

CODAGE

GUIDE D'ENSEIGNEMENT EFFICACE DES MATHÉMATIQUES



TABLE DES MATIÈRES

1. POURQUOI.....	4	2. COMMENT	15
INTRODUCTION.....	4	CODER AVEC INTENTION.....	15
DÉMYSTIFIER LE CODAGE.....	4	L'intégration du codage dans la planification à long terme	15
« APPRENDRE À CODER » ET « CODER POUR APPRENDRE ».....	5	Planifier une activité de codage.....	16
Le pseudocode, un outil indispensable.....	5	Les pratiques pédagogiques à fort impact	16
Passer du codage débranché au codage branché.....	5	Résultats d'apprentissage, critères d'évaluation et rétroaction descriptive.....	17
Le codage par blocs	7	Enseignement explicite.....	17
Le codage textuel.....	8	Tâches et expériences liées à la résolution de problèmes.....	17
DÉVELOPPER LA PENSÉE COMPUTATIONNELLE.....	9	Enseignement quant à la résolution de problèmes	17
Les six processus mentaux.....	10	Outils et représentations	18
Les concepts mathématiques et le codage.....	10	Conversations mathématiques.....	18
Les processus mathématiques et les habiletés socioémotionnelles en codage.....	11	Enseignement en petits groupes.....	18
L'importance des erreurs.....	12	Pratique délibérée	19
Le débogage	12	Regroupements flexibles	19
Le codage au service du développement des compétences transférables	13	Comment choisir un logiciel de codage	19
La créativité de l'élève est au volant!.....	13	ET L'ÉVALUATION?.....	21
Communiquer, c'est essentiel!.....	13	Le codage et l'évaluation	21
		Évaluer le codage et évaluer avec le codage.....	21
		Les critères d'évaluation et les quatre compétences	22
		Comment recueillir des preuves d'apprentissage	24
		Observations.....	24
		Conversations.....	25
		Productions.....	27
		La rétroaction.....	27
		La rétroaction descriptive	28
		La rétroaction par questionnement	28
		L'évaluation par les pairs.....	28

3. QUOI 29 4. ET ENSUITE... 159

LA MAIN À LA PÂTE 29

À quoi peut ressembler le codage? 29

Le codage débranché 29

Le pseudocode 31

Le codage dans un contexte mathématique 32

Les habiletés liées au codage 33

Progresser à travers les connaissances en codage 34

Les événements séquentiels 34

Les événements simultanés 36

Les événements répétitifs 38

Les événements imbriqués 40

Les instructions conditionnelles 43

L'efficacité dans le contexte du codage 46

Les sous-programmes 47

L'analyse de données avec le codage 49

Situation d'apprentissage – 1^{re} année 51

Situation d'apprentissage – 2^e année 60

Situation d'apprentissage – 3^e année 73

Situation d'apprentissage – 4^e année 88

Situation d'apprentissage – 5^e année 105

Situation d'apprentissage – 6^e année 116

Situation d'apprentissage – 7^e année 130

Situation d'apprentissage – 8^e année 149

L'EXPÉRIENCE DE L'ÉLÈVE 159

Les liens entre le codage et l'enseignement des STIM
(des problèmes modernes nécessitent des
solutions modernes) 159

S – Sciences 159

T – Technologie 160

I – Ingénierie 160

M – Mathématiques 161

Les liens entre le codage et la vie quotidienne 162

Les liens entre le codage et les carrières 164

LA TERMINOLOGIE DU CODAGE 165

ANNEXE

MODÉLISATION MATHÉMATIQUE 170

1. POURQUOI

Introduction

Le *Guide d'enseignement efficace en mathématiques - Codage* a été conçu pour appuyer la mise en œuvre en salle de classe du codage en offrant des stratégies d'enseignement efficaces et des exemples de pratiques réussies. Le codage est une partie d'un concept encore plus englobant : la programmation informatique. Les gens en programmation résolvent des problèmes du quotidien en créant du code pour répondre à des besoins divers. Pour ce faire, ils doivent faire preuve d'empathie envers les gens qu'ils veulent aider et apprendre à connaître leurs besoins. Il ne s'agit donc pas seulement de coder, mais aussi de concevoir une interface afin de s'assurer que les utilisatrices et les utilisateurs ont une expérience agréable. Finalement, la programmation nécessite aussi la mise à l'essai du code auprès des utilisatrices et des utilisateurs, et l'application de changements, au besoin. Ce guide cible plus particulièrement l'étape pendant laquelle l'élève transforme une situation mathématique en un langage pouvant être compris par une machine.

Démystifier le codage

Afin de mieux saisir ce à quoi le codage ressemble aujourd'hui, il peut être utile de retourner à ses débuts. Bien que le codage soit peut-être une nouveauté dans les salles de classe, il s'agit, en fait, d'un concept qui date de 1843 lorsqu'une pionnière de l'informatique, Ada Lovelace, a créé le langage de programmation permettant de coder. C'est dans les années 1950 que plusieurs langages ont fait leur apparition, comme l'Assembleur, le COBOL et le LISP. Ces langages pouvaient sembler très rébarbatifs aux novices, ce qui fait qu'à l'époque seulement les informaticiennes et les informaticiens avaient le privilège de coder les ordinateurs.

C'est dans les années 1980 que, finalement, un premier langage de programmation « grand public » a fait son apparition. Les gens étaient capables de coder un budget, un livre de recettes interactif ou même un jeu vidéo. Depuis, le codage et les langages de programmation ont beaucoup évolué, les rendant accessibles à toutes et à tous, même aux jeunes enfants. Au début des années 2000, la programmation graphique à l'aide de « blocs » a fait son apparition, réduisant ainsi l'incidence des erreurs de syntaxe que l'on trouvait dans les langages textuels. Plusieurs interfaces, robots et microcontrôleurs utilisant du codage à l'aide de blocs sont apparus.



Le codage aide à faire exécuter des tâches aux ordinateurs et aux appareils. Afin que les élèves puissent en voir la pertinence, il est important de leur faire prendre conscience que le codage et les mathématiques sont partout autour de nous. Que ce soit le thermostat intelligent, la cuisinière, la télécommande, la montre connectée, la brosse à dents électrique, la voiture autonome, ces appareils contiennent tous du code et font tous appel à des concepts mathématiques pour fonctionner. Par exemple, la brosse à dents utilise la mesure des angles et la mesure de la force pour recueillir des données sur les habitudes de brossage, le thermostat et la cuisinière utilisent des comparaisons ($>$, $<$ ou $=$) pour vérifier si une certaine température a été atteinte et le GPS d'un téléphone intelligent utilise des distances et la trigonométrie afin de prédire le temps requis pour se rendre du point A au point B. Le personnel enseignant est invité à profiter de tous les contextes de la vie quotidienne pour établir des liens.

Le codage est accessible à toutes et à tous. Des interfaces et des technologies d'assistance aident maintenant les élèves avec des besoins particuliers à faire du codage. La surdit , la c cit , la mobilit  r duite ou les d fis cognitifs, par exemple, ne pr sentent plus un obstacle   l'apprentissage du codage. Plusieurs plateformes de codage accessibles sont maintenant offertes en ligne et plusieurs articles sont disponibles chez les fournisseurs en  ducation sp cialis e.

« Apprendre à coder » et « coder pour apprendre »

Bien qu'il soit important au départ pour l'élève d'apprendre à coder en acquérant les rudiments des blocs de code ou de la syntaxe du langage, il est crucial de comprendre que l'objectif du codage dans la salle de classe est de « coder pour apprendre ». Dans cette optique, l'élève utilise le codage pour explorer de nouveaux concepts en mathématiques ou consolider des apprentissages existants.

Explorer de nouveaux concepts : l'élève pourrait créer du code qui permet de faire des milliers de tirages à la seconde pour comparer la probabilité expérimentale avec la probabilité théorique. L'élève pourrait aussi tenter de tracer des polygones pour déterminer les régularités dans les angles extérieurs et intérieurs en fonction du nombre de côtés. Dans ce cas, l'élève crée des simulations à l'aide du code pour tirer des conclusions.

Consolider des apprentissages existants : l'élève pourrait utiliser le codage pour montrer sa compréhension du concept de l'emplacement relatif d'objets (par exemple, en faisant déplacer un personnage à gauche, à droite, au-dessus, en dessous d'un objet) ou utiliser le théorème de Pythagore pour coder un déplacement en forme de triangle rectangle. Dans ces cas, l'élève connaît déjà le concept mathématique et utilise le codage pour le représenter, lui permettant de le mettre en application et de consolider sa compréhension.

Bien que plusieurs activités et plans de leçons existent en ligne pour raconter des histoires ou créer des jeux vidéo à l'aide du codage, il faut se rappeler que l'objectif principal ici est de représenter des concepts mathématiques. Ces activités peuvent aider à apprendre à coder, mais aux fins d'évaluation de l'attente, il faut s'assurer que l'élève code aussi pour approfondir sa compréhension des concepts mathématiques. Les concepts et les situations à représenter peuvent, et devraient, venir de tous les domaines. Le codage peut être un appui important à la planification en spirale, étant donné que les élèves peuvent toujours s'appuyer sur leurs connaissances et leurs expériences afin de représenter de nouvelles situations et de résoudre de nouveaux problèmes. Le codage ne devrait donc pas être enseigné une seule fois à un seul moment.

Le pseudocode, un outil indispensable

Le terme **pseudocode** signifie simplement que la séquence de directives que l'ordinateur devra suivre est écrite dans un langage familier et organisé comme une séquence de code. Il est beaucoup moins intimidant pour l'élève de parler une langue qui lui est familière au départ avant de devoir utiliser des blocs de code ou un langage informatique comme Python ou JavaScript. De plus, le pseudocode n'appartient à aucune langue, alors l'organisation et la syntaxe utilisées doivent être comprises par l'élève et ses pairs, et non par un ordinateur.

Le pseudocode est aussi une approche intéressante au débogage, soit la détection et la correction d'erreurs dans un code. En rédigeant un pseudocode et en le comparant avec un code non fonctionnel, l'élève peut trouver l'endroit où une erreur s'est insérée.

Des exemples de pseudocodes, dans ce guide, sont souvent organisés sous forme de tableaux, et peuvent être trouvés dans les situations d'apprentissage et dans la section traitant du [pseudocode](#).

Passer du codage débranché au codage branché

Lorsqu'on code sans ordinateur, il s'agit de **codage débranché**, un nom qui fait référence au fait que nous n'avons pas besoin d'être « branchés » à l'ordinateur pour coder. Lorsque l'élève accomplit une tâche mathématique qui ressemble à une séquence de code, il s'agit de codage débranché. Le **pseudocode** est aussi une forme particulière de codage débranché et peut agir comme un intermédiaire efficace afin de passer du codage débranché au codage branché.

Lorsque l'élève commence à apprendre à coder, le personnel enseignant utilise souvent des activités débranchées pour présenter le concept. Ces activités aident l'élève à comprendre de façon très concrète des concepts comme la séquence sans l'utilisation d'un ordinateur, d'une tablette, d'un robot ou d'un microcontrôleur. Par exemple, dans une activité débranchée, une personne peut prendre le rôle du codeur et une autre peut prendre le rôle de la machine. Une personne pourrait donner une séquence d'instructions à un pair pour faire un lancer dans un panier de basketball ou donner une séquence d'instruction pour que deux autres élèves puissent se taper dans les mains.



L'élève se rendra compte que, même si ses instructions sont rudimentaires, l'élève agissant comme une machine complètera souvent l'information manquante. Par exemple, si l'on dit « lève la main » sans donner de distance, l'élève ne continuera pas à lever la main indéfiniment. L'utilisation du codage débranché en situation d'apprentissage est abordé dans la section [Main à la pâte](#).

Bien que les activités débranchées puissent servir d'amorce au codage, il est aussi important de passer à la machine pour que l'élève expérimente les limites de celle-ci, qui ne complètera pas l'information manquante.

L'expression **codage branché** est utilisée pour décrire le codage qui se fait à l'ordinateur. Il existe une énorme variété de logiciels et de langages avec lesquels du code pourrait être rédigé, comme nous l'avons mentionné dans l'introduction. Cela étant dit, on entend souvent parler de « codage par blocs », qui consiste à créer un code visuel à l'aide de blocs ayant différentes fonctions, et de « codage textuel », soit l'utilisation des caractères pour rédiger notre code. Les caractères peuvent être des mots, des abréviations ou des symboles qui sont reconnus par la langue de programmation en question.

Il est important de passer du **codage débranché** au **codage branché** afin de montrer la différence entre le respect d'une marche à suivre chez un être humain et chez un robot. L'être humain peut, par exemple, utiliser ses connaissances antérieures pour faire des prédictions et compléter des directives imprécises, tandis que l'ordinateur n'a pas cette capacité. Les directives seront respectées à la lettre. Cela peut parfois être source de frustration chez les élèves. On entend souvent des commentaires comme « Mon robot n'a pas fait ce que j'ai demandé! ». L'avantage de l'utilisation du codage pour résoudre des problèmes mathématiques est la rapidité avec laquelle l'ordinateur peut représenter et résoudre des problèmes complexes; par exemple, dans un problème de probabilité, plus on recueille des données, plus la probabilité expérimentale approche la probabilité théorique. L'ordinateur pourra générer le résultat d'un événement aléatoire des milliers de fois avant que l'être humain en génère une dizaine. Cette puissance ouvre la porte à la modélisation de situations complexes et à des simulations qui ne sont pas possibles avec le codage débranché.

Il n'existe pas de moment précis pour passer du codage débranché au codage branché. Il est même possible de passer d'un à l'autre de manière entrelacée tout le long de l'année scolaire. L'important, c'est de reconnaître l'aisance des élèves avec le codage dans des contextes mathématiques. Il est essentiel d'assigner des tâches les aidant à explorer davantage les logiciels de codage de façon délibérée, ciblée et bien échelonnée. Si, au cours d'une activité de codage, on observe que les élèves ont de la difficulté à représenter une situation mathématique donnée avec du code, rien n'empêche de retourner au codage débranché, de faire un pseudocode, de montrer les blocs ou les commandes nécessaires à la représentation de la situation, et de tenter le code de nouveau.

Le codage par blocs

Dans le contexte du codage et de la programmation, le codage par blocs est une idée très récente si l'on compare sa création dans les années 2000 avec celle de la première langue de codage en 1843! Le codage par blocs rend le codage accessible à une plus grande population malgré le fait qu'on l'utilise souvent comme une introduction avec nos élèves. Plusieurs adultes utilisent des plateformes de codage par blocs avec une langue de codage textuelle afin de générer des preuves de concept de manière rapide avant de passer au codage textuel.

Le codage par blocs offre plusieurs avantages à l'élève qui commence son parcours d'apprentissage en codage. Dans un **langage graphique à base de blocs**, l'élève peut assembler des séquences d'instructions à l'aide de blocs qui ont chacun une fonction. L'élève n'a pas à taper de longues phrases dans un langage qui semble inconnu. Chaque bloc ressemble à une pièce de casse-tête qui s'assemble aux autres. Afin de faciliter la compréhension des élèves, les blocs prennent des formes précises en fonction de ce qu'ils représentent; par exemple, une valeur numérique pourrait prendre une forme arrondie, alors qu'une valeur booléenne (valeur logique qui correspond à vrai ou à faux) prendrait la forme d'un hexagone.



On voit ici quelques exemples de blocs qui sont compatibles ou non les uns avec les autres.

Les blocs en forme de casse-tête peuvent être assemblés pour faire une séquence, mais il n'est pas possible d'attacher un bloc en forme d'ovale à la séquence, car les blocs en ovale représentent des quantités dans ce logiciel. Il ne serait donc pas logique de donner sans contexte deux directions suivies d'une quantité. Il est cependant possible d'ajouter un bloc ovale à l'intérieur d'un bloc qui contient une valeur (dans un ovale).

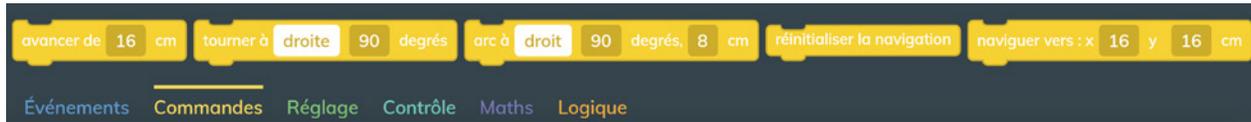
De même, il n'est pas possible d'ajouter une quantité (ovale) toute seule dans un bloc qui nécessite une valeur booléenne (hexagone), car la valeur booléenne doit correspondre à une question à laquelle on répond par oui ou non. Si l'on écrit la phrase dictée par le code, on voit bien cette logique. L'instruction « Si (ma variable) est plus grande que 50, alors... » a un sens logique.

Au cours du codage, si l'élève ne réussit pas à attacher un bloc à un autre, c'est parce qu'ils ne sont pas compatibles. Cette particularité du codage par blocs permet une diminution considérable d'erreurs de syntaxe, soit des erreurs liées à la structure même du code. De plus, lorsqu'il y a du texte associé aux blocs, le texte est en langage usuel et non en langage de programmation. Certains logiciels de codage par blocs offrent même la possibilité de convertir le code dans une langue de codage textuelle afin d'appuyer l'élève au cours du transfert du codage par blocs en langages comme Python et JavaScript.

Afin de répondre aux besoins des élèves, plusieurs langages offrent deux versions différentes : une avec des icônes (nécessitant peu ou pas de lecture) et une avec des descriptifs.

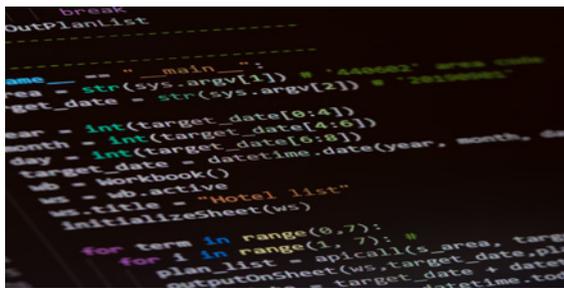


Exemple d'un logiciel de codage par blocs qui utilise des icônes



Exemple d'un logiciel de codage par blocs qui utilise du texte

Le codage textuel



Le **codage textuel** peut être vu comme la « prochaine étape » et est ce qui est utilisé dans l'industrie. Bien que ce type de codage puisse sembler intimidant au départ, plusieurs outils facilitent l'apprentissage. L'élève qui est en mesure de faire la transition du codage en blocs au codage textuel peut utiliser des plateformes de codage qui aident à faire la conversion en un seul clic. L'élève ne part pas de zéro. Il est possible de passer d'un mode à l'autre et d'apporter des modifications. En date de la rédaction de ce guide, les langages

populaires sont Python et JavaScript. L'évolution du domaine de l'informatique ainsi que la découverte de nouveaux problèmes à résoudre font en sorte que les langages de codage textuel évoluent aussi. Cela étant dit, l'élève découvrira rapidement que tous les langages se ressemblent et utilisent des structures semblables : des séquences, des boucles, des structures conditionnelles, des variables, etc. Il est aussi à noter que, dans la grande majorité des cas, les langages de codage textuel sont principalement basés sur de la terminologie anglophone. Les commentaires, en revanche, peuvent être rédigés dans n'importe quelle langue. Les commentaires sont abordés à la section [Communiquer, c'est essentiel!](#).



Le logiciel de codage MakeCode de Microsoft permet de basculer entre les blocs et Python en cliquant sur un bouton. Cela aide l'élève à voir, en temps réel, les équivalents textuels de ses blocs de code. Plusieurs logiciels de codage par blocs offrent cette fonction.

Développer la pensée computationnelle

Pour enseigner à une machine à accomplir une tâche, l'élève doit « traduire » sa pensée ou son intention dans un langage qui peut être compris par une puce informatique. Ce processus de conversion est possible grâce à la **pensée computationnelle** (aussi appelée *pensée algorithmique* ou *pensée informatique*). De la même façon que l'on fournit les outils aux élèves pour développer leur pensée algébrique, le raisonnement proportionnel et la pensée probabiliste, il est crucial de mettre en place des stratégies aidant l'élève à accroître sa pensée computationnelle en lui offrant des occasions de s'exercer. L'élève doit s'entraîner à « penser comme la machine ».

Il est essentiel que l'élève puisse reconnaître que les puces informatiques traitent l'information très différemment du cerveau humain. Le cerveau humain peut gérer des instructions qui sont floues en procédant par inférences. La machine, quant à elle, a besoin d'instructions très précises et exactes, y compris des valeurs numériques.

Par exemple, si l'on enseigne à une personne à porter une bouteille d'eau à sa bouche et à l'incliner pour boire de l'eau, des instructions extrêmement précises ne seraient pas nécessaires.

- Monter la bouteille d'eau.
- Approcher de la bouche.
- Incliner jusqu'à ce que l'eau coule dans la bouche.
- Remettre en position verticale lorsque la bouche est pleine.



L'être humain peut utiliser des inférences, des gestes et du modelage afin de faire comprendre la tâche à un autre être humain. Enseigner cette tâche à un robot est beaucoup plus ardu. Par exemple :



- Incliner le verre à un angle de 0° .
- Déplacer le verre selon l'axe des y sur une distance de 20 cm.
- Déplacer le verre selon l'axe des x sur une distance de 8 cm.
- Commencer à faire tourner le moteur dans le sens horaire pour incliner le verre jusqu'à un angle de 120° .
- Attendre 5 secondes.
- Commencer à faire tourner le moteur dans le sens antihoraire pour remettre le verre à un angle de 0° .
- Déplacer le verre selon l'axe des x sur une distance de - 8 cm.
- Déplacer le verre selon l'axe des y sur une distance de - 20 cm.

Cet exemple montre bien à quel point la rédaction d'instructions pour une machine doit être précise et détaillée. On fait aussi appel à plusieurs concepts de mesure, de géométrie et de sens du nombre. Selon l'intention et le contexte, l'élève pourrait faire un lien avec l'algèbre en utilisant des variables pour représenter les distances, le temps et les angles afin que le code puisse s'appliquer à différentes situations [par exemple, quelqu'un qui est plus grand, quelqu'un qui boit plus rapidement].

Les six processus mentaux

Le développement de la pensée computationnelle se fait au moyen de **six processus mentaux**.

PROCESSUS MENTAL	EXPLICATION
La décomposition	Un problème ou une tâche peut sembler difficile à résoudre. La décomposition consiste à reconnaître les différentes sous-tâches dans le problème et à morceler la tâche. L'élève peut donc résoudre son problème « en petites bouchées ».
La reconnaissance des régularités	Dans un problème, certaines actions ou certains éléments peuvent revenir plus d'une fois. L'élève doit donc les reconnaître et utiliser des structures de codage, comme la répétition, ou recycler son code.
L'utilisation d' algorithmes	La résolution de problèmes se fait à l'aide d'algorithmes; par exemple, l'élève peut reconnaître que, pour résoudre son problème, un algorithme de tri, un algorithme de comparaison ou un algorithme de déplacement est nécessaire. En concevant ces algorithmes, l'élève conçoit une liste pas-à-pas des étapes pour résoudre le problème. Remarque : Le personnel enseignant peut faire le lien avec les algorithmes utilisés pour apprendre les opérations.
L' abstraction	Certains problèmes comprennent des éléments superflus qui ne servent pas à résoudre le problème principal. L'élève doit donc abstraire l'essentiel du problème en laissant tomber les détails insignifiants. Ces détails peuvent être rajoutés après avoir résolu le problème.
La modélisation et la simulation	Il faut créer le modèle en convertissant l'algorithme en un langage compris par l'appareil, puis l'exécuter et simuler son fonctionnement. L'élève peut visualiser le fonctionnement de son code.
L' évaluation	Il faut évaluer si la solution répond au problème et déterminer s'il faut changer le code.

Remarque : Il est important de noter qu'il s'agit de processus et non d'étapes. Il n'y a pas d'ordre précis pour les accomplir.

Les concepts mathématiques et le codage

L'élève se rendra compte rapidement que les concepts mathématiques prennent vie et s'avèrent d'une très grande utilité. C'est une occasion parfaite de répondre à la fameuse question « Pourquoi apprend-on cela? ». Par exemple, l'élève qui fait avancer un robot en indiquant un nombre de rotations des roues pourra se poser la question « Qu'est-ce qui arrive si quatre tours de roue ne sont pas suffisants, mais cinq tours de roue, c'est trop? ». On se trouve donc devant un contexte idéal pour présenter les nombres décimaux. L'exploration pourrait se poursuivre à l'aide de la question suivante : « Qu'est-ce qui arrive si 4,5 tours de roue ne sont pas suffisants, mais 4,6 tours de roue, c'est trop? ». On se trouve maintenant dans un contexte idéal pour faire le passage de valeurs de position des dixièmes aux centièmes. Cet exemple donne un sens de la quantité que représentent un dixième et un centième dans un contexte réel et pertinent. L'élève pourrait aussi utiliser le raisonnement proportionnel : « Je crois que mon objet devrait se déplacer sur une distance au moins deux fois et demie plus grande. » L'élève est constamment dans un processus de résolution de problèmes.

Les processus mathématiques et les habiletés socioémotionnelles en codage



Il y a des preuves convaincantes que le développement des habiletés socioémotionnelles à l'école contribue à la santé générale des élèves et à leur bien-être, ainsi qu'à l'amélioration de leur rendement scolaire. Cela appuie la santé mentale et la capacité des élèves à apprendre, à améliorer leur résilience et à s'épanouir¹.



Comme tout autre aspect de l'enseignement, l'intégration du codage dans la salle de classe nécessite l'engagement du personnel enseignant ainsi que celui de l'élève afin que ce soit une réussite. Lorsqu'on parle d'engagement dans le contexte scolaire, on parle de la motivation à apprendre, du désir de participer, de connaître l'intention pédagogique derrière les diverses activités et de reconnaître et de croire en nos capacités. Comment peut-on alors créer un environnement propice à l'apprentissage lorsqu'on se trouve dans un domaine si vaste et, pour plusieurs, inconnu?

Au début, il est important de créer un espace sûr et de développer des relations de confiance avec les élèves. Les possibilités illimitées d'exploration à l'aide du codage sont à la fois un cadeau et une source possible de frustrations. Il importe de reconnaître les émotions que l'élève et l'adulte pourraient ressentir dans une situation où la réussite de la tâche ne viendra pas du premier coup. Ces émotions deviennent des occasions de développer des habiletés socioémotionnelles afin de développer une identité positive en tant qu'apprenantes et apprenants du codage. Une conversation intentionnelle sur les émotions ressenties lors de la tâche (la fierté, la confusion, le stress, l'inconfort, le doute, l'enthousiasme) permet de reconnaître et de valider les réalités du groupe-classe, y compris celles du personnel enseignant, qui est aussi placé dans une situation de nouvel apprentissage. Le modelage de la reconnaissance et de la gestion des émotions par un modèle accessible peut rendre la pratique plus facile à comprendre et à réaliser pour l'élève. C'est aussi une belle occasion de créer un sentiment de communauté. Le codage s'apprend en groupe, et il ne faut pas avoir une expertise préalable afin de profiter du codage comme outil dans des contextes mathématiques.

Par exemple, l'élève qui dit « je ne sais pas comment faire » ou « c'est trop difficile » vit peut-être une surcharge cognitive à cause de son état émotionnel élevé, ce qui rend l'apprentissage difficile, voire impossible. Le personnel enseignant peut commencer par reconnaître ses émotions et l'encourager à prendre un pas de recul. Lorsque l'élève montre une ouverture à l'apprentissage du codage, le personnel enseignant peut poser des questions sur ses connaissances antérieures afin de trouver un contexte semblable qui servira à rassurer l'élève sur sa capacité à réussir ou lui offrir des pistes d'exploration qui pourraient donner des résultats semblables. Le but est de placer l'élève dans une situation de réussite.

Le codage peut aussi servir à développer des habiletés telle que la persévérance. Au cours d'une activité de codage, l'élève n'a pas besoin de demander au personnel enseignant de valider son code. L'élève, voyant le dénouement à la suite de l'exécution du code, sait immédiatement si son code donne le résultat escompté. Par la suite, l'élève peut décider de procéder à des changements, soit en modifiant un code fonctionnel pour le rendre plus efficace ou plus précis, soit en « débogant » son code qui ne donne pas le résultat désiré. Le processus est très riche et très valorisant, aidant non seulement à l'élève à vivre les mathématiques qui l'entourent, mais aussi à développer ses habiletés socioémotionnelles dans un contexte d'apprentissage bienveillant et sécuritaire.

Il importe de faire prendre conscience à l'élève que, malgré le fait qu'enseigner des tâches aux machines peut sembler fastidieux, ces dernières présentent des avantages importants. Dans les chaînes de montage, elles peuvent répéter des tâches avec précision sans s'épuiser. Les ordinateurs peuvent aussi faire des millions de calculs par seconde sans faire d'erreurs (s'ils ont été bien programmés!).

1 Curriculum de l'Ontario. Programme-cadre de mathématiques, de la 1^{re} à la 8^e année ministère de l'Éducation de l'Ontario, 2020.

L'importance des erreurs

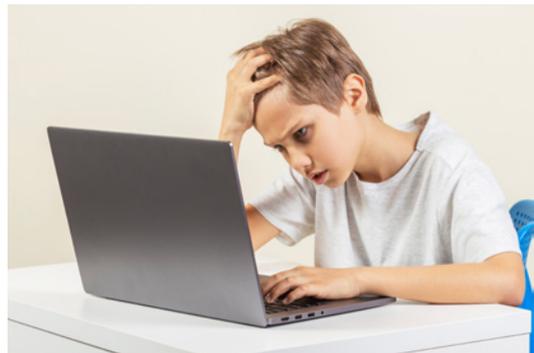
L'importance des erreurs est fondée sur les neurosciences. Il existe des recherches qui ont prouvé que c'est quand l'être humain se trompe que le cerveau « s'allume », où l'on voit des régions du cerveau qui sont actives². Dans certains cas, on voit même que les régions du cerveau responsables de la mémoire et de l'attention sont actives uniquement à la suite d'erreurs. C'est dans ces moments précis, où il y a erreur, que l'apprentissage est à son sommet, que l'élève est plus apte à se rappeler des concepts appris et à transférer l'apprentissage dans un nouveau contexte.

Il ne faut donc pas oublier de valoriser l'erreur en codage. Heureusement, par sa nature même, le codage s'apprend très bien en commettant des erreurs. Il y a peu ou pas de risque lorsque l'élève expérimente dans une séquence de code – la pire chose qui pourrait arriver, c'est que le code ne fonctionne pas. Il est donc important d'aborder le codage dans un contexte mathématique avec une perspective artistique, où la créativité et la prise de risques peuvent être appréciées et valorisées. Chaque erreur vécue par l'élève est un nouvel apprentissage. Même si le résultat n'est pas ce qui était prévu, il arrive souvent de découvrir la façon dont certains éléments de code interagissent en commettant des erreurs. Parfois, l'exploration des solutions possibles et l'élimination des erreurs importent plus que de connaître toutes les réponses.

L'erreur peut facilement être une cause de stress chez les élèves. Le personnel enseignant peut modéliser des situations où il n'a pas la réponse exacte à la question posée. Résoudre un problème ensemble avec du code peut montrer à quel point on peut se rendre loin sans nécessairement avoir toutes les connaissances en codage. Les élèves voient le personnel enseignant modifier le code, formuler des suggestions, effectuer des recherches dans Internet et même prendre des pauses si les frustrations montent. À la fin du processus, lorsque le résultat désiré est atteint, les élèves auront expérimenté un modelage sur les processus du codage. Il faut se rappeler que même les programmeuses et les programmeurs informatiques réussissent très rarement à rédiger un code parfaitement fonctionnel du premier coup!

Le débogage

Parfois, le processus de résolution de problèmes au cours du codage peut devenir frustrant pour les élèves, surtout lorsqu'il y a un bogue dans le code qui semble introuvable. Le terme *bogue* est utilisé en informatique pour faire référence à une erreur qui nuit au bon fonctionnement du code. Lorsque l'élève cherche les bogues pour les réparer, on utilise le terme « **débogage** ». Afin d'aider l'élève à développer des habiletés socioémotionnelles lorsque des difficultés sont rencontrées au cours du débogage, plusieurs stratégies peuvent être utilisées :



- Fermer le code et l'oublier pendant quelques minutes ou quelques heures. Au retour, l'élève verra des éléments non perçus lors du codage initial.
- Éliminer toutes les lignes de code non essentielles pour se concentrer sur l'algorithme principal.
- Demander à une autre personne de regarder le code pour obtenir un second avis.
- Utiliser le **pseudocode** avant de commencer à rédiger le code et s'y référer tout le long du processus de codage.
- Exécuter le code une ligne à la fois pour déterminer le moment où se produit le bogue. Certaines interfaces de codage offrent l'option « Suivi de l'exécution ». L'élève peut aussi verbaliser l'intention voulue de chaque bloc afin de trouver les incongruences.
- Inclure un fichier sonore ou un message à l'écran à un certain endroit dans le code pour vérifier si l'on atteint cette partie du code au cours de l'exécution.

2 Science of Learning Research Centre.

Le codage au service du développement des compétences transférables

Le codage peut aussi être une porte d'entrée au développement des compétences transférables. L'apprentissage du codage et l'apprentissage par le codage développent plusieurs compétences qui ne sont pas exclusives au codage. L'utilisation du codage dans des contextes mathématiques peut avoir un effet positif sur l'élève dans toutes les matières. Les deux exemples qui suivent portent sur les compétences de l'innovation, de la créativité et de l'entrepreneuriat, de la communication et de la collaboration, mais les sept compétences transférables sont présentes dans les exemples et les situations d'apprentissage tout le long du guide.

La créativité de l'élève est au volant!

Bien que plusieurs tâches présentées aux élèves comprennent une marche à suivre avec des paramètres bien précis, les tâches de codage aident toujours l'élève à laisser libre cours à sa créativité en modifiant le code, en y greffant de nouvelles fonctionnalités, en perfectionnant l'interface utilisateur, en bonifiant l'expérience utilisateur et en soignant l'apparence visuelle.

Par exemple, l'élève qui travaille sur un algorithme de conversion de devises pourrait ajouter des drapeaux pour rendre son programme plus attrayant, aller chercher des taux de change en temps réel, permettre à l'utilisatrice ou à l'utilisateur d'utiliser le glisser-déplacer pour en faciliter l'utilisation, ajouter des instructions et des effets sonores, ajouter, au besoin, de frais administratifs de transactions et ajouter de la synthèse vocale pour améliorer l'accessibilité. Bien que, dans cet exemple, les élèves aient reçu la même consigne (créer un algorithme de conversion de devises), les projets seront très diversifiés et témoigneront d'une créativité sans limites. En tant que membre du personnel enseignant, il ne faut pas brider cette créativité en limitant les élèves qui sortent des sentiers battus, mais plutôt la célébrer.



Le codage est aussi un bac à sable pour la prise de risques. Il faut encourager l'élève à altérer le code pour observer les résultats obtenus. En partant d'un code déjà existant, même s'il semble inintelligible au départ, l'élève peut y apporter de petits changements incrémentiels et constater les variations dans les résultats obtenus (par exemple, changer la valeur d'une variable, modifier la valeur seuil d'une comparaison dans une structure conditionnelle, changer le nombre de répétitions dans une boucle). Ce processus d'enquête l'aidera à approfondir sa compréhension du code et du langage de programmation utilisé.

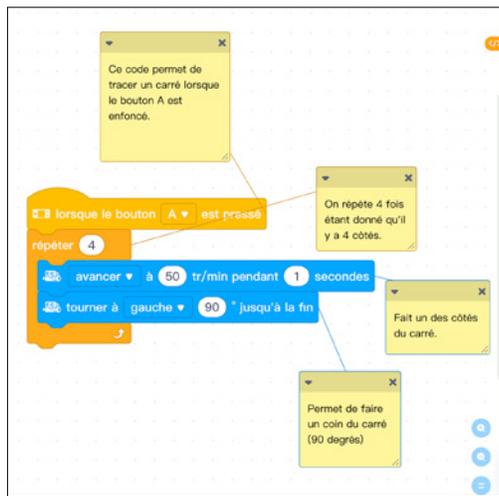
Un avantage important de la prise de risques en codage est l'absence de répercussions néfastes. Il n'est pas possible d'endommager l'appareil, le robot ou le microcontrôleur en changeant le code. Le pire qui puisse se produire, c'est que le code ne sera plus fonctionnel. Il est toujours possible de revenir en arrière! On devrait donc inciter l'élève à disséquer et à remanier son code.

Communiquer, c'est essentiel!

La communication est une compétence essentielle dans toutes les matières et elle l'est particulièrement en codage. Dans l'industrie, plusieurs programmeuses et programmeurs travaillent souvent sur un même projet. Lorsque le code devient très long, il peut être difficile de savoir ce que fait chaque ligne de code, même pour des expertes et des experts dans le domaine.

Tous les langages informatiques donnent la possibilité d'inclure des commentaires. Ces commentaires sont ignorés par l'appareil au cours de l'exécution du code, mais ils permettent aux êtres humains de mieux comprendre sa structure. Dans le cas du codage avec des blocs, les commentaires prennent souvent la forme de papillons adhésifs amovibles. Dans un langage textuel, un caractère est utilisé pour indiquer à l'appareil d'ignorer ce qui suit; par exemple, en Python, on utilise le symbole # pour indiquer un commentaire.

Cette documentation du code est non seulement utile à l'élève, qui comprend mieux son code lorsque celui-ci est revu, mais donne aussi l'occasion au personnel enseignant de recueillir des preuves d'apprentissage aux fins d'évaluation.



```
1 #Ce code permet de tracer un carré lorsque le bouton A est enfoncé.  
2 import mbot2, event, time, cyberpi  
3  
4 @event.is_press('a')  
5 def is_btn_press():  
6     # On répète 4 fois étant donné qu'il y a 4 côtés.  
7     for count in range(4):  
8         # Fait un des côtés du carré.  
9         mbot2.forward(50, 1)  
10        # Permet de faire un coin du carré (90 degrés)  
11        mbot2.turn(-90)  
12
```

2. COMMENT

Coder avec intention

L'intégration du codage dans la planification à long terme

Bien que l'on puisse intégrer le codage à diverses matières, comme la littérature, les arts, les sciences et la technologie, il est essentiel que l'intégration soit réalisée dans le but d'explorer, d'apprendre ou de consolider des concepts mathématiques.

La forme que le codage prendra dans une planification à long terme dépend de plusieurs facteurs. Heureusement, le codage peut s'insérer à n'importe quel moment d'une planification à long terme.

Par exemple, le codage débranché, soit le codage dans lequel l'élève joue le rôle de l'ordinateur, est une excellente porte d'entrée pour expliquer aux élèves la pensée computationnelle. Pour faire le pont vers le codage à l'ordinateur, une tâche de consolidation liée à un concept mathématique pourrait être présentée à l'aide d'un logiciel de codage par blocs. De cette façon, l'élève peut vraiment se concentrer sur l'exploration du logiciel, puisque le concept mathématique serait déjà acquis.

Avec de la pratique, l'aisance des élèves à représenter des situations mathématiques avec du codage augmente, et des situations mathématiques de plus en plus complexes peuvent être représentées avec du code. Puisque le codage est une porte d'entrée accessible aux élèves et à un plafond très élevé, donner aux élèves une seule occasion sous la forme d'une « unité » de codage présente un grand désavantage. Il importe de profiter de la polyvalence du codage aux fins d'apprentissage diversifié. L'élève peut même choisir d'utiliser le code pour représenter et résoudre un problème mathématique sans que ce soit exigé par le personnel enseignant, un peu comme l'élève qui choisit de se servir de matériel concret.

Les nombreuses possibilités d'intégration du codage dans le cours de mathématiques rendent le codage très utile dans un modèle de planification en spirale. Dans l'approche en spirale, il existe trois façons d'utiliser le codage.

1. Un même concept mathématique est représenté, à de multiples reprises dans l'année, à l'aide de codes de plus en plus complexes; par exemple, une série de déplacements peut être présentée en codage débranché, puis être transformée en pseudocode, en séquence de blocs ou en code textuel. Il serait aussi possible d'utiliser cette approche pour présenter des concepts précis de codage, comme des événements répétitifs dans une séquence, ensuite dans une boucle et, finalement, avec des variables pour représenter les valeurs dans le code.
2. Un logiciel de codage est utilisé pour représenter un concept mathématique qui devient de plus en plus complexe; par exemple, l'élève peut utiliser des instructions conditionnelles (si, alors, sinon) pour recueillir des entrées de l'utilisatrice ou de l'utilisateur (qui deviennent des données), les analyser sous forme de liste et calculer les mesures de tendances centrales.
3. Les connaissances et les habiletés en codage ainsi que les situations mathématiques deviennent de plus en plus complexes au cours de l'année; par exemple, utiliser des propriétés du rectangle pour rédiger un code, puis modifier les paramètres du code pour créer une variété de formes géométriques. De façon graduelle, le code pourrait aussi être utilisé pour situer, dans un plan cartésien, les sommets de la forme géométrique à l'aide de coordonnées précises, puis pour faire des transformations.

Heureusement, les logiciels et les langues de codage permettent la sauvegarde du travail accompli. Il est donc possible pour l'élève de consulter ou de modifier un code tout le long de l'année sans devoir le refaire complètement chaque fois. (À noter qu'afin d'avoir la possibilité de sauvegarder, certaines plateformes de codage nécessitent la création d'un compte. Il est tout d'abord primordial de s'assurer que le logiciel ou la plateforme respecte la gestion des données selon les directives administratives de votre conseil scolaire.)

Planifier une activité de codage

Comme toute activité pédagogique, la réussite d'une activité de codage repose sur son organisation et sur son intention. Ainsi, il ne faut pas négliger la planification de l'activité en codage. Cette planification peut contenir différentes étapes, comme le choix de la plateforme de codage ou l'apprentissage et le temps d'exploration de la plateforme choisie afin d'aider l'élève à développer ses connaissances et sa confiance. Il est primordial d'inclure les élèves dans l'élaboration d'une activité afin que les mathématiques ciblées soient pertinentes et intéressantes. Cela crée un lien plus étroit et durable entre l'élève et les mathématiques qui l'entourent, développe son sens de la représentation et, par conséquent, son sentiment de redevabilité.

Le point de départ pour la planification d'une activité de codage doit être l'intention pédagogique liée au programme-cadre de mathématiques. Il est important de bien cibler l'intention pédagogique pour que le codage serve réellement à représenter une situation mathématique.

Les résultats d'apprentissage sont au cœur de la planification d'une activité et découlent directement de l'intention pédagogique. On peut alors se demander : Qu'est-ce que l'élève devra être capable de faire à la fin de l'activité? En expliquant ces résultats d'apprentissage aux élèves, le personnel enseignant donne un sens à l'activité en guidant leur questionnement au moyen de pistes. Par sa nature même, le codage peut faire en sorte que les élèves expérimentent et s'éloignent du résultat désiré. Les résultats d'apprentissage peuvent alors guider l'élève vers l'apprentissage ciblé.

En plus de l'apprentissage du codage, il est important de donner aux élèves des occasions d'apprendre **avec** le codage. Dans le contexte d'un cours de mathématiques, cela veut dire qu'un concept ou qu'une situation mathématique est représentée avec du code, donnant à l'élève une nouvelle perspective des mathématiques. Il est aussi possible de demander à l'élève de lire du code déjà séquencé pour en prédire le résultat en utilisant ses connaissances antérieures en mathématiques.

La structure d'une activité de codage n'est pas différente d'une activité d'un autre domaine ou sujet. Il y a les trois mêmes étapes : la mise en situation, le déroulement et l'objectivation/la consolidation. Il faut se souvenir, cependant, qu'il existe des connaissances et des habiletés en codage ciblées pour chaque année d'études, et que **le codage devra être présenté conjointement avec un autre concept mathématique**. Une activité de codage peut donc avoir des résultats d'apprentissage et des critères d'évaluation liés aux connaissances et aux habiletés du codage ainsi qu'à un autre concept, à une autre situation ou à un autre domaine mathématique, soit celui que l'on modélise avec le codage.

Tout le long du processus, il est important pour le personnel enseignant de recueillir des preuves d'apprentissage en observant et en écoutant les échanges entre les élèves, en plus de recueillir des productions finales. [L'évaluation du codage](#) sera vue plus loin dans le guide.

Les pratiques pédagogiques à fort impact

Plusieurs façons existent de mettre à profit les pratiques pédagogiques à fort impact dans le contexte du codage. L'utilisation des pratiques pédagogiques à fort impact devrait aussi être intentionnelle et planifiée. Ce qui suit est un aperçu de l'utilisation des pratiques pédagogiques à fort impact dans des situations dans lesquelles le codage est utilisé pour présenter, mettre en pratique ou consolider un concept mathématique. La liste n'est cependant pas exhaustive, et le personnel enseignant doit choisir les pratiques qui s'arriment le mieux à l'intention pédagogique.

Résultats d'apprentissage, critères d'évaluation et rétroaction descriptive

Le codage, par sa nature même, est un sujet vaste et complexe. Cette complexité présente plusieurs avantages, mais deux grands désavantages :

- l'élève pourrait ne pas savoir par où commencer;
- l'élève est distraite ou distrait par les nombreuses possibilités et s'éloigne de l'intention.

Ces deux désavantages sont compensés par des résultats d'apprentissage clairs, ce qui aidera les élèves à connaître précisément les éléments d'une plateforme de codage qui sont à l'étude à un moment donné.



Les élèves tireront profit du fait que leurs enseignantes et enseignants collaborent avec elles et eux pour déterminer les critères d'évaluation qui seront utilisés tant dans le cadre de l'évaluation au service de l'apprentissage que dans le cadre de l'évaluation de l'apprentissage³.



Puisque le codage servira d'outil de représentation des mathématiques, les critères devront être précis, expliquer l'habileté en codage ciblé et indiquer la façon dont celle-ci s'applique au contexte mathématique à l'étude, lorsque c'est pertinent.

Par exemple, au lieu de dire « je peux utiliser une boucle pour répéter une séquence de code », on peut rendre le critère encore plus précis et ciblé en disant « je peux représenter une multiplication à l'aide de boucles ».

Il est à noter que la [rétroaction descriptive](#) est davantage expliquée dans la section du guide portant sur l'évaluation.

Enseignement explicite

L'enseignement explicite peut passer de l'information à un groupe d'élèves ou vérifier la compréhension d'une ou d'un élève à la fois. Les réponses des élèves deviennent des preuves d'apprentissage. Le personnel enseignant peut fournir de la rétroaction immédiate.

Tâches et expériences liées à la résolution de problèmes

Souvent connues comme « leçon en trois temps », « leçon de mathématiques en trois actes » ou « classe collabo-réflexive », elles ressemblent aux situations d'apprentissage présentées dans ce guide. Au départ, on active les connaissances antérieures des élèves, puis on les invite à choisir une stratégie efficace afin de montrer leur compréhension des concepts. En codage, il est intéressant de voir les appareils ou les logiciels que l'élève choisit pour effectuer son travail. Bien entendu, il faut avoir vu et exploré des outils de codage pour que l'élève soit en mesure d'en choisir un qui lui plaît et qui répond aux exigences de la tâche.

Enseignement quant à la résolution de problèmes

Il est très rare, même chez les professionnelles et les professionnels en codage, de créer une séquence de code parfaitement fonctionnelle du premier coup. La création du code nécessite des connaissances sur les fonctions des blocs ou des commandes, mais pour connaître la façon dont ceux-ci interagissent, il faut procéder par essais et erreurs. En essayant plusieurs combinaisons de blocs ou de commandes, l'élève crée une banque de possibilités à utiliser dans un prochain projet ou une future résolution de problèmes.



3 Faire croître le succès, p. 20.

En plus du fait qu'il faille procéder par essais et erreurs pour développer des habiletés en résolution de problèmes en codage, il faut aussi s'exercer à lire des séquences de code afin de repérer l'erreur. Ce processus, aussi nommé [débogage](#), nécessite une bonne compréhension de la façon dont les blocs ou les commandes interagissent et invite les élèves à prendre des risques et à commettre des erreurs dans la quête du « bogue ». L'élève doit analyser et comprendre le but du code et le rôle des séquences qui s'y trouvent. Chaque processus de débogage crée une nouvelle connaissance qui pourra être mobilisée au cours d'un prochain processus de débogage.

Outils et représentations

Le codage peut être un outil de représentation d'une situation mathématique. De plus, comme pour le matériel de manipulation, il existe plusieurs façons de représenter une même situation mathématique avec une séquence de code. Le personnel enseignant doit cibler des moments au cours desquels le sujet d'étude est propice à l'utilisation de codes afin de mettre les élèves dans une position de réussite.

Les outils et les représentations ne sont pas des réponses à des problèmes mathématiques, mais des facilitateurs du questionnement et de la réflexion. En utilisant le code comme outil, l'élève devra réfléchir à plusieurs éléments et prendre des décisions. Au fur et à mesure qu'il y a une progression dans ses habiletés en codage, l'élève pourra faire appel à des séquences plus complexes et à des blocs spécialisés pour rendre son code unique et représentatif de sa façon de penser.

Conversations mathématiques

Il est possible d'avoir des conversations mathématiques avec les élèves au cours de l'exploration d'un concept. Cela aidera le personnel enseignant à voir les connaissances acquises des élèves, la progression de leurs apprentissages ainsi que leur degré de confort avec les divers outils.

Questions possibles pour amorcer les conversations :

- Que remarques-tu?
- Quelles sont les ressemblances entre les codes? Quelles sont les différences?
- Comment voudrais-tu modifier le ou les codes?
- Quels blocs de code as-tu utilisés pour arriver à la solution?

Toutes ces questions ouvertes favorisent la discussion au sujet du codage et des mathématiques représentées par le codage. Cette conversation pourrait faire partie d'une mise en situation avant de lancer une activité de codage ou servir d'objectivation après l'exploration.

Enseignement en petits groupes

La nature accessible, mais à plafond élevé, du codage fait en sorte que des tâches de codage se prêtent très bien à l'enseignement en petits groupes. Pendant qu'un petit groupe reçoit de l'enseignement ciblé, la productivité du reste du groupe-classe peut se poursuivre. C'est ici que la variété d'approches pour résoudre un problème avec le codage peut être avantageuse. Les élèves ne recevant pas l'enseignement direct du personnel enseignant travaillent en vue d'atteindre les résultats d'apprentissage au moyen d'une approche. L'apprentissage se fait au cours de ce type d'expérimentation libre. Au cours de l'enseignement en petits groupes, les résultats d'apprentissage devront être fragmentés afin que les séances soient d'une courte durée, mais assez longues pour permettre l'exploration.

Pratique délibérée

Afin de s'améliorer, il faut s'exercer. La compréhension des concepts de base du codage est importante. L'apprentissage a lieu au cours de l'expérimentation avec les blocs ou les commandes à la disposition de l'élève.

Lorsque les élèves sont moins à l'aise avec le codage, l'approche du personnel enseignant peut ressembler à un modelage pour que le code des élèves soit fonctionnel. Ensuite, pendant l'expérimentation, les élèves peuvent changer, déplacer, supprimer et ajouter des blocs ou des commandes afin de découvrir les possibilités qui existent avec le logiciel ou la langue de codage en question. Cette expérimentation favorise l'apprentissage des concepts. Le personnel enseignant est à la disposition des élèves pour les appuyer et corriger des approches erronées.

Lorsque les élèves sont plus à l'aise avec le codage, le modelage initial peut être écourté ou éliminé. Les élèves peuvent ainsi expérimenter de façon indépendante une situation mathématique. Il s'agit maintenant de pratique délibérée, car les élèves développent leur aisance avec le codage en appliquant des concepts appris lors du modelage et de l'expérimentation.

Regroupements flexibles

Cette approche donne aux élèves l'occasion de travailler dans le groupement qui répond le mieux à l'intention pédagogique de l'activité pour leur permettre d'apprendre plus efficacement. Cette flexibilité se prête bien au codage, c'est même une méthode souvent employée dans l'industrie de la programmation.

Au cours d'une activité de codage, les regroupements pourraient différer selon l'intention ou la progression; par exemple, le personnel enseignant pourrait modéliser l'utilisation d'un code au groupe-classe. Après cette première exploration, les élèves travaillent individuellement afin de relever le défi assigné. Des équipes de deux sont créées de manière stratégique par le personnel enseignant afin que les élèves puissent vérifier leur code ou poser des questions. Ensuite, les élèves forment de petits groupes ou discutent en groupe-classe pour présenter leurs solutions.

Cette tâche pourrait aussi être réalisée avec des regroupements aléatoires qui donneraient l'occasion aux élèves de travailler avec de nouvelles personnes et les exposerait à différentes idées et façons de penser. Des regroupements hétérogènes ouvrent la porte à l'enseignement ciblé aux besoins particuliers des élèves. Ces regroupements aident le personnel enseignant à se déplacer plus librement dans la salle de classe pour observer, aider et vérifier la compréhension de ses élèves.

Des pratiques telles que des regroupements flexibles aident les élèves à discuter et à présenter leurs apprentissages et leurs découvertes tout le long de l'activité. Le personnel enseignant peut aussi adapter son enseignement et ses stratégies selon le regroupement d'élèves.

Comment choisir un logiciel de codage

Afin de rendre l'apprentissage par le codage accessible à toutes et à tous, il est important de choisir un logiciel de codage que les élèves pourront utiliser de manière indépendante. Un logiciel de codage par blocs visuels (par exemple, Scratch J, Hopscotch, code.org) ou des robots avec des fonctions limitées aux déplacements (par exemple, Bee-Bot) seraient idéals pour commencer.

Au fur et à mesure que l'élève développe ses habiletés en communication orale ainsi qu'en lecture, un plus grand éventail de logiciels et de plateformes devient accessible pour le codage par blocs (par exemple, Scratch, MakeCode) et la robotique (par exemple, Ozobot, Sphero, Cue). Des plateformes qui présentent un tutoriel (par exemple, Blockly, code.org) pourraient donner aux élèves l'occasion d'explorer les logiciels de codage à leur propre rythme avant de s'attaquer aux représentations de problèmes mathématiques plus complexes.

Le codage par blocs est un excellent moyen de commencer la représentation de problèmes plus complexes en offrant un peu plus de liberté et de choix aux élèves en ce qui concerne leur code, et limite le nombre et le type d'erreur à l'aide de formes et de codes de couleur. Plusieurs plateformes de robotique et des microcontrôleurs (par exemple, Micro:bit, Arduino, Adafruit) ont une structure de codage par blocs.

Pour les élèves qui peuvent faire le saut vers le codage textuel, il existe d'excellents logiciels intermédiaires entre le codage par blocs et les langues textuelles (par exemple, MakeCode, Swift Playgrounds). Ces logiciels peuvent rendre la transition plus facile en convertissant un code en blocs en un code en JavaScript ou en Python.

Au cycle intermédiaire, les logiciels de codage par blocs sont d'excellents outils afin de représenter des déplacements et des opérations de manière très visuelle, mais ne sont peut-être pas les meilleurs pour certains domaines, comme la littératie financière. Un chiffrier électronique (par exemple, Excel, Google Sheets) serait un meilleur outil pour effectuer des calculs liés aux budgets, bien qu'il soit possible de le faire avec un logiciel de codage par blocs. Le recueil de données peut aussi se faire avec un logiciel de codage par blocs, mais une plateforme dédiée à ce genre de recueil (par exemple, Google Forms, Mentimeter, Poll Everywhere) peut rendre l'analyse des données beaucoup plus efficace. Dans un logiciel de codage par blocs, par exemple, le calcul d'une médiane dans une liste de données nécessite plus de 10 lignes de code, tandis que, dans un tableur, il en faut juste une.

Le monde du codage est en constante évolution. Il est important de faire régulièrement une veille technologique afin de découvrir des logiciels ou des plateformes adaptés aux besoins des élèves et aux intentions pédagogiques ciblées.

Avant de choisir un logiciel, une plateforme ou un appareil de codage, il est important de réfléchir à l'intention pédagogique de son utilisation et le niveau actuel des élèves. Il faut d'abord se poser quelques questions.

- Est-ce que le logiciel, la plateforme ou l'appareil nécessite une approbation spéciale avant son achat ou son utilisation?
 - Est-ce que le logiciel, la plateforme ou l'appareil nécessite la création de comptes élèves?
 - Est-ce que le logiciel, la plateforme ou l'appareil a été approuvé par le conseil scolaire?
- Quelle est l'intention pédagogique de l'utilisation du codage?
 - Est-ce qu'un logiciel ou un appareil spécialisé est nécessaire afin de représenter les situations mathématiques?
 - Quelles activités peuvent être accomplies avec ce logiciel ou cet appareil?
- Quel est le degré de maîtrise actuel des élèves en codage?
 - Est-ce la première fois que le codage est abordé ou est-ce que les élèves ont déjà une base sur laquelle s'appuyer pour faire de nouvelles connaissances?
 - Est-ce que les élèves peuvent passer au codage textuel?
- Quel est le profil du groupe-classe? Est-ce que la tâche et le logiciel choisis sont accessibles?
- Y a-t-il un coût associé au logiciel, à la plateforme ou à l'appareil choisi?
 - Est-ce que cette dépense sera utile aux élèves d'autres années d'études?
- Le logiciel, la plateforme ou l'appareil permettra-t-il à l'élève de faire des choix?
 - Sera-t-il possible d'offrir des options de différenciation avec l'outil choisi?

Et l'évaluation?

Le codage et l'évaluation

La pensée computationnelle contribue à développer les compétences de résolution de problèmes, la logique et la créativité. Et la technologie est en train de transformer toutes les industries au monde. Les élèves d'aujourd'hui doivent apprendre à fabriquer la technologie, et non seulement à l'utiliser⁴.

Au cours de la planification d'une évaluation, il ne faut pas perdre de vue le but ultime de l'évaluation : améliorer l'apprentissage de l'élève. Bien qu'il soit possible d'évaluer l'apprentissage en se basant sur le produit final d'une activité de codage, une quantité importante d'information liée à l'apprentissage de l'élève peut être recueillie tout le long du processus. Les observations prises pendant l'activité de codage ainsi que les conversations entendues entre les élèves sont autant de preuves d'apprentissage. Ce qui ne change pas est l'importance de critères d'évaluation clairs, précis et pertinents, et l'engagement des élèves dans leur création afin de leur donner un sentiment d'appartenance et de redevabilité au travail à accomplir⁵.

Coder avec des blocs ou du texte exige de nombreux essais et erreurs, des traces qui se documentent difficilement. Toutefois, ce contexte permet au personnel enseignant de donner aux élèves une rétroaction descriptive et immédiate tout en étant positive autant que constructive. L'important est que la rétroaction porte sur une connaissance ou une habileté précise en codage et qu'elle soit liée aux critères d'évaluation. Cette rétroaction aide l'élève à s'ajuster et à cibler les prochaines étapes (ce qui peut aussi se faire en équipe avec ses pairs ou le personnel enseignant). La nature même du codage encourage le personnel enseignant à utiliser des repères visuels afin de connaître le moment et la façon de fournir de l'enseignement guidé aux élèves qui en ont besoin (par exemple, en créant des groupes de travail de manière stratégique ou en offrant un enseignement explicite d'un concept à tout le groupe-classe).

Au cours de l'évaluation en tant qu'apprentissage, l'élève devient « maître » de ses apprentissages avec autonomie en se fixant des objectifs personnels inspirés des résultats d'apprentissage et de la rétroaction du personnel enseignant ainsi que de ses pairs. La nature accessible, mais à plafond élevé, du codage incite les élèves à se fixer des défis ou des buts personnels selon leurs champs d'intérêt tout en restant fidèles aux résultats d'apprentissage et aux critères d'évaluation. La tâche de codage devient alors beaucoup plus que le fait de créer une série d'instructions. La tâche devient toute une expérience d'apprentissage permettant au personnel enseignant de recueillir des preuves et à l'élève de s'épanouir et d'explorer des sujets et des problèmes qui l'intéressent⁶.

Évaluer le codage et évaluer avec le codage

On évalue l'attente portant sur le codage lorsque les élèves font des activités de codage. L'élève lit, écrit, modifie et prédit l'effet de changements sur des codes pour montrer ses habiletés et ses connaissances en codage. Il ne faut pas oublier que le codage, dans le contexte du cours de mathématiques, doit être utilisé afin de représenter et de résoudre des situations mathématiques. Le codage devient pertinent lorsqu'il est arrimé à un autre concept mathématique, car le codage peut être difficile à évaluer seul.

Lorsque le codage est jumelé à un autre concept mathématique, il devient une forme de représentation, un peu comme le matériel de manipulation, afin d'aider l'élève à mieux comprendre les mathématiques; par exemple, la création d'une séquence de code ayant des éléments répétitifs, ou des boucles, pourrait être faite au moment de l'apprentissage de l'addition répétée ou des suites

4 [Hour of code.](#)

5 [Faire croître le succès](#), p. 48.

6 [Faire croître le succès](#), p. 39-40.

à motifs répétés. L'élève pourra alors utiliser le code dans un contexte d'évaluation afin de montrer sa compréhension de l'effet de la répétition sur la situation mathématique.

L'évaluation avec le codage exige que l'élève se serve du codage pour montrer ses compétences dans d'autres domaines mathématiques. On peut alors évaluer le codage et évaluer avec le codage de manière simultanée. Cela aura certainement une incidence sur la grille d'évaluation abordée ci-après.

Les critères d'évaluation et les quatre compétences

Puisque le personnel enseignant est appelé à évaluer des habiletés en codage ainsi que des habiletés d'autres concepts mathématiques, il est important que la grille d'évaluation soit claire et précise afin que l'élève connaisse les attentes liées à chaque domaine mathématique si plus d'un domaine est évalué.

Il existe trois scénarios possibles concernant l'évaluation du codage.

Dans le premier, seul le codage est évalué. Dans ce cas, les concepts mathématiques arrimés ne font pas l'objet d'une évaluation et ne font donc pas partie de la grille d'évaluation. La grille d'évaluation devrait contenir des critères qui portent uniquement sur les connaissances et les habiletés en codage.

Dans le deuxième, le codage et le concept mathématique arrimé feront l'objet d'une évaluation. La grille d'évaluation devra alors préciser si le critère est propre au codage, propre au concept mathématique ou propre aux deux.

Dans le dernier, le codage ne fait pas l'objet d'une évaluation. L'élève pourrait choisir de représenter une situation mathématique à l'aide de ses connaissances et de ses habiletés en codage. La grille d'évaluation se concentre plutôt sur le concept mathématique à l'étude.

À noter que, dans les trois scénarios, le codage est toujours utilisé dans un contexte mathématique.

Ce qui suit sont des exemples de compétences qui rendent possible l'évaluation du codage en mathématiques, rédigés sous la forme de critères d'évaluation. Certains critères exigent explicitement qu'ils puissent être évalués en parallèle avec un concept mathématique si cela répond à l'intention pédagogique. Il est aussi important de noter que la coconstruction des critères d'évaluation demeure une approche à prioriser et que les critères de la salle de classe pourraient différer de ceux-ci.

CONNAISSANCE ET COMPRÉHENSION

- J'**écris** et j'**exécute** des codes portant sur des situations mathématiques.
- J'**indique** les blocs (ou les commandes) et les actions initiées par chaque bloc (ou commande) relatifs aux situations mathématiques ciblées.
- Je **comprends** que je **représente** une série d'instructions à l'aide d'un algorithme.

HABILETÉS DE LA PENSÉE

Utilisation des habiletés de planification

- Je me sers du **pseudocode** afin de **planifier** la structure de mon code.
- Je peux **décomposer** mon code en sections ayant des résultats précis et **apporter des changements** pour satisfaire à une intention précise.

Utilisation des habiletés de traitement de l'information

- Je peux **déboguer** un code non fonctionnel envoyé par le personnel enseignant ou mes pairs.
 - Je peux **repérer** les erreurs de **logique** (le code fonctionne, mais ne donne pas le résultat prévu) et les erreurs de **syntaxe** (le code ne fonctionne pas du tout).
-

Utilisation des processus de la pensée critique et de la pensée créative

- Je peux **vérifier** la structure et la fonctionnalité de mon code.
- Je peux **critiquer** de façon constructive un code afin de déterminer différentes façons d'arriver au même but.

COMMUNICATION

Expression et organisation des idées et de l'information

- J'**exprime** et j'**organise** les idées et l'information de manière à faciliter la lecture d'un code ou d'un pseudocode.
 - Je **respecte** la syntaxe de la langue de programmation à l'étude.
 - Lorsqu'un code est modifié, je peux **décrire** l'incidence des changements sur les résultats.
-

Communication des idées et de l'information de façon orale, écrite ou visuelle à des fins précises et pour des auditoires précis

- J'utilise des **notes**, des **commentaires** ou des **éléments multimédias** (sons, vidéos) dans mon code pour faciliter son interprétation et sa révision.
-

Utilisation des conventions et de la terminologie à l'étude

- J'utilise la **terminologie** associée aux connaissances et aux habiletés en codage (par exemple, événements séquentiels, événements simultanés, événements répétitifs, événements imbriqués, structure de contrôle, blocs opérateurs, instructions conditionnelles) pour exprimer mes idées.

MISE EN APPLICATION

Application des connaissances et des habiletés dans des contextes familiers

- Je peux **lire** un code, puis le **modifier** afin de l'adapter à une situation mathématique précise.
-

Transfert des connaissances et des habiletés à de nouveaux contextes

- Je peux rédiger une séquence de code qui représente une situation mathématique précise **en partant** du langage familier, du pseudocode ou d'une autre langue de programmation.
-

Établissement de liens

- Je peux indiquer des problèmes mathématiques **de vie réelle** ou portant sur **mes champs d'intérêt**, qui pourraient être résolus par le codage.
 - Je peux utiliser le codage afin de représenter **une nouvelle** situation mathématique.
-

Ces exemples de critères ne sont pas adaptés à toutes les situations. Il est possible qu'en début d'apprentissage d'une nouvelle langue de programmation (textuelle ou par blocs) l'accent soit mis sur le respect de la syntaxe et la reconnaissance de blocs ou de commandes et leurs fonctions. Au fur et à mesure que l'élève progresse dans son apprentissage du codage, la cible de l'évaluation devrait aussi progresser.

Grâce au codage, il est aussi possible d'évaluer plusieurs domaines en même temps. Prenons, par exemple, une activité dans laquelle l'élève doit rédiger un code afin de dessiner un rectangle et de déterminer son aire et son périmètre. Dans cette activité, nous trouvons des habiletés en codage (Algèbre), en mesure (Sens de l'espace) et liées aux opérations (Nombres). Le codage devient alors un outil de représentation puissant rendant possible d'autres concepts mathématiques⁷.

Comment recueillir des preuves d'apprentissage

Lorsqu'on évalue le codage, tout le processus peut être évalué, du départ où l'écran est vide jusqu'au produit final. Le personnel enseignant peut recueillir des preuves d'apprentissage par triangulation liées aux critères d'évaluation. L'élève devra avoir plusieurs occasions de recevoir de la rétroaction et de s'ajuster en conséquence. Le personnel enseignant devra recueillir des preuves d'apprentissage de plusieurs sources et à plusieurs moments tout le long de la séquence d'apprentissage. Les trois sommets de la triangulation, soit les observations, les conversations et les productions, peuvent tous être évalués de différentes façons. Voici quelques exemples.

Observations

Tout le long du processus de codage, l'élève devra mettre en application ses apprentissages et en faire de nouveaux. Il importe que le personnel enseignant soit présent tout le long du processus afin de tenir compte de la mise en application des connaissances antérieures et de l'application ou du transfert des nouvelles connaissances et habiletés. De plus, les stratégies employées afin d'explorer et de déboguer le code sont d'excellentes sources de preuves que l'élève apprend activement. Le personnel enseignant peut utiliser des outils tels qu'une grille d'observation ou des photos ou des vidéos prises lors de la conception du projet. La grille d'observation doit cibler des apprentissages inspirés par les critères d'évaluation. Avec une grille, il est possible de noter des observations de manière rapide, efficace et continue.

Exemples de grilles d'observation

Une première grille d'observation pourrait utiliser un code de couleur afin de déterminer le degré d'atteinte des critères en question. La grille pourrait être faite à l'ordinateur ou sur papier en utilisant des surligneurs pour faciliter les observations. Dans l'exemple qui suit, on utilise le vert pour indiquer qu'un critère est atteint, le jaune pour indiquer que l'élève est sur la bonne piste et le rouge pour indiquer que l'élève devra continuer à s'exercer afin d'atteindre le critère. Les commentaires serviront d'inspiration pour la rétroaction descriptive qui sera donnée à l'élève relativement aux connaissances et aux habiletés observées.

Les critères pour la tâche pourrait être les suivants :

- Je peux planifier une séquence de code en utilisant des schémas ou du pseudocode.
- J'utilise correctement une variété de blocs dans mon code afin d'atteindre un but précis.
- Je peux expliquer la raison des différences entre mon pseudocode et mon code actuel.

7 Formation CFORP Le codage, oui, mais comment l'évaluer?.

NOM DE L'ÉLÈVE	PLANIFICATION DE MON CODE	APPLICATION DE MON CODE	JUSTIFICATION DE MON CODE	COMMENTAIRE

Une deuxième grille d'évaluation mise sur des connaissances et des habiletés précises et observables liées aux critères d'évaluation établis. Le personnel enseignant utilise des symboles pour prendre en note les observations; par exemple, la lettre encadrée indique l'acquisition de l'habileté, la lettre rayée indique que l'habileté est en voie d'acquisition et une croix (X) sur la lettre indique que l'habileté n'est pas acquise.

Les critères d'évaluation pour la tâche pourraient être les suivants :

- Je modifie mes codes pour corriger des erreurs de syntaxe ou de logique.
- Je crée des codes pour recueillir des données et les mettre dans une liste.

On pourrait utiliser les lettres D (débugage), É (écriture de codes) et L (liste de données).

Nom de l'élève ↓	Date →	
		D É L

En circulant parmi les élèves, le personnel enseignant aura la tâche d'observer le degré auquel les élèves appliquent le processus de débogage (par exemple, en fragmentant le code en plus petites séquences afin de trouver une erreur plus facilement), utilisent une variété de blocs pour rendre leur code plus efficace (par exemple, en créant des boucles pour des événements qui se répètent) et utilisent des commentaires ou un journal de bord pour noter les différences entre leur pseudocode et leur code actuel, un peu comme un brouillon avant de publier un texte.

Conversations

Au cours de la création d'un code, la conversation vient jouer un grand rôle. Puisque les élèves vont souvent utiliser des stratégies telles que l'essai et l'erreur ou la rétroaction des pairs, il est important de tenir des conversations liées aux problématiques. Ces conversations sont souvent instiguées par un questionnement.

Exemple



- Peux-tu m'expliquer ce que fera le *sprite*?
- Pourquoi as-tu choisi de mettre les blocs dans cet ordre? Pourrais-tu trouver une autre façon de déplacer ton *sprite* au même endroit?
- Je remarque que ton *sprite* fait un pas vers la droite, un pas vers le haut, un pas vers la droite, un pas vers le haut... Y aurait-il une façon de reproduire ce même mouvement?

Les conversations peuvent se faire entre les élèves, entre le personnel enseignant et l'élève, ou entre le personnel enseignant et un petit groupe d'élèves. Il importe de prendre le temps de planifier des moments à l'horaire pour converser avec les élèves ou écouter les élèves lors de conversations. En conversant, l'élève explique et approfondit des concepts ou des raisonnements mathématiques.

Pistes de questionnement

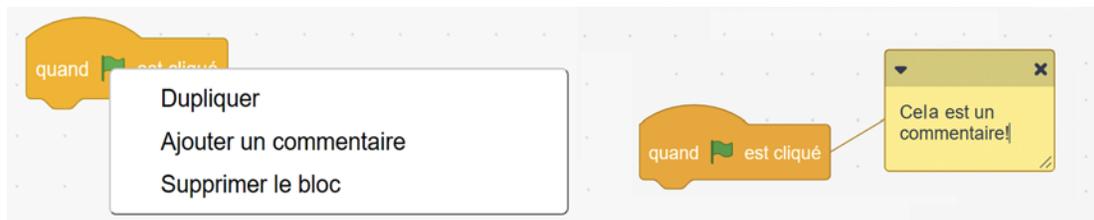
- Quels mots emploierais-tu pour expliquer ce que tu devais faire dans cette tâche?
- Que pourrais-tu faire de différent?
- Quelle démarche as-tu utilisée?
- Comment le sais-tu?

Le codage est un sport d'équipe, autant dans la salle de classe que dans le monde de la programmation professionnelle. Il est important de montrer son code à des collègues (ou, dans ce cas, à d'autres élèves du groupe-classe) afin de recevoir de la rétroaction et des suggestions constructives. Une écoute active entre les élèves permettra de reconnaître les rétroactions qui montrent une bonne compréhension des concepts. Cela étant dit, dès que l'élève est en mesure de communiquer aisément de façon écrite, les commentaires dans le code sont aussi une belle façon de recueillir des preuves d'apprentissage, car ce sont des conversations entre la personne qui a rédigé le code et la personne qui le lira. Les commentaires, dans un code, peuvent se faire de différentes façons.

Voici deux exemples courants d'un logiciel de codage par blocs (Scratch) et d'une langue de programmation textuelle (Python).

Codage par blocs :

Un clic droit sur le bloc que nous voulons commenter affichera un onglet dans lequel se trouve l'option **Ajouter un commentaire**, qui ressemble à un papillon adhésif.



Python :

Le croisillon (#) suivi d'un espace signale le début d'un commentaire. Chaque ligne du commentaire doit commencer par ces deux caractères. Les commentaires sont ignorés lorsque le code est compilé.

```
# La prochaine ligne de code imprimera à l'écran «Bonjour, le monde!»  
print(« Bonjour, le monde! »)
```

Productions

On sous-entend souvent que la production est le produit final, mais ce n'est pas toujours le cas. Une production, dans le contexte du codage, peut aussi être le pseudocode rédigé avant le début d'une tâche de codage, ou un journal de bord qui explique les défis rencontrés en codant et les stratégies de débogage qui ont été appliquées. Ce type de preuve d'apprentissage aide à évaluer le processus de codage en plus du produit final.

La production aide l'élève à appliquer ses connaissances antérieures et acquises, et à montrer sa maîtrise des résultats d'apprentissage. L'évaluation du produit final doit mettre l'accent sur le code et sur la manière dont le code répond aux critères d'évaluation. Il est parfois facile de perdre de vue des critères d'évaluation lorsqu'un code semble très complexe ou dépasse le degré de complexité recherché. Dans ce cas, il est important de revenir aux critères afin d'offrir une évaluation juste et objective.

Avec l'autoévaluation liée aux critères d'évaluation en plus de son code, l'élève obtient essentiellement une liste de vérification l'aidant à réviser et à réajuster son code en fonction des critères. Tout le code, y compris les commentaires, devrait être remis comme production, que ce soit en utilisant une fonction de partage par lien, une capture d'écran ou, dans le cas de code textuel, une transcription du code dans un logiciel de traitement de texte.

Finalement, un élément clé dans l'évaluation de la production est le choix de l'outil par rapport aux critères d'évaluation. Le choix d'outils pour créer un code doit être diversifié et accessible afin que l'élève choisisse l'outil qui puisse créer un produit qui met en valeur les résultats ciblés. Il est aussi important de différencier le type de production pour un même groupe d'élèves. En atteignant les résultats d'apprentissage avec du pseudocode descriptif, en créant une séquence dans un logiciel de codage par blocs ou en programmant un robot, l'élève a davantage de choix et, par conséquent, pourra trouver l'approche ou l'outil qui répondra le mieux à ses besoins. La différenciation des productions aide aussi à valoriser les différences, et encourage les élèves à apprendre de leurs pairs en observant les productions des autres.

La rétroaction



Lorsqu'il y a de multiples occasions de rétroaction et de suivi, toutes et tous les élèves acquièrent des habiletés pour évaluer leur propre apprentissage au fur et à mesure qu'elles et ils réfléchissent aux critères d'évaluation⁸.



La rétroaction est une partie intégrale de l'apprentissage. Pour qu'une rétroaction soit réussie, elle doit être utile, précise et bienveillante. Le dialogue avec les élèves doit être présent, à plusieurs reprises, dans la situation d'apprentissage. Il faut permettre à l'élève de s'ajuster et d'apporter des modifications, si cela est nécessaire, à la suite d'une rétroaction de la part de ses pairs ou de la part du personnel enseignant. Comme dans toutes les autres matières, la rétroaction en codage doit être basée sur les critères d'évaluation et doit se faire en collaboration avec l'élève. Garder en tête que l'élève participe à la rétroaction.

8 Source : *Pratiques pédagogiques à fort impact en mathématiques*, p. 7.

La rétroaction descriptive

La rétroaction, qu'elle soit du personnel enseignant ou d'un pair, est essentielle en codage. La rétroaction aide l'élève non seulement à sécuriser dans ses apprentissages, mais aussi à se sentir en confiance avec les divers outils. Une rétroaction descriptive tout le long du processus aide l'élève à atteindre les résultats ciblés tout en solidifiant ses apprentissages.

Une certaine précision est nécessaire dans l'information fournie aux élèves pour que ce soit de la rétroaction descriptive. Par exemple :

AU LIEU DE...	ESSAYER
Beau travail!	Tu as bien utilisé une grande variété de blocs dans ton code!
Ton code est très complexe.	Tu as créé une séquence complexe de blocs pour arriver à ton but. Crois-tu pouvoir atteindre ton but avec moins de blocs?
Tu n'as pas tout à fait réussi la tâche. Essaie encore!	Pourrais-tu revoir l'ordre des commandes dans ton code afin qu'elles suivent bien la syntaxe de la langue de programmation?

Ce ne sont que des exemples génériques. Dans la classe, le personnel enseignant devra fournir de la rétroaction qui porte sur l'atteinte des critères d'évaluation.

La rétroaction par questionnement

Le questionnement et la rétroaction descriptive ne sont pas mutuellement exclusifs. Au contraire, le questionnement peut être la première étape vers une rétroaction encore plus descriptive et ciblée en incitant l'élève à prendre part au processus de rétroaction. Celle-ci force l'élève à réfléchir au but et à la méthode au lieu de tout simplement créer du code et attendre les prochaines étapes. Cette approche a aussi l'avantage de permettre au personnel enseignant de confirmer l'acquisition des résultats d'apprentissage ciblés.

Voici quelques exemples de questions génériques qui pourraient mener à une conversation et à une rétroaction très ciblée. Le personnel enseignant a la tâche d'adapter le questionnement aux critères d'évaluation ciblés.

- Pourrais-tu me montrer la façon dont tu as créé ce code?
- Je vois que tu as utilisé les blocs « variables ». Quel effet ces blocs ont-ils eu sur ton code?
- Puis-je te suggérer quelque chose? Je vois que tu as ajouté le bloc « mouvement » plusieurs fois. Connais-tu un bloc qui permet de répéter un mouvement plusieurs fois? Montre-moi la façon dont tu peux modifier ton code.
- Pourrais-tu me montrer l'aide-mémoire des blocs de code et leur fonction? J'aimerais que l'on vérifie quelque chose ensemble.

L'évaluation par les pairs

Ce n'est pas seulement le personnel enseignant qui peut fournir de la rétroaction. Au contraire, la rétroaction au moment d'une évaluation par les pairs peut être encore plus puissante que celle du personnel enseignant. L'évaluation par les pairs n'est pas un outil pour influencer la note finale de l'élève sur la grille d'évaluation, mais est plutôt une occasion pour les élèves de présenter leurs accomplissements, de se poser des questions et d'offrir des critiques constructives ou des suggestions. L'évaluation par les pairs ne devrait donc pas être réservée au produit final, mais devrait plutôt être faite à de multiples endroits tout le long du processus de codage.

La modélisation de cette pratique en début d'année est primordiale à sa bonne utilisation, et les élèves ont besoin d'occasions pour s'exercer à jouer ce rôle au sein du groupe-classe. Les élèves pourront développer leurs compétences afin de donner une rétroaction qui aidera leurs pairs, par exemple, à créer un code utile et efficace. Cette rétroaction leur donnera aussi le vocabulaire et l'expérience pour s'évaluer plus efficacement. L'évaluation par les pairs est aussi l'occasion de fournir des preuves d'apprentissage sous forme de conversations.

3. QUOI

La main à la pâte

À quoi peut ressembler le codage?

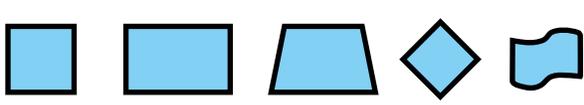
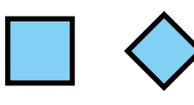
Plusieurs pensent encore que le codage est représenté par des écrans d'ordinateur recouverts de texte et de symboles incompréhensibles, ressemblant même parfois à une langue étrangère! C'est une image très commune, mais le codage peut prendre plusieurs formes, et certains éléments du codage sont même plus faciles à aborder sans la présence d'un ordinateur!

Les prochaines sections abordent une progression possible de l'utilisation des différentes formes de codage.

Le codage débranché

Comme il est mentionné dans la première partie, le **codage débranché**, ou sans ordinateur, est un outil très puissant qui aide l'élève à mieux comprendre la façon dont un ordinateur traite l'information. Ce type de codage ressemble énormément à une marche à suivre – une liste séquentielle de directives, commençant par des verbes à l'impératif ou à l'infinitif, dont le but est clair. D'abord, il est possible de demander aux élèves de « penser comme un ordinateur », c'est-à-dire de respecter la marche à suivre mot à mot, sans faire de suppositions et sans utiliser leurs connaissances antérieures.

Par exemple, en demandant de dessiner un carré, il est possible pour les élèves d'accéder à leurs connaissances antérieures, de se souvenir des propriétés du carré, et ensuite de le tracer sur le papier. Cependant, si les élèves doivent décrire à une personne la façon de dessiner un carré, la tâche devient intéressante – surtout si la personne, qui agit comme « ordinateur » dans cette activité, suit les directives à la lettre sans faire de suppositions!

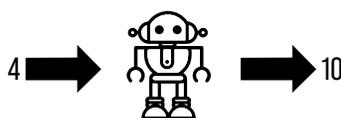
EXEMPLE DE DIRECTIVE DONNÉE PAR L'ÉLÈVE (ENTRÉE)	EXEMPLES DE DESSINS DE LA PERSONNE (SORTIES)
« Dessine une forme avec 4 côtés. »	
« Dessine une forme avec 4 côtés égaux. »	
« Dessine une forme avec 4 côtés égaux et 4 angles droits. »	

Dans cet exemple, le nombre de possibilités de « SORTIES » diminue avec la complexité des directives. Au cours d'une activité de codage débranché comme celle-ci, il est possible d'observer et d'écouter les conversations afin de recueillir des preuves de la compréhension conceptuelle de l'élève. L'élève qui aura simplement mémorisé la forme du carré ou qui n'a pas bien saisi le vocabulaire mathématique associé aura de la difficulté à fournir des directives claires.

Le codage débranché peut aussi servir à présenter ou à approfondir des connaissances dans d'autres domaines. Le robot algébrique, par exemple, aide les élèves à imaginer le « code » qui transforme un nombre en un autre nombre.

Par exemple :

Les élèves fournissent les nombres 4, 9 et 3. Le personnel enseignant, qui connaît le « code » recherché, écrit ceci :



Après la première transformation du robot, les élèves tentent de trouver les codes possibles pour donner une telle réponse.

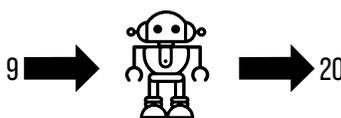
Exemples de codes possibles :

« Le robot additionne 6 au nombre. »

« Le robot multiplie le nombre par 2, puis additionne 2 à la réponse. »

« Le robot multiplie le nombre par 2,5. »

Ensuite, le robot peut transformer le prochain nombre.



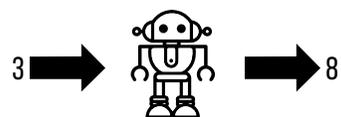
Les élèves peuvent maintenant vérifier les codes pour voir lesquels donnent toujours une bonne réponse, ou même créer de nouveaux codes, s'il y a lieu.

Exemples de réflexions :

« Le code qui additionne 6 au nombre de départ ne fonctionne pas, car $9 + 6$ ne fait pas 20. »

« Le robot multiplie le nombre par 2, puis additionne 2 à la réponse. »

Ensuite, il est possible de passer au prochain nombre.



Exemple de conclusion :

« Le code où le robot multiplie le nombre par 2 et additionne 2 à la réponse fonctionne pour les trois cas. »

S'il n'y a pas consensus sur le code, les élèves fournissent de nouveau des entrées pour que le robot puisse les transformer jusqu'à ce qu'il y ait consensus sur le « code » secret. Outre qu'elle donne l'occasion de travailler les équations et les inégalités, cette activité aide les élèves à voir qu'il est possible d'avoir plus d'une bonne réponse au cours de l'écriture d'un code, et que la méthode essai-erreur est un élément important du codage.

Les activités débranchées peuvent être enrichissantes et valorisantes en soi, mais afin d'explorer la puissance du codage dans un contexte mathématique, il faut passer du codage débranché au codage branché. Les activités débranchées sont un prérequis au codage sur ordinateur.

Le pseudocode

Le **pseudocode** peut agir comme intermédiaire efficace entre le codage débranché et le codage branché. Puisqu'il n'y a aucun lien avec une langue ou un logiciel de programmation spécifique, la rédaction d'un pseudocode aide l'élève à décrire les actions à effectuer par l'ordinateur et le résultat désiré. L'élève peut donc utiliser du langage familier et la terminologie à l'étude afin de créer un plan pour son code. Cette étape est comparable au brouillon dans le processus d'écriture – les grandes idées demeurent stables, mais il y aura plusieurs modifications entre le pseudocode et la séquence de code finale.

Par exemple, un groupe d'élèves travaille les concepts d'aire et de périmètre, et veut créer un code qui permettra à un *sprite* de se déplacer sur une trajectoire rectangulaire, puis dire l'aire et le périmètre du rectangle. Voici à quoi le pseudocode ressemblerait :

<i>Sprite</i> position = coin inférieur gauche orientation = vers la droite Variables [base] = base du rectangle [hauteur] = hauteur du rectangle [aire] = aire du rectangle [périmètre] = périmètre du rectangle
Condition de départ = vrai
Demander : « <i>Quelle est la base du rectangle?</i> » Valeur de [base] = réponse
Demander : « <i>Quelle est la hauteur du rectangle?</i> » Valeur de [hauteur] = réponse
Valeur de [aire] = [base] * [hauteur]
Valeur de [périmètre] = [base] + [hauteur] + [base] + [hauteur]
Répéter 2x
Déplacer - [base] pas vers l'avant
Rotation - ¼ tour sens horaire
Déplacer - [hauteur] pas vers l'avant
Rotation - ¼ tour sens horaire
Dire : « L'aire du rectangle est de [aire] . »
Dire : « Le périmètre du rectangle est de [périmètre] . »
Fin

L'emplacement du texte dans le pseudocode nous aide à suivre la séquence de directives; par exemple, les directives à répéter sous la mention « répéter 2x » sont davantage en retrait que la mention elle-même. Les élèves développeront leurs propres manières de rédiger du pseudocode avec des abréviations et des symboles qui leur sont propres. Cette flexibilité rend le codage plus accessible aux élèves au cours de l'apprentissage de l'utilisation d'un logiciel de codage. Le pseudocode peut aussi servir de preuve d'apprentissage, montrant ce que l'élève comprend de la situation mathématique à coder et de la pensée computationnelle, même si l'élève n'est pas encore habile avec un logiciel de codage.

Le codage dans un contexte mathématique

Une intention mathématique claire et précise devrait être le point de départ d'une aventure en codage. Un enseignement efficace du codage est axé sur la **résolution de problèmes** et sur la **représentation de situations mathématiques de façon computationnelle**. Ces deux composantes sont les bases de l'utilisation du codage tout le long des cycles primaire, moyen et intermédiaire. Les habiletés et les concepts qui y sont liés fourniront, pour leur part, une progression des apprentissages.

Il est primordial que l'élève apprenne à utiliser le code dans des contextes mathématiques afin de satisfaire à cette attente. Or, on doit se souvenir que l'utilisation d'un logiciel de codage en soi ne répond pas nécessairement à cette attente si le code écrit, exécuté ou modifié n'est pas directement lié à un contexte mathématique.

Par exemple, les logiciels de codage sont souvent utilisés pour animer un personnage, couramment nommé *sprite*, en utilisant des commandes directionnelles. Le pseudocode et le code en blocs suivants pourraient montrer le mouvement d'un chat sur une piste rectangulaire.

<i>Sprite « chat »</i>
position = centre
orientation = droite
Condition de départ = vrai
Déplacer - 5 pas vers l'avant
Rotation - ¼ tour sens horaire
Déplacer - 2 pas vers l'avant
Rotation - ¼ tour sens horaire
Déplacer - 5 pas vers l'avant
Rotation - ¼ tour sens horaire
Déplacer - 2 pas vers l'avant



L'élève qui crée ce pseudocode, ou un code semblable dans un logiciel de codage, montre certainement des habiletés en codage, mais le contexte dans lequel celles-ci ont été appliquées n'est pas nécessairement un contexte de résolution de problèmes ou de représentation mathématique. Afin de rendre cette situation pertinente au programme-cadre de mathématiques, il faut d'abord consulter d'autres domaines afin de déterminer la façon dont le codage peut rendre l'apprentissage ou la mise en application d'un concept mathématique plus pertinent et accessible. Le codage devient d'abord un outil de représentation et de résolution de problèmes mathématiques, et ne peut pas être enseigné et évalué seul.

Les habiletés liées au codage

Les habiletés en codage peuvent se résumer en deux grands thèmes : la **rédaction** d'un code et la **lecture** de celui-ci. Une ressemblance avec les pratiques gagnantes en littérature est à noter, dans le sens où l'apprentissage de l'écriture et de la lecture se fait plus efficacement de manière entrelacée⁹. Nous n'avons donc pas à enseigner une langue ou un logiciel de code en entier avant de pouvoir profiter des occasions d'apprentissage initiées par le codage; par exemple, l'élève pourrait recevoir un bout de code afin d'observer le résultat de son exécution, puis **apporter des modifications** à certains endroits pour le personnaliser. L'élève pourrait aussi recevoir des codes qui contiennent des erreurs et tenter de **prédire le résultat** de son exécution et essayer de le corriger (autrement nommé *débogage*). Dans ces deux exemples, les élèves font la **lecture et l'écriture de codes** de manière entrelacée, sans nécessairement avoir besoin d'une connaissance compréhensive de toutes les modalités et de toutes les fonctionnalités d'un logiciel ou d'une langue de codage.

Résoudre des problèmes avec du code nécessite l'activation de connaissances antérieures chez l'élève, ainsi qu'une quantité non négligeable de débrouillardise et de curiosité. Par sa nature même, le codage se prête bien à la méthode essai-erreur – l'erreur étant un outil d'apprentissage très puissant, car il est rare de pouvoir rédiger une séquence de code qui résout un problème du premier coup. Pour résoudre le problème, les élèves doivent d'abord s'appuyer sur leurs connaissances antérieures, essayer différentes stratégies, établir des liens et tirer des conclusions.

Le codage est aussi un **outil de représentation mathématique**. Il est difficile de donner des tâches qui permettront aux élèves de s'exercer et de développer les compétences trouvées dans les contenus du programme-cadre sans contextualiser les tâches avec d'autres domaines en mathématiques. Le codage peut et devrait d'abord servir d'outil de représentation – un peu comme le matériel de manipulation – afin d'aider l'élève à mieux comprendre une situation mathématique. Les exemples fournis dans ce guide ne sont donc que des points de départ.

Rédiger un code, soit l'acte de rédiger des directives textuelles ou d'assembler des blocs de code, n'est pas la fin du processus de création. Afin de vérifier si le code donne le résultat espéré, il faut aussi **l'exécuter**. L'exécution du code peut comprendre une seule étape, par exemple cliquer sur un drapeau ou sur un bouton « exécuter », ou peut nécessiter la compilation de l'information trouvée dans le code dans une séquence de directives qui peuvent être comprises par l'ordinateur. Dans le cas d'un robot ou d'un microcontrôleur, par exemple, il faut téléverser le code dans l'appareil avant de pouvoir l'exécuter. Il est à noter que l'exécution à une seule étape comprend aussi une étape de « compilation », mais celle-ci se trouve en arrière-scène, donnant l'impression d'une exécution instantanée.

9 Pour plus de détails, voir [GEEM Lecture M-3](#), p.260-261 et [GEEM Littérature 4-6](#), p. 24-25 [consultés sur Edusource le 5 décembre 2022].

Progresser à travers les connaissances en codage

Il existe une suite à l'enseignement des connaissances en codage afin de s'assurer que les élèves ont certaines connaissances de base avant de passer à du code plus complexe. La séquence suggérée est la suivante :

ANNÉE D'ÉTUDES	CONNAISSANCE EN CODAGE À PRIORISER
1 ^{re}	Événements séquentiels
2 ^e	Événements simultanés (+ séquentiels)
3 ^e	Événements répétitifs (+ séquentiels, simultanés)
4 ^e	Événements imbriqués (+ séquentiels, simultanés, répétitifs)
5 ^e	Instructions conditionnelles (+ séquentiels, simultanés, répétitifs, imbriqués)
6 ^e	Efficacité du code, avec toutes les structures de contrôle vues de la 1 ^{re} à la 5 ^e année
7 ^e	Sous-programmes et dénombrement prédéfini, avec toutes les structures de contrôle vues de la 1 ^{re} à la 6 ^e année
8 ^e	Analyse de données pour la prise de décisions éclairées, avec toutes les structures de contrôle vues de la 1 ^{re} à la 7 ^e année

Bien que cette séquence d'enseignement des connaissances liées au codage ait une suite logique, avec des connaissances qui permettent d'accéder à des fonctions de plus en plus complexes d'année en année, il ne s'agit pas d'une « limite ». Les élèves vont certainement découvrir des fonctionnalités dans les langues et les logiciels de codage utilisés qui dépassent les connaissances à prioriser dans l'année d'étude en question. Cette curiosité et ce désir de créer des codes plus complexes doivent être permis, encouragés et valorisés, car c'est une façon d'engager les élèves dans les mathématiques.

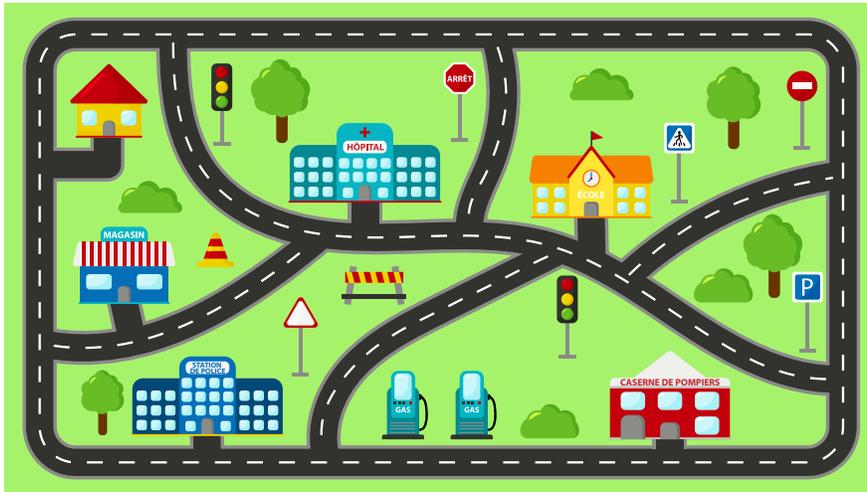
Les événements séquentiels

La séquence est vraiment la structure de base du codage. Le terme *séquence de code* est d'ailleurs souvent utilisé pour discuter du codage. Il importe de commencer par la séquence et les types d'actions qui peuvent être faits.

Dans un logiciel de codage par blocs, la séquence dans laquelle les blocs sont placés détermine l'ordre dans lequel les instructions seront exécutées. Dans ce contexte, le codage devient essentiellement une marche à suivre. Les types d'événements dans une séquence ne sont pas limités aux déplacements. Les sons, les changements visuels et même les moments d'attente peuvent faire partie d'une séquence et deviennent des événements séquentiels.

Événements séquentiels :
ensemble d'instructions exécutées les unes après les autres, habituellement de haut en bas ou de gauche à droite sur un écran.

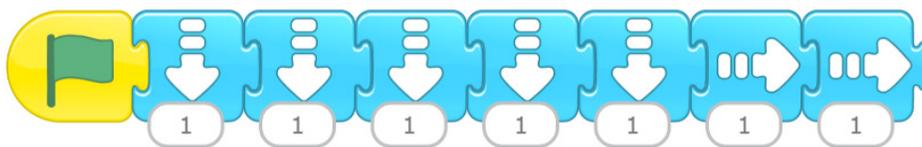
Prenons la carte d'une ville fictive.



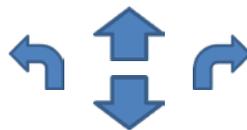
Un logiciel de codage ou de robotique visuel (par exemple, le codage par blocs) permettrait à l'élève d'animer un *sprite* afin que celui-ci puisse se déplacer à des endroits précis. Un pseudocode pour effectuer un déplacement de la maison au poste de police, en respectant les chemins, pourrait ressembler à ceci :

Position initiale = à gauche de la maison jaune
Condition de départ = vrai
Déplacer 5 pas Direction = bas
Déplacer 2 pas Direction = droite
Position finale = devant le poste de police

Le code correspondant à ce pseudocode, tel que programmé dans un logiciel de codage par blocs, pourrait ressembler à ceci :



Ce n'est qu'un exemple de séquence de déplacements qui permettrait au personnage de se rendre à destination. Il est aussi à noter que, dans certains contextes de codage, les possibilités de déplacement sont plus limitées; par exemple, certains robots fréquemment utilisés dans la salle de classe utilisent uniquement les commandes suivantes :



Dans ce cas, le vocabulaire *quart de tour*, *sens horaire* et *sens antihoraire* remplacent la gauche et la droite. Les directives demeurent, par contre, des événements séquentiels, car elles sont exécutées les unes après les autres.

Toujours selon la carte de la ville fictive, voici un pseudocode possible afin de partir du poste de police pour se rendre à l'école, toujours en suivant les chemins prévus :

Position = Poste de police
Condition de départ = vrai
Déplacer 6 pas Direction = droite →
Déplacer 2 pas Direction = haut ↑
Déplacer 4 pas Direction = diagonale ↗

Dans la plupart des cas, les logiciels de codage en blocs n'ont pas la capacité de donner des directives de déplacement en diagonale, ce qui serait nécessaire afin de satisfaire au but et de respecter le pseudocode. Il faut d'abord se servir d'un mouvement horizontal et vertical de manière **simultanée** afin d'avoir cet effet.

Les événements simultanés

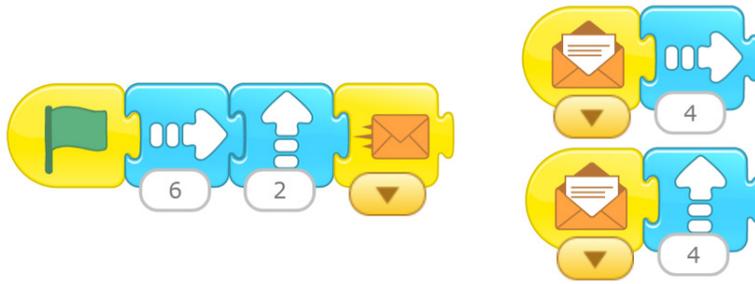
Le sens du mot *simultané* est bien connu. Il est à noter, en revanche, que des événements simultanés dans le contexte du codage sont des événements qui sont déclenchés par la même action. Il suffit de penser aux événements simultanés, comme une course d'automobile. Le feu vert signale le début de la course et toutes les voitures commencent la course en même temps. Les voitures n'effectueront pas nécessairement les mêmes manœuvres et ne roulent pas à la même vitesse, mais elles ont toutes reçu le même signal de départ. Il s'agit d'événements **simultanés**, même si les actions [vitesses, manœuvre, etc.] ne sont pas identiques.

Événements simultanés :
plusieurs événements
qui se produisent en
même temps.

Un pseudocode qui permet au *sprite* de passer du poste de police à l'école devient donc :

position = Poste de police
Condition de départ = vrai
Déplacer 6 pas Direction = droite →
Déplacer 2 pas Direction = haut ↑
Commencer le mouvement diagonal
Déplacer 4 pas Direction = droite →
Direction = haut ↑

La séquence dans un logiciel de codage pourrait ressembler à ceci :



Afin de mieux visualiser le code, l'élève pourrait aussi superposer son code sur son pseudocode. Cela aide à mieux relever les erreurs, s'il y en a, pour ensuite les corriger, tout en laissant des traces de la compréhension de l'élève des concepts mathématiques à l'étude. Cette pratique peut être très utile à tous les niveaux, surtout pour apprendre à utiliser un nouveau logiciel ou une nouvelle langue de codage.

Position = Poste de police	
Condition de départ = vrai	
Déplacer 6 pas	
Direction = droite →	
Déplacer 2 pas	
Direction = haut ↑	
Commencer le mouvement diagonal	
Déplacer 4 pas	
Direction = droite →	
Direction = haut ↑	

Note : Le logiciel de codage utilisé dans cet exemple emploie les événements « envoyer un message » et « recevoir un message ». Lorsque ce code est exécuté, l'arrivée au bloc « envoyer un message » (enveloppe fermée) déclenche, de manière simultanée, toutes les séquences qui commencent par le bloc « recevoir un message » (enveloppe ouverte). Les logiciels de codage peuvent varier dans l'imagerie pour atteindre ce même résultat.

L'élève pourrait aussi utiliser deux *sprites* différents qui naviguent sur la carte en même temps, ce qui satisfait aussi à l'exigence d'événements simultanés.

Les événements répétitifs

Il arrive parfois que la nature même d'une situation mathématique nécessite de la répétition. Par exemple, l'addition répétée est utilisée pour améliorer la compréhension conceptuelle de la multiplication, et la suite à motif répété est un prérequis à la suite croissante et à la reconnaissance de motifs.

Dans le monde du codage, le terme *boucle* est souvent utilisé pour nommer une répétition. Cette structure permet aussi de simplifier le code en limitant le nombre de blocs ou de lignes nécessaires pour atteindre un but.

Dans un logiciel de codage en blocs, la boucle prend souvent la forme d'une parenthèse carrée avec le code à répéter à l'intérieur. Voici quelques exemples de différentes plateformes de codage par blocs :

Événements répétitifs :
événements qui se répètent. Dans le cadre d'activités de codage, les boucles sont utilisées dans le code pour répéter les instructions.

SCRATCH JR	SCRATCH	MAKE.CODE
		

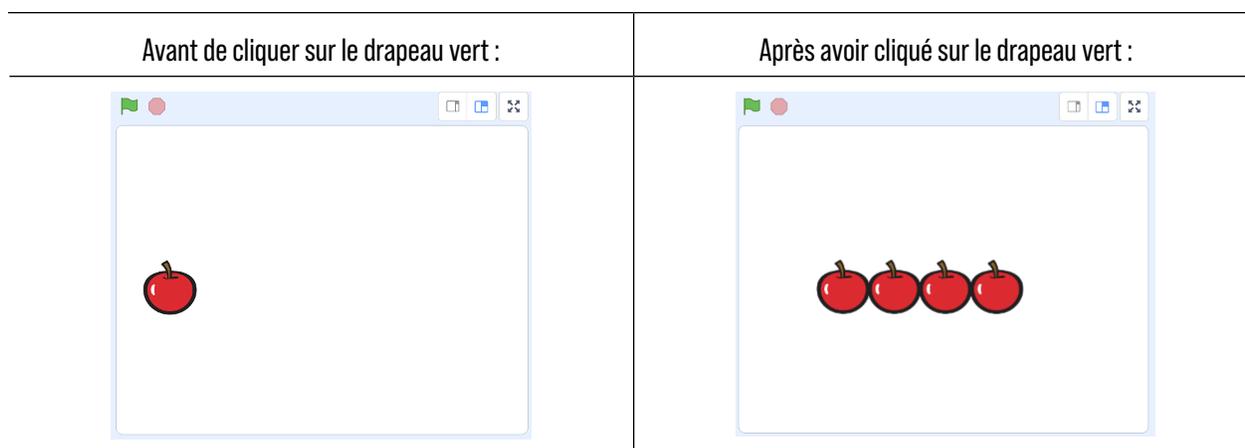
Dans chaque cas montré ci-dessus, le code qui se trouve à l'intérieur de la boucle serait répété 4 fois.

La boucle étant analogue à la multiplication en raison de sa nature, l'utilisation d'événements répétitifs est idéale pour résoudre des problèmes faisant appel à la pensée multiplicative.

Nous pouvons alors utiliser la boucle pour représenter une multiplication. Prenons le code suivant :



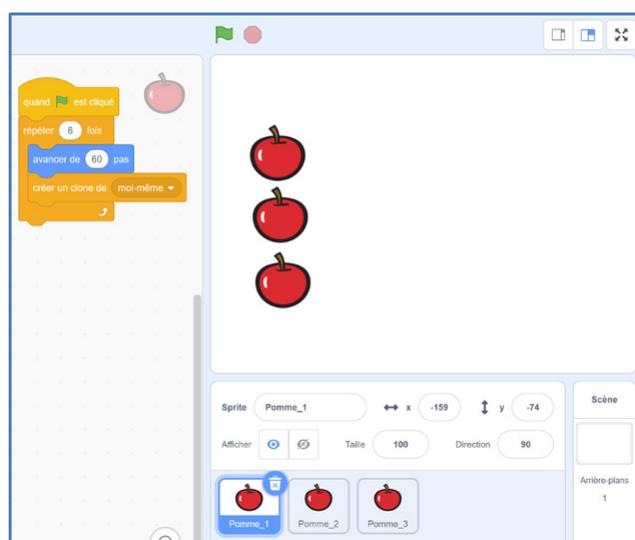
Le comportement du *sprite* par rapport à ce code sera de se déplacer de 60 pas (dans ce contexte, un pas est un pixel) et de faire une copie de lui-même. Ce comportement sera répété 4 fois, tel que défini par la boucle. Le code est donc une représentation de l'opération 4×1 , et le résultat est représenté à l'écran après avoir cliqué sur le drapeau vert.



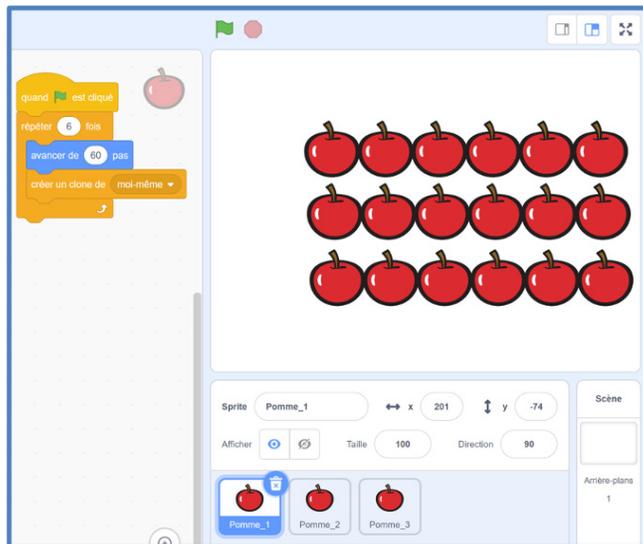
Des multiplications encore plus complexes peuvent être représentées en créant davantage de *sprites*.

Note : Dans l'exemple ci-dessus, le but mathématique est de représenter l'opération 4×1 avec du matériel, soit des pommes. La distance entre l'emplacement initial et l'emplacement final de la pomme pourrait aussi être représentée, soit 4×60 . Cette flexibilité dans la manière d'interpréter le code est une des grandes forces de l'utilisation du codage afin de représenter des situations mathématiques, mais cela montre aussi l'importance d'avoir une intention pédagogique claire.

En ajoutant des *sprites*, l'opération qui est représentée peut être modifiée. Dans cet exemple, il y a trois *sprites*, soit Pomme_1, Pomme_2 et Pomme_3, qui ont chacun un code identique.



Lorsque le code est exécuté en cliquant sur le drapeau vert, une représentation en disposition rectangulaire de l'opération 3×6 apparaît, soit 3 rangées de 6 pommes.



Dans la langue Python, la boucle se définit avec les termes *for* (indiquant un nombre de répétitions) et *while* (indiquant une répétition basée sur une condition). Bien qu'il soit peu probable que la langue Python soit vue en 3^e année, la terminologie associée peut aider à mieux comprendre la nature de la boucle. Dans le cas de l'exemple ci-dessus, il s'agit d'une boucle de type « for » – la répétition a une durée définie. Les conditions seront abordées davantage un peu plus loin dans ce guide.

Ce code représente bien l'opération 3×6 avec une disposition rectangulaire, mais il y a un grand désavantage – il est difficile à modifier. Pour représenter la multiplication 4×8 , par exemple, il faudrait créer un autre *sprite*, Pomme_4, et ensuite modifier le code pour que chaque *sprite* ait 8 répétitions au lieu de 6. Ce processus peut être long, et la structure du code fait en sorte que le débogage peut être difficile à accomplir, car le code se trouve à plusieurs endroits. La solution à ce dilemme est dévoilée en faisant travailler deux boucles ensemble.

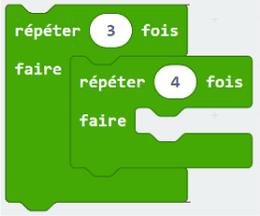
Les événements imbriqués

Lorsque des structures de contrôle travaillent ensemble, il s'agit d'**événements imbriqués**.

Événements imbriqués : structures de contrôle placées à l'intérieur d'autres structures de contrôle; par exemple, des boucles survenant à l'intérieur de boucles ou une instruction conditionnelle en cours d'évaluation dans une boucle.

Structures de contrôle : ligne ou bloc de code qui influence l'ordre dans lequel l'autre code est exécuté. Les structures de contrôle ont un effet sur le flux du programme et comprennent le séquençement des lignes de code, la répétition des lignes de code ou la sélection pour exécuter ou non des lignes de code précises. La séquence, la sélection (instructions conditionnelles) et la répétition (boucles) sont toutes des structures de contrôle.

De prime abord, des événements imbriqués sont des événements qui se trouvent à l'intérieur d'autres événements. Cela peut ressembler à plusieurs situations différentes selon le logiciel ou la langue de codage choisie.

SCRATCH	MAKE:CODE
	
PYTHON	
<pre> 2 for index in range(3): 2 for index2 in range(4): </pre>	
PSEUDOCODE (EXEMPLE)	
<pre> Répéter (3 fois) Répéter (4 fois) </pre>	

Ici, le contenu de la boucle intérieure sera répété 4 fois. Une fois l'opération terminée, la boucle extérieure prend le contrôle, ce qui force la boucle intérieure à recommencer. Cela se produira 3 fois.

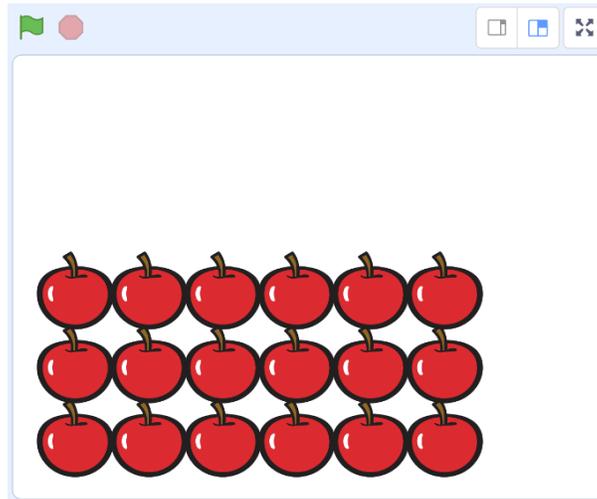
Certaines similarités peuvent être remarquées entre les logiciels de codage en blocs et les langues de codage. Le code se lit de haut en bas, et des retraits sont utilisés afin de montrer les éléments de code qui se trouvent dans une structure de contrôle. Cette observation pourrait être utile pour les élèves afin de bien organiser leur pseudocode avant de commencer le codage.

Pour revenir à l'exemple de la multiplication des pommes en disposition rectangulaire, il est alors possible de faire en sorte que le nombre de rangées de pommes soit défini par une boucle au lieu de dupliquer des *sprites*. De cette façon, le code sera contenu au même endroit, ce qui pourrait faciliter le débogage et réduire les endroits où des erreurs pourraient s'insérer.

Exemple de code (Scratch)

The image shows a Scratch script with the following blocks: 'quand est cliqué', 'aller à x: -190 y: -130', 'mettre rangée à 1', 'répéter 3 fois' (outer loop), 'montrer', 'répéter 6 fois' (inner loop), 'créer un clone de moi-même', 'avancer de 60 pas', 'aller à x: -190 y: -130', 'ajouter 60 à rangée à y', 'ajouter 1 à rangée', and 'cacher'. Five callout boxes provide explanations: 1. 'Ceci est la position initiale de la pomme, soit le coin inférieur gauche.' (points to the 'aller à' block). 2. 'Assure que la pomme est visible avant de commencer la multiplication.' (points to the 'montrer' block). 3. 'Après avoir créé une rangée de pommes, la pomme retourne à sa position initiale, et on la fait monter (axe y) de 60 pas x le nombre de rangées.' (points to the 'aller à' block). 4. 'La valeur de "rangée" est augmentée de 1 pour chaque répétition afin que la pomme débute chaque rangée au bon endroit.' (points to the 'ajouter 1 à rangée' block). 5. 'Cache la pomme originale pour ne pas avoir une pomme de trop à l'écran!' (points to the 'cacher' block).

Résultat de l'exécution



Le code représente maintenant une multiplication à l'aide de la disposition rectangulaire en utilisant une boucle imbriquée dans une autre boucle. La boucle externe représente le nombre de rangées, tandis que la boucle interne représente le nombre de pommes par rangée. La multiplication peut donc s'écrire comme suit :

$$[\text{répétitions boucle externe}] \times [\text{répétitions boucle interne}]$$

Ou, dans le cas de cet exemple,

$$3 \times 6$$

Pour aller plus loin avec ce même code, l'élève pourrait utiliser des variables pour les deux nombres à multiplier, représentant ainsi n'importe quelle multiplication en disposition rectangulaire. Un lien intéressant peut se faire ici avec l'aire et le périmètre d'un rectangle.

Les instructions conditionnelles

La transition des événements séquentiels vers les **instructions conditionnelles** représente un grand pas pour l'élève. Les instructions conditionnelles ouvrent la porte à toutes sortes de possibilités sur le plan du codage.

Les instructions conditionnelles sont d'abord présentes lorsque l'exécution d'un événement ou d'une séquence de code dépend de l'atteinte d'une condition précise. Des valeurs booléennes (vrai ou faux) sont souvent utilisées et associées à des termes tels que *si*, *sinon*, *jusqu'à* et *lorsque*.

Par exemple, un code représentant des multiplications à l'aide de la disposition rectangulaire pourrait être transformé afin de déterminer si la disposition représente un carré ou non. Il s'agirait alors du pseudocode suivant :

Si « valeur_1 » = « valeur_2 »

Alors

Dire « Oui, c'est un carré! »

Sinon

Dire « Non, ce n'est pas un carré! »

Puisqu'il est question d'une forme géométrique, les noms donnés à « valeur_1 » et à « valeur_2 » pourraient être plus précis, comme base et hauteur. Voici un exemple de code représentatif de ce pseudocode :

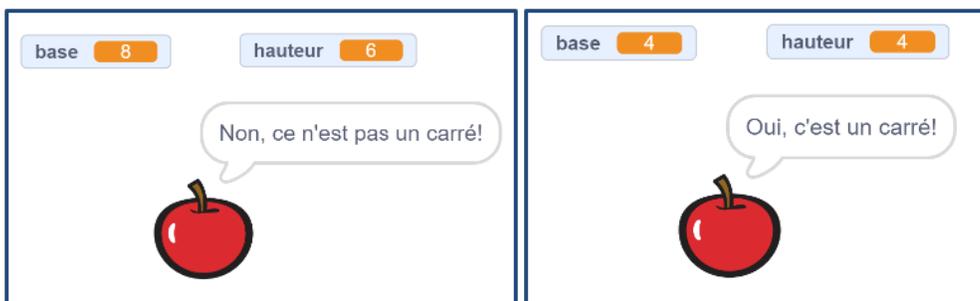


Afin de faire fonctionner ce code, il faut définir les variables « base » et « hauteur ». Quelques lignes de code peuvent être ajoutées pour demander à l'utilisateur d'entrer des valeurs afin de déterminer s'il s'agit d'un carré ou non.



Instructions conditionnelles : type d'instruction de codage qui ordonne à l'ordinateur de comparer des valeurs et des expressions et de prendre des décisions. Une instruction conditionnelle indique à un programme d'exécuter une action selon que la condition est vraie ou fausse.

Le résultat : le *sprite* indique si les dimensions entrées correspondent à la définition d'un carré.



Ce n'est qu'un exemple parmi un nombre infini de possibilités. Les instructions conditionnelles peuvent aussi utiliser la puissance des événements imbriqués. Au lieu de la disposition rectangulaire, il est possible de représenter une situation dans laquelle la probabilité théorique d'obtenir « pile » ou « face » en faisant tourner une pièce de monnaie est déterminée.

Dans ce contexte, il existe deux possibilités, soit « pile » ou « face ». Ces deux possibilités peuvent être associées à deux nombres, soit 1 et 2, et générer des résultats aléatoires qui sont ensuite compilés dans un tableau. Un pseudocode pour accomplir ce but pourrait ressembler à ceci :

Variables
Pile (mettre à 0)
Face (mettre à 0)
Lancers
Résultat
Demander « Combien de lancers voulez-vous générer? »
Lancers = Réponse
Répéter (« Lancers ») fois
Mettre « Résultats » à (nombre aléatoire entre 1 et 2)
Si « Résultat » = 2
Alors ajouter 1 à Pile
Sinon
Ajouter 1 à Face

Le code correspondant peut ressembler à ceci :

```
quand est cliqué
mettre Pile à 0
mettre Face à 0
demander "Combien de lancers voulez-vous générer?" et attendre
mettre Lancers à réponse
répéter Lancers fois
mettre Résultat à nombre aléatoire entre 1 et 2
si Résultat = 1 alors
ajouter 1 à Pile
sinon
ajouter 1 à Face
```

À titre d'exemple, voici le résultat généré par ce code lorsque 1 000 lancers ont été saisis :

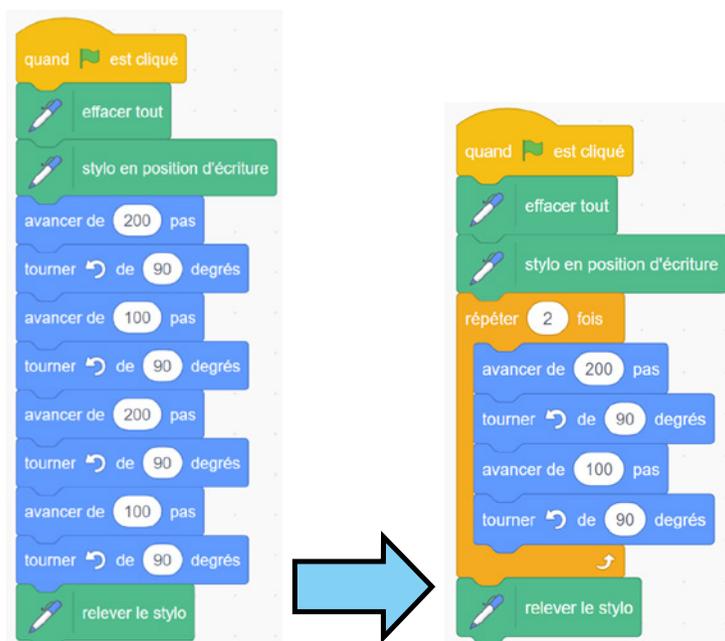


Ce résultat a été généré en moins d'une seconde. En comparant cela au temps requis pour accomplir 1 000 lancers en temps réel, il est possible de voir l'avantage d'utiliser un code pour résoudre le problème. De plus, les élèves peuvent maintenant utiliser les variables à l'intérieur de calculs afin de comparer la probabilité théorique et la probabilité expérimentale. Ou l'élève pourrait utiliser cette base pour déterminer la probabilité expérimentale d'une autre situation mathématique.

L'efficacité dans le contexte du codage

L'ajout d'instructions conditionnelles permet à l'élève de représenter et de résoudre des problèmes mathématiques très complexes. Il est important, cependant, de distinguer la complexité de la longueur du code! Un code peut être très complexe sans être très long. Il est alors important de présenter aux élèves le défi de produire un code fonctionnel dans un minimum de lignes de code. Plus une personne travaille sur un code, plus elle y ajoute des éléments, sans réaliser qu'il y a peut-être une façon plus efficace d'arriver au même résultat. Un code trop long, ou trop encombré, peut devenir plus difficile à déboguer et se prête mal au travail d'équipe.

Il existe plusieurs façons de rendre un code plus efficace. Il est possible, par exemple, d'insérer une boucle lorsqu'il y a des éléments du code qui se répètent. Dans cet exemple, le but est de dessiner un rectangle.



Ces deux codes accomplissent le même but, soit dessiner un rectangle, mais le code de droite le fait de manière plus efficace.

Plusieurs logiciels de codage textuels et par blocs permettent de personnaliser des blocs. Cela fait en sorte qu'une séquence de code peut être rédigée une seule fois, mais peut apparaître à plusieurs endroits dans le code principal. C'est une autre façon de rendre un code plus efficace.

Les sous-programmes

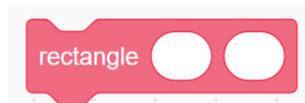
Un **sous-programme** permet de créer une petite séquence de code à laquelle il sera peut-être possible de faire appel plus souvent. Cette séquence ne fait pas partie du code principal. Il faut un événement déclencheur, un message, pour commencer le sous-programme. Lorsque le message est reçu, le sous-programme est exécuté en entier avant que le reste du code principal soit exécuté.

En ce qui concerne l'exemple du dessin des formes géométriques, il est possible de créer un bloc personnalisé qui, lorsqu'il est exécuté, fait appel à une série de directives précises (un sous-programme). Il s'agit de créer un bloc ayant deux paramètres réglables, soit la base et la hauteur. Dans ce cas, le bloc « rectangle » envoie le message de commencer à exécuter le code qui se trouve en dessous du bloc « définir : rectangle ». Cela fait en sorte qu'un rectangle de n'importe quelle taille peut être généré n'importe où sur l'écran, car le rectangle a été défini. Lorsque le rectangle aura été tracé, le code principal va continuer.

Sous-programme : petit ensemble d'instructions qui permet d'effectuer une tâche simple. Les sous-programmes peuvent être combinés dans un programme principal pour accomplir une tâche importante en utilisant des étapes simples.



Pour insérer le rectangle, il faut simplement ajouter le nouveau bloc « rectangle » au code et indiquer les valeurs de la base et de la hauteur.



Note : Dans le logiciel Scratch, il est possible de créer des blocs personnalisés à partir du menu « Mes blocs ». La manière d'accéder aux sous-programmes de ce genre dépend du logiciel ou de la langue choisie.

Ce genre de généralisation se prête bien à la modélisation mathématique, où un modèle peut être défini à l'intérieur d'un bloc personnalisé et ensuite utilisé dans un code afin de vérifier si le modèle est fonctionnel. Si un modèle qui représente bien la situation mathématique est créé, le bloc personnalisé sera une manière très efficace d'insérer le modèle, qui pourrait être très complexe, dans le code principal.

Les **instructions conditionnelles** peuvent aussi jouer un rôle important dans un tel code. Un exemple de cette intégration peut être démontré dans une situation de littératie financière, où l'intention est de donner une réduction si une personne dépense un certain montant. Il faut d'abord contrôler plusieurs paramètres afin que le code soit fonctionnel.

- Il faut être capable d'additionner le coût des articles qui seront achetés.
- Il faut confirmer que toutes les dépenses ont été saisies en s'assurant que la condition est remplie.
- Il faut définir les conditions pour obtenir une réduction.
- Il faut calculer la réduction et l'appliquer au total.

Si le nombre exact d'articles dans la liste est connu, le nombre de répétitions dans la boucle devient facile à prédire. Cependant, si ce nombre est inconnu, il faut utiliser un **dénombrement prédéfini**.

Dans ce cas, le bloc « répéter jusqu'à ce que » devient très utile, car il devient possible de définir la situation qui mettra fin à la boucle. Un exemple de code qui pourrait donc répondre aux besoins énumérés ci-dessus serait le suivant :

Dénombrement prédéfini :
 nombre de fois où des instructions sont répétées en fonction d'une valeur prédéfinie ou jusqu'à ce qu'une condition soit remplie.

```

    quand est cliqué
    mettre Somme des dépenses à 0
    supprimer tous les éléments de la liste Coût d'items achetés
    répéter jusqu'à ce que réponse = non
    demander Quel est le coût de l'item? et attendre
    ajouter réponse à Coût d'items achetés
    ajouter réponse à Somme des dépenses
    demander Voulez-vous entrer un autre item? et attendre
    si Somme des dépenses > 100 alors
        Pourcentage de rabais 20
    sinon
        si Somme des dépenses < 100 et Somme des dépenses > 75 alors
            Pourcentage de rabais 15
        sinon
            si Somme des dépenses < 75 et Somme des dépenses > 50 alors
                Pourcentage de rabais 10
    définir Pourcentage de rabais montant
    mettre rabais à montant / 100 * Somme des dépenses
    mettre Total à payer à Somme des dépenses - rabais
    
```

Ce code montre l'utilité de plusieurs structures de codage en même temps. La boucle à dénombrement prédéfini permet l'entrée d'un nombre illimité d'articles, jusqu'à ce que les conditions soient atteintes pour son arrêt. Les événements imbriqués permettent de déterminer si les conditions pour une réduction sont atteintes. Finalement, le bloc personnalisé « pourcentage de réduction » nous permet de modifier le pourcentage selon le montant dépensé.

L'élève pourrait ensuite ajouter du code pour calculer la taxe ou la monnaie à rendre. L'élève pourrait prendre le total et appliquer le taux d'intérêt d'une carte de crédit afin de déterminer si c'est un bon mode de paiement pour la dépense en question. L'élève pourrait aussi ajouter des blocs personnalisés pour des modalités de paiement (mensuel, trimestriel, hebdomadaire, etc.). Il existe plusieurs façons de personnaliser un code comme celui-ci.

L'analyse de données avec le codage

Afin de prendre de bonnes décisions, il faut de bonnes données. Les logiciels de codage et les microcontrôleurs (Arduino, Micro:bit, etc.) automatisent **la collecte et l'analyse des données**, ce qui permet les prises de décisions et fournit de bons arguments pour appuyer ces décisions. L'entrée de données peut se faire de plusieurs façons; par exemple, un code pourrait être rédigé afin de permettre la collecte de données au moyen d'un formulaire qui compilera les données dans un tableur électronique. La majorité des tableurs électroniques acceptent les codes pour faciliter l'analyse de données.

Combien de temps passes-tu à l'écran au cours d'une journée typique?

Moins d'une heure

Entre 1 h et 2 h

Entre 2 h et 3 h

Plus de 3 h

	A
1	Combien de temps passes-tu à l'écran lors d'une journée typique?
2	Entre 1 h et 2 h
3	Entre 1 h et 2 h
4	Entre 1 h et 2 h
5	Entre 1 h et 2 h
6	Entre 1 h et 2 h
7	Entre 1 h et 2 h
8	Entre 1 h et 2 h
9	Entre 1 h et 2 h
10	Entre 2 h et 3 h

Les données brutes peuvent être transformées en tableau de fréquences relatives, et une mise en forme conditionnelle peut être appliquée aux cellules pour mettre en surbrillance des données ou des tendances précises.

Combien de temps passes-tu à l'écran au cours d'une journée typique?			
Réponse	Fréquence	Fréquence cumulée	Fréquence relative exprimée en pourcentage
Moins d'une heure	2	2	8
Entre 1 h et 2 h	8	10	32
Entre 2 h et 3 h	11	21	44
Plus de 3 h	4	25	16

Règles de mise en forme conditionnelle ×

123 La valeur est supérieure à 20
D12:D15

123 La valeur est inférieure ou égale à 20
D12:D15

+ Ajouter une règle

Par exemple, la première règle pourrait être rédigée comme suit :
 « Si (valeur) > 20, ALORS couleur de la cellule = rouge. »

Note : Concernant la capture de la mise en forme conditionnelle, il faut faire un clic à droite sur la cellule et choisir « mise en forme conditionnelle ».

Les capteurs nous incitent à aller un pas plus loin. Plusieurs microcontrôleurs ont des capteurs installés ou enfichables qui permettent la collecte des données; par exemple, un microcontrôleur muni d'un capteur de lumière pourrait compiler le nombre de minutes par jour pendant lesquelles les lumières sont allumées dans une salle de classe, puis déterminer le coût en électricité basé sur l'efficacité écrite sur les ampoules. Le code pour cette tâche pourrait ressembler à ceci :

```

toujours
si niveau d'intensité lumineuse > 0 alors
  modifier compteur de 1
  pause (ms) 1000
lorsque le bouton A est pressé
  montrer nombre compteur
  
```

Dans cette circonstance, il faut ajouter une seconde au compteur pour chaque seconde pendant laquelle le capteur de lumière détecte de l'énergie lumineuse. Un bouton sur le microcontrôleur est associé au compteur afin d'afficher le nombre de secondes total sur un écran à DEL. L'élève devrait utiliser la méthode d'essai-erreur pour déterminer le seuil de détection de lumière appropriée pour ne pas avoir de données erronées.

Ces microcontrôleurs se trouvent partout dans le quotidien de l'élève – le thermostat intelligent qui agit comme interrupteur qui active le système de ventilation lorsqu'une température est atteinte, ou l'accéléromètre d'un téléphone intelligent ou une montre intelligente qui détermine le nombre de pas qu'une personne fait dans la journée afin de déterminer la distance parcourue. Des discussions sur les façons dont les données sont recueillies et analysées au moyen du codage permettront à l'élève d'explorer davantage les limites de ce qui est possible d'accomplir avec du code.

Situation d'apprentissage – 1^{re} année

Titre : La chasse aux objets

Durée : 75 minutes

Sommaire

Dans cette situation d'apprentissage, l'élève déplace un *sprite* dans une forêt à l'aide d'un logiciel de programmation par blocs. L'élève utilise aussi ses connaissances des faits numériques afin que chaque déplacement du *sprite* totalise 10 mouvements.

Cette situation d'apprentissage comprend des activités de codage branché, c'est-à-dire du codage nécessitant un logiciel, une plateforme ou un appareil de codage. Il est possible, cependant, d'adapter les activités afin qu'elles soient débranchées.

Attentes et contenus d'apprentissage ciblés

Nombres

B2. Utiliser ses connaissances des nombres et des opérations pour résoudre des problèmes mathématiques de la vie quotidienne.

B2.2 Se rappeler les faits d'addition de nombres jusqu'à 10 et les faits de soustraction associés, et démontrer sa compréhension de ces faits.

Algèbre

C3. Résoudre des problèmes et créer des représentations de situations mathématiques de façon computationnelle à l'aide de concepts et d'habiletés en codage.

C3.1 Résoudre des problèmes et créer des représentations de situations mathématiques de façon computationnelle en écrivant et en exécutant des codes, y compris des codes comprenant des événements séquentiels.

C3.2 Lire et modifier des codes donnés, y compris des codes comprenant des événements séquentiels, et décrire l'incidence des changements sur les résultats.

Pratiques pédagogiques à fort impact en mathématiques à privilégier

Résultats d'apprentissage, critères d'évaluation et rétroaction descriptive

Dans cette situation d'apprentissage, les résultats d'apprentissage et les critères d'évaluation pourront être affichés sur une grande feuille, accessible aux élèves. Les élèves doivent connaître les résultats attendus ainsi que la façon d'y accéder. Pendant toute la leçon, l'adulte circule, observe, questionne et commente le travail des élèves en leur donnant une rétroaction descriptive axée sur les résultats d'apprentissage et les critères d'évaluation.

Outils et représentations

L'utilisation de représentations de faits numériques est importante dans cette situation d'apprentissage. La représentation de 10 à l'aide d'une droite numérique, d'un Rekenrek, d'un cadre à 10 cases, d'une grille de nombres et d'objets, comme des jetons, est utilisée afin que l'élève puisse représenter différents déplacements qui totalisent 10. De plus, l'élève connaît aussi les soustractions associées.

Conversations mathématiques

Dès que la mise en situation débute, la conversation débute. Lorsque l'adulte présente les illustrations, les élèves se questionnent et conversent au sujet de leurs hypothèses et de leurs idées. Tout le long du déroulement, les élèves expliquent leurs déplacements à l'aide de blocs.

Connaissances et habiletés préalables

- Se servir de matériel, comme un Rekenrek, un cadre à 10 cases ou une droite perlée.
- Connaître les blocs de déplacement et les blocs de départ (possiblement vus en codage débranché).



Résultats d'apprentissage

À la fin de cette situation d'apprentissage, l'élève pourra :

- connaître les faits numériques d'addition et de soustraction jusqu'à 10 pour faire des déplacements;
- créer des codes d'événements séquentiels;
- modifier des codes pour en faire des codes séquentiels efficaces.

Critères d'évaluation selon les grilles d'évaluation du rendement

Connaissance et compréhension

- L'élève détermine les événements séquentiels d'un code.
- L'élève utilise les faits d'addition et de soustraction de 10.

Habiletés de la pensée

- L'élève modifie son code pour le rendre plus efficace ou plus original.

Communication

- L'élève utilise le vocabulaire approprié au codage (bloc, *sprite*).
- L'élève explique, avec clarté, les blocs qui sont dans son code.

Mise en application

- L'élève crée un code efficace qui fait bouger son *sprite*.
- L'élève utilise les variables dans ses blocs pour faire bouger le *sprite*.

Matériel requis

- cadre à 10 cases
- Rekenrek
- droite numérique
- cubes ou tout autre objet pour compter
- droite perlée
- jetons bicolores
- logiciel de programmation par blocs
- [annexe \[Mise en situation\]](#)

Note : Les exemples dans cette situation d'apprentissage ont été créés à partir de Scratch J.

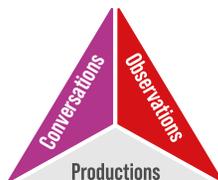
Vocabulaire mathématique

faits numériques, *sprite*, code, bloc, événement séquentiel

CONTENU

Mise en situation (15 minutes)

L'évaluation
peut se faire par les...



Présenter aux élèves les [illustrations](#) ci-dessous.

Pistes de questionnement

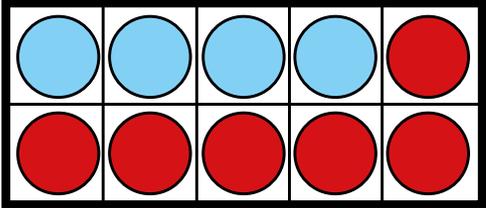
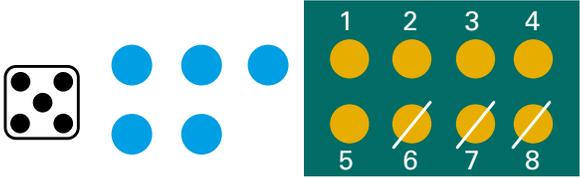
- Que vois-tu?
- Combien de doigts dois-tu ajouter pour en avoir 10?
- Combien d'arbres vois-tu?
- Combien d'arbres dois-tu ajouter pour en obtenir 10?



Présenter aux élèves du matériel de manipulation, comme le cadre à 10 cases, des jetons bicolores, un Rekenrek et une droite perlée.

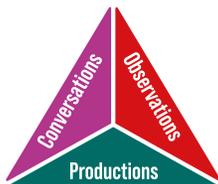
Demander aux élèves de trouver différentes façons de représenter 10.

Circuler et observer ce que font les élèves.

OBSERVATION POSSIBLE	PISTES D'INTERVENTION
L'élève a de la difficulté à représenter 10.	<p>Proposer à l'élève d'utiliser les jetons bicolores et de les placer dans le cadre à 10 cases.</p> <ul style="list-style-type: none">• Combien de jetons bleus as-tu placés?• Combien de jetons rouges as-tu placés?  <p>Rappeler à l'élève que l'on peut représenter 10 avec des objets, des opérations, des symboles et des dessins.</p> <p>Donner à l'élève un exemple pour le nombre 5.</p> <ul style="list-style-type: none">• Que vois-tu?• Comment sais-tu ce que ça représente? 

Déroulement (45 minutes)

L'évaluation
peut se faire par les...



Présenter aux élèves les flèches ci-dessous. Leur demander de se lever et de faire les déplacements.



Pistes de questionnement

- Combien de pas as-tu faits?
- Combien de flèches dois-tu ajouter pour faire 10 pas en tout?
- Observe les flèches ci-dessous. Combien de déplacements, vers la gauche, dois-tu effectuer pour avoir un total de 8?



Proposer aux élèves de créer des déplacements dans un logiciel de programmation par blocs. Les déplacements du *sprite* doivent représenter les faits d'addition jusqu'à 10 ou les faits de soustraction associés.

Présenter aux élèves le code suivant.



Pistes de questionnement

- Que va faire la grenouille?
- Combien de mouvements dois-tu ajouter pour qu'elle fasse 10 déplacements en tout?
- Qu'est-ce qui manque au code pour qu'il soit fonctionnel?

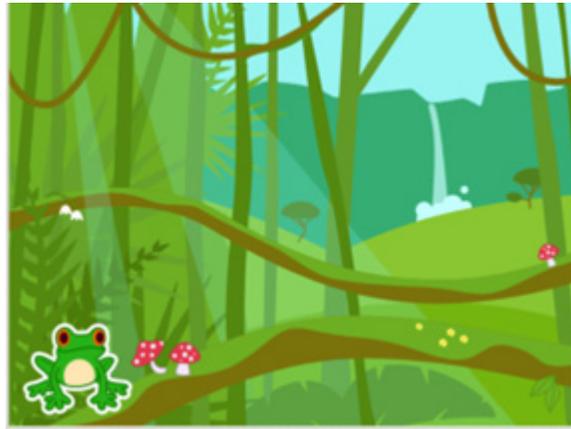
Guider la discussion et amener les élèves à proposer un code fonctionnel qui comporte 10 déplacements.

Réponse possible

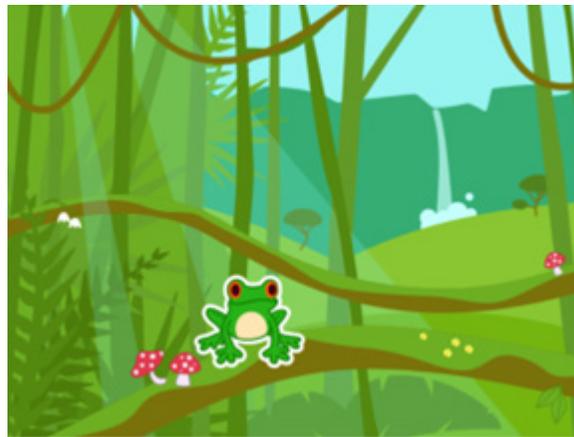
Voici maintenant le code pour que la grenouille fasse 10 déplacements. L'élève suggère d'ajouter 1 flèche vers le haut, 3 flèches vers la droite et 1 flèche vers le bas. L'ajout du bloc de départ et de celui d'arrêt est aussi nécessaire pour que le code fonctionne.



Voici la position de la grenouille avant les déplacements :

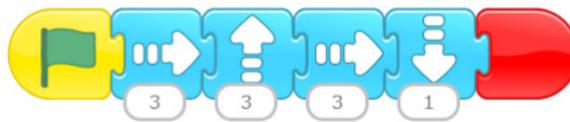


Voici la position de la grenouille après les déplacements :



- Faire ressortir les événements séquentiels du code.
- Demander aux élèves s'il y a une façon de modifier le code pour qu'il soit plus efficace. (Dans ce contexte, « plus efficace » pourrait être interprété comme l'utilisation de moins de blocs pour en arriver au même but.)

Voici un exemple de code plus efficace. Il faut ajouter des variables en-dessous des blocs. Le code demeure une série d'événements séquentiels.



- Proposer aux élèves de créer d'autres déplacements de leur grenouille dans la forêt à l'aide d'un logiciel de programmation par blocs.
- Mentionner aux élèves que les déplacements de la grenouille doivent représenter des faits d'addition ou de soustraction du nombre 10.

Réponses possibles

L'élève peut alors ajouter un bloc vert

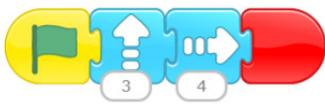


dans lequel la réponse est dictée; par exemple, l'élève dit : « Neuf plus un égalent dix. »



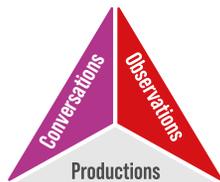
La grenouille est placée sur la grille quadrillée à 10. L'élève crée le code ci-dessous pour que la grenouille se