Préambule

Ce livre s'adresse aux personnes qui ont envie d'apprendre à programmer et qui aiment les mathématiques. Il propose une initiation à la programmation dans un contexte mathématique. La programmation informatique est, en effet, devenue un enjeu important de notre société. Le milieu éducatif québécois réfléchit actuellement aux enjeux didactiques et citoyens de son intégration à l'école. En effet, parmi les compétences numériques attendues, celles liées à l'usage de la programmation informatique sont de plus en plus au cœur des débats éducatifs.

Le contenu proposé dans ce libre s'inspire des résultats d'un projet de recherche intitulé « Intégration de la programmation en mathématiques et en sciences au secondaire : quels enjeux pour la formation des enseignants ? » Ce projet a été mené en collaboration avec Martin Baril du Centre de Services Scolaires de la Capitale et de Sonya Fiset, du RECIT_MST. Il a été financé par le ministère de l'éducation du Québec et le Fonds de recherche du Québec-Société et culture (FRQSC).

La particularité de ce livre est de faire le lien entre les mathématiques et la programmation informatique. Tous les exemples et exercices utilisés dans ce livre sont mathématiques. La pensée algorithmique, celle que l'on met en œuvre quand on programme, partage de nombreuses caractéristiques avec la pensée mathématique. On peut s'appuyer sur l'une pour développer l'autre. La programmation peut permettre, par exemple, d'explorer les régularités de certains nombres ou bien tester des conjectures mathématiques sur un grand nombre de cas.

Nous avons choisi d'illustrer notre approche avec la plateforme Scratch, mais notre propos algorithmique est général, et tous les exemples pourraient très bien être programmés dans un autre langage. Ce livre ne constitue pas à proprement parler une initiation à Scratch. On n'y trouvera pas, par exemple, d'information sur les possibilités énormes d'animation et de création visuelles dans Scratch. Il est centré sur les possibilités algorithmiques et mathématiques que l'on peut exploiter dans Scratch.

La résolution algorithmique d'un problème mathématique est rarement unique. Les exemples ou les corrections que nous proposons sont donc toujours à considérer comme des solutions possibles, parmi tant d'autres. N'hésitez pas à partir à la recherche de celles qui vous sont propres !

Chapitre 1

Programmes et instructions élémentaires

Les hommes n'ont pas attendu l'arrivée des ordinateurs pour programmer. En fait, les premiers algorithmes remontent à l'Antiquité, dès l'apparition des premières tablettes (faites d'argile). Dès cette époque, ils sont liés aux mathématiques et aux calculs. Le plus connu des algorithmes mathématiques (à défaut d'être le premier) est celui d'Euclide, aux environs de -300 avant notre ère. L'algorithme d'Euclide permet de déterminer le plus grand commun diviseur de deux nombres entiers, sans connaître leur factorisation. On le trouve dans le livre VII des *Éléments* d'Euclide. Dans la même période, Archimède (287 av. J.-C.) a proposé une méthode pour calculer la somme d'une série infinie donnant une approximation de Pi et d'Ératosthène (276 av. J.-C.), et a établi « le crible d'Ératosthène », un algorithme qui permet de déterminer par exclusion tous les nombres premiers. Le mot *algorithme* lui-même est apparu un peu plus tard. Il vient du nom d'un mathématicien persan Al Khwarizmi né en 780 qui a rédigé un des premiers ouvrages recensant des méthodes pour résoudre des équations. Toutes les méthodes que nous venons de citer étaient destinées à être exécutées par des êtres humains.

Beaucoup plus tard, lorsque les premiers ordinateurs sont apparus, on a eu l'idée d'écrire des algorithmes exécutables par une machine sous forme de programmes écrits dans des langages de programmation. C'est la naissance de la programmation informatique.

Définition 1
Programmer

Programmer, c'est donner des instructions à un ordinateur pour qu'il les exécute dans le but de réaliser une tâche.

Il est important de comprendre que l'ordinateur ne prend aucune initiative. Il exécute fidèlement les instructions qu'on lui donne. **Un ordinateur ne fait QUE ce qu'on lui demande. Il fait EXACTEMENT ce qu'on lui demande.**

Définition 2
Instruction

Une instruction est une opération de base qui peut être exécutée par l'ordinateur ; cela signifie, entre autres, que cette instruction doit être écrite dans un langage que l'ordinateur peut comprendre. Pour faire cela, nous utilisons des langages de programmation.

