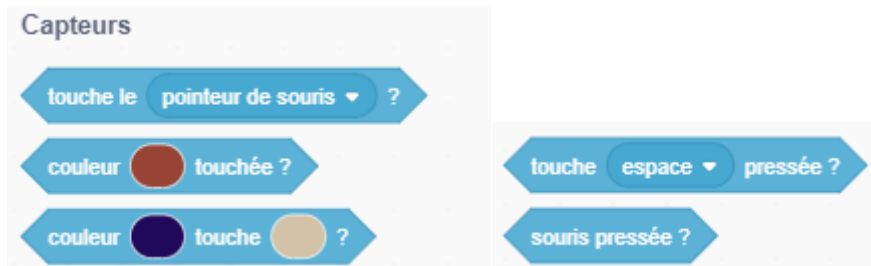
- ❖ **Si vos élèves n'ont jamais utilisé Scratch**, il peut être intéressant (voire nécessaire) de **faire une séance préalable de découverte de l'interface**. Cette séance peut être menée en classe entière si vous disposez d'un TBI (ou simplement d'un vidéoprojecteur). Laissez les élèves observer, puis demandez-leur de décrire ce qu'ils voient. Profitez-en pour leur **apporter le vocabulaire précis** pour faciliter les futurs échanges (il est plus simple de trouver le drapeau vert si on dit qu'il est « en haut à gauche de la scène » plutôt que de dire qu'il est « presque au milieu en haut »...).
- ❖ Par expérience, **l'organisation la plus efficace** pour faire réaliser ces défis aux élèves est de les mettre par **groupe de 2** (en veillant à ce qu'il n'y ait pas de trop grande différence de niveau entre ces 2 élèves). Il peut parfois être nécessaire d'instaurer des règles pour établir un « tour de manipulation de l'ordinateur » en expliquant l'importance du rôle de l'élève qui ne manipule pas pour vérifier ce qui est fait et apporter son aide. Par exemple, on peut proposer un changement à chaque étape d'un défi (un élève prend la souris pour l'étape *a*) et l'autre pour l'étape *b*).
- ❖ Comme le temps passe toujours vite quand on fait de la programmation, je vous conseille de prendre un petit temps pour **faire enregistrer aux élèves leur projet avant de commencer à programmer**.
 - Avec le logiciel installé :
Leur indiquer dans quel dossier enregistrer leur programme et le nom à lui donner. Par exemple, les prénoms des élèves créateurs, suivis du titre du projet. Une fois la séance terminée, ils auront juste à cliquer sur « Sauvegarder sur votre ordinateur », puis sur « Enregistrer » dans la fenêtre qui va s'ouvrir et enfin sur « Oui » pour confirmer le remplacement de la version précédente.
 - Avec la version en ligne : vous avez 2 possibilités
 1. Enregistrer les projets sur l'ordinateur (vous pourrez ensuite les rouvrir en cliquant sur « Load from your computer ») : remplacez « Untitled » par les prénoms des élèves créateurs, suivis du titre du projet, puis enregistrez le projet (vous le retrouverez dans vos téléchargements).
 2. Enregistrer les projets en ligne : crée un compte pour la classe au préalable, connectez-vous, puis remplacez « Untitled » par les prénoms des élèves créateurs, suivis du titre du projet. Cliquez sur « Enregistrer maintenant ». Vous retrouverez les projets enregistrés en ligne en cliquant sur le dossier un haut à droite (à côté de votre identifiant de connexion).
- ❖ **Lors de la première séance**, il vous faudra trouver une organisation de classe (fonctionnement en atelier par exemple) qui vous permettra d'être assez disponible pour être certain que les élèves qui réalisent les défis ne se dispersent pas, pour pouvoir leur donner de petits indices pour les faire avancer et surtout **les inciter à faire des essais**. Comme indiqué dans l'introduction de ce document ne donnez les fiches « Aide » qu'en dernier recours ou si vous ne pouvez pas vous rendre disponible pour aider les élèves « bloqués ». **Pour les séances suivantes, les élèves devraient pouvoir être autonomes**. Vous aurez juste à donner les fiches « Aide » après un délai raisonnable de recherche infructueuse.
- ❖ Si vous en avez la possibilité, imprimez les fiches « Aide » en couleurs (elles seront plus lisibles pour les élèves et leur permettront de trouver plus facilement les blocs à utiliser) et plastifiez-les afin de pouvoir les conserver dans le temps.

Et après ?

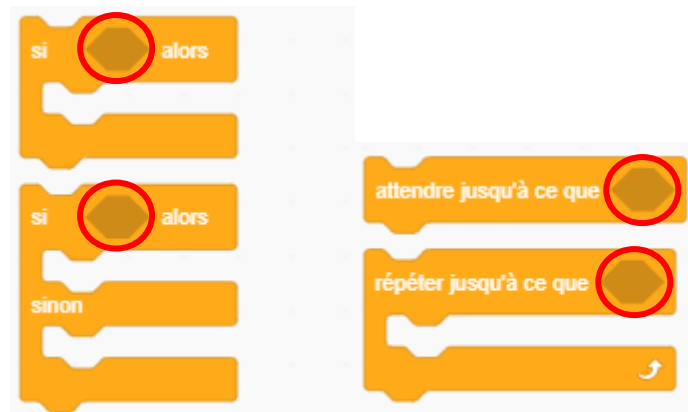
Rendez-vous sur le site Numérique éducatif 80 (<http://tice.dsden80.ac-amiens.fr/>) pour retrouver les défis programmation proposés par l'équipe des ERUN de la Somme depuis 2017.

Utiliser les capteurs

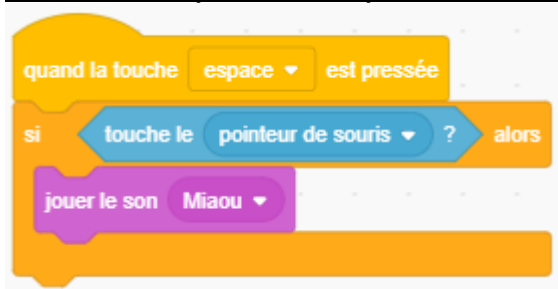
L'utilisation de capteurs permet de savoir si une couleur, un sprite, le pointeur de souris, le bord de la scène, sont touchés ou si une touche du clavier ou de la souris est pressée.



Ces blocs sont « en pointe » et s'insèrent dans les blocs de contrôle suivants :



Crée le script suivant pour Scratch :



Place le pointeur de souris (la flèche) sur Scratch puis appuie sur la touche « espace ». **Que se passe-t-il ?**

Remplace le bloc « quand la touche espace est pressée » par . Clique sur le drapeau vert puis place le pointeur de souris sur Scratch. **Que se passe-t-il ?**



Programme ce script (on ajoute « répéter 100 000 fois »). Clique sur le drapeau vert puis place tout de suite le pointeur de souris sur Scratch. **Que se passe-t-il ?**

Défi :

Une fois le drapeau vert cliqué, à chaque passage du pointeur de la souris sur Scratch, on entend le son « Miaou ».