

# Initiation à Scratch :

## une somme pour commencer

### Fiche professeur

6<sup>e</sup>

Auteur : Nom de l'auteur

**But de l'activité :** Découvrir l'environnement Scratch après une 1ère initiation à l'usage des blocs pour commander la construction d'une figure hors Scratch, découvrir la gestion des entrées/sorties et des variables.

#### Compétences engagées :

- ✓ Prendre en main l'interface du logiciel Scratch : scène, lutin, blocs, catégories, interactivité, données, opérateurs, sauvegarde
- ✓ Compétences de base en algorithmique et programmation : entrées, sorties, variables, calculs
- ✓ Compétences de base dans la gestion d'un projet : retrouver une information déjà donnée dans un projet pour la réutiliser dans un nouveau projet

#### Pré-requis :

- ✓ Informatique : savoir lancer le navigateur internet et régler les problèmes de sécurité
- ✓ Mathématiques : vocabulaire de la somme et du produit

#### Matériels utilisés :

- ✓ Site informatique
- ✓ Vidéoprojecteur ou logiciel de présentation du poste professeur sur les postes clients en classe pupitre
- ✓ Un TNI/VPI est un plus pour montrer les manipulations

#### Durée indicative : 2h

- ✓ 1h pour faire le I Somme
- ✓ + 1h pour la correction et II Produit

#### Nom des logiciels utilisés :

- ✓ Navigateur Internet (FireFox version 40) pour le site Scratch : <https://scratch.mit.edu/projects/editor/>
- ✓ (alternative : logiciel Scratch 2 installé)

#### Documents utiles à télécharger :

- ✓ La fiche élève à ventiler via le réseau pour conserver la couleur des blocs !
- ✓ Exemples dans le [Studio Produit](#) pour la correction du projet Somme par exemple

#### Déroulement de la séance :

##### a) Contexte :

Cette séance est la 2ème après une première heure passée sur le site [Studio.Code.org](http://Studio.Code.org) afin d'initier les élèves à la programmation par blocs. Il s'agissait de faire du « pseudo » Logo via le [parcours L'Artiste](#) : piloter un petit bonhomme qui construit des polygones en utilisant des blocs « Avance » et « Tourne », et découvrir le bloc « Répéter » .

Outre le contexte géométrique (mesures et angles), les élèves ont ainsi pratiqué une interface plus simple que celle de Scratch, de manière efficace pour le professeur : le succès de l'élève est évalué par le logiciel ce qui permet au professeur de se concentrer sur l'aide aux élèves en difficulté (notion mathématiques, interface ou notion d'algorithmique) et de prendre la main pour faire des mises au point collective (présentation initiale, puis au besoin comme pour préciser la boucle « répéter » par exemple).

- b) Au début de la séance, le professeur prend en charge la connexion sur le site. Notamment il faut gérer les problèmes d'autorisation pour l'usage du plugin Flash en plus du réglage éventuel de la langue et du titre du projet. (la 2nde fois les élèves savent faire en général). Avec la version installée, ces difficultés pratiques et les lenteurs du 1<sup>er</sup> chargement sont inexistantes
- c) Le professeur indique l'objectif d'un premier projet d'initiation à la programmation : réaliser un programme/un logiciel pour faire la somme de 2 nombres. Un exemple est choisi collectivement et le vocabulaire mathématique est précisé : 1<sup>er</sup> terme, 2<sup>e</sup> terme, somme. Cela servira à la question 6 et 7 à préciser que les variables stockent les nombres saisis et que leur nom est choisi pour être parlant.
- d) Ensuite le professeur fait une démonstration en s'appuyant sur l'expérience du parcours L'Artiste pour détailler l'interface du logiciel en faisant bouger le Lutin pour préciser les « lieux » et le vocabulaire : lutin, bloc, catégorie ... Ainsi la pose du bloc « Avancer » nécessite l'ajout d'un bloc « Quand drapeau vert cliqué » afin de pouvoir déclencher la procédure. Le bloc se trouve dans la catégorie « Événements » : cela permet d'indiquer aux élèves l'importance de la couleur des blocs qu'ils auront dans les fiches tutoriels pour savoir où les chercher/trouver.
- e) Dès lors, les élèves terminent les questions à 1 à 3.
- f) Question 6, il faut parler du rôle de ces variables car il y a une variable préexistante dans Scratch : la variable « réponse » pour récupérer ce que l'utilisateur a écrit quand le lutin demande une saisie à l'utilisateur. Dans les langages de programmation par texte, demander une saisie se fait avec une fonction qui renvoie directement la valeur de la réponse (texte ou nombre) laquelle doit être directement affectée à une variable : `var terme1=prompt("Quel est le 1er terme ? »)`. Avec Scratch, « demander » n'est pas un bloc fonctionnel comme un bloc « Opérateur » pour une addition. D'où cette variable globale très utile.
- g) D'ailleurs certains élèves ne trouvant pas « réponse » malgré sa couleur crée une variable réponse. Il est possible de la supprimer

par clic droit sur sa déclaration dans la catégorie Données. On peut aussi renommer les variables mal orthographiées (même si cela n'a aucune importance sur le comportement du programme).

- h) Au pire il faut faire ou refaire un point sur les variables à l'occasion de la question 7 ou 8 quand les élèves bloquent à cause de la question 7 « oubliée ».
- i) La question 9 est quelques fois oubliée.
- j) Enfin les plus rapides font les questions 10 et 11.
- la 10 est délicate à comprendre, surtout quand certains ont découvert qu'en décochant un variable dans la catégorie Donnée, on la rendait invisible sur la scène .
  - la 11 est par contre parfois faite assez rapidement dès début de séance par les élèves « cliqueurs » qui découvrent tout seul.
- k) L'enregistrement est nécessaire, non seulement pour revoir le projet la fois suivante si besoin est, mais surtout pour montrer la procédure afin que les séances suivantes elle ne pose pas de problème. Cela permet au professeur de retrouver les projets avec le bon nom au bon endroit afin de les récupérer pour les évaluer.
- l) La séance suivante a débuté par une correction du projet, c'est-à-dire la présentation d'un projet simple puis d'un projet avec les embellissements par étape : d'abord la gestion du masquage/affichage des variables qui est important pour l'ergonomie du programme, puis par les embellissements importants pour donner envie à l'utilisateur potentiel (tout en évitant de prendre le dessus sur l'ergonomie/l'objectif essentiel : le programme doit faire le travail qu'on attend de lui !)
- m) Ensuite, le II a été lu et précisé, essentiellement sur la décomposition du travail :
- même si c'est tentant de faire le contraire, faire les embellissements APRES que le programme soit réellement fonctionnel, notamment pour les moins rapides pour finir l'essentiel dans les temps ;
  - ne pas repartir du projet somme mais repartir des consignes de la fiche pour réaliser le projet Somme. Pour les élèves/groupes plus faibles, montrer le projet Somme en vidéoprojection peut aider à atteindre l'objectif principal (avoir un algorithme complet fonctionnel) et permet de gérer les demandes d'aide des élèves motivés.
- Des projets remarquables ont été rassemblés sur le site Scratch dans un [Studio Produit](#) :
- 2 projets Somme : une version basique et une version embellie
  - quelques projets Produit réalisés par des élèves.

### **Variantes / Pour aller plus loin / Références :**

✓ **Pour aller plus loin :**

en DM ou exercice, proposer de réaliser une amélioration du projet Produit indiquant le signe du résultat après une découverte du bloc conditionnel « si ... alors... » (gestion positif, négatif et nul !). Voir activité IREM n° ...

✓ **Pour aller plus loin :**

en DM ou exercice, proposer de réaliser un projet division avec gestion de la division par zéro (toujours avec bloc conditionnel « si ... alors ... »).

✓ **Pour aller plus loin :**

en DM ou exercice, proposer de réaliser un projet où l'utilisateur doit indiquer l'opération (+-\*/) avant de demander les nombres à utiliser (toujours avec bloc conditionnel « si ... alors ... »).