Dans cet ouvrage, nous utilisons le langage Scratch. Par exemple, l'instruction ci-dessous est une instruction du langage Scratch permettant d'additionner deux valeurs.

Quand elle est exécutée, elle renvoie le résultat de l'opération.

Pour afficher le résultat de ce calcul, il faut donner une autre instruction à l'ordinateur. L'exécution de cette deuxième instruction permet de faire apparaître le résultat du calcul à l'écran. Dans Scratch, c'est un personnage, appelé lutin, qui effectue à l'écran les actions demandées.

Instructions	Affichage obtenu
	

Définition 3
Programme

Un programme informatique est un ensemble structuré d'instructions écrites dans un langage de programmation et destinées à être exécutées par un ordinateur.

Exercice corrigé 1.1

Voici un programme Scratch :

Pouvez-vous lister les instructions élémentaires qui le composent ? Que fait ce programme quand on l'exécute ?

Réponse possible

Ce programme comporte 4 instructions élémentaires.

Instructions	Affichage obtenu	
Afficher un message pendant deux secondes.		Durée : 2 secondes
Afficher un message pendant deux secondes.		Durée : 2 secondes
Calculer 3 + 5.	Aucun	
Afficher le résultat de l'instruction précédente.		Durée : infinie

Dans le programme que nous venons d'écrire, les instructions sont exécutées les unes après les autres, dans l'ordre d'écriture. On dit qu'un tel programme est séquentiel.

Définition 4
Séquence

La séquence est la forme la plus simple d'enchaînement d'instructions composant un programme. Les instructions sont écrites et exécutées les unes après les autres.

Nous voulons maintenant modifier le programme de l'exercice 1. Nous cherchons à le généraliser et à faire en sorte qu'il puisse effectuer l'addition d'autres nombres. Pour cela, nous avons besoin de pouvoir changer les valeurs de ces nombres. Ces valeurs devront être enregistrées dans la mémoire de l'ordinateur pour ensuite pouvoir être utilisées dans le calcul. On utilise pour cela des variables.

Définition 5
Variable

Une variable est un espace mémoire dans lequel on stocke une valeur qui va pouvoir changer durant le déroulement du programme. Cet espace mémoire est repéré par un nom qui donne accès à la valeur. Dans l'exemple 2 ci-dessous, la variable se nomme *nombre1* et elle contient la valeur 10.

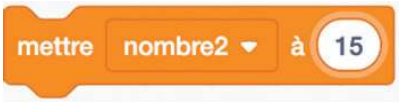
Une instruction peut combiner une opération et des variables. L'instruction ci-dessous permet d'additionner la valeur contenue dans la variable *nombre1* avec celle contenue dans la variable *nombre2*.

Pour attribuer une valeur à une variable, on utilise une opération appelée *affectation*. Dans Scratch, elle est représentée par l'instruction *mettre à*.

Définition 6
Affectation

L'affectation est l'opération qui permet d'attribuer une valeur à une variable. Si la variable contient déjà une valeur, celle-ci disparaît. On dit qu'elle est écrasée par la nouvelle valeur.

L'instruction ci-dessous affecte la valeur 15 à la variable *nombre2*.



Définition 7
Entrées et sorties

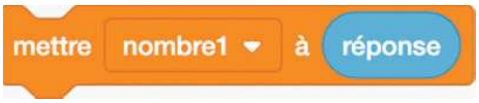
Le terme *Entrées/Sorties* recouvre toutes les façons dont un ordinateur peut communiquer avec l'extérieur, pour recevoir de l'information (entrée) ou pour en envoyer (sortie).

Dans cet ouvrage, l'entrée d'information dans un programme se fera par le clavier. Dans le langage Scratch, c'est l'instruction `demander` qui permet d'afficher une fenêtre dans laquelle un curseur indique à l'utilisateur qu'il doit entrer quelque chose (un mot, un message, un nombre…) à l'aide du clavier.

On peut ajouter un message dans la bulle vide pour donner des instructions à l'utilisateur. Par exemple, l'instruction ci-dessous provoque l'apparition du message « Quel est ton nom ? » et d'une fenêtre de dialogue sous le lutin. Le système reste en attente jusqu'à ce qu'une valeur soit entrée au clavier.

Instruction	Affichage obtenu
	

La valeur entrée au clavier est capturée dans une variable prédéfinie appelée *réponse*. On peut ensuite affecter cette valeur à une autre variable.



Nous sommes maintenant capables d'écrire un programme qui effectue l'addition de deux nombres entrés par l'utilisateur et qui affiche le résultat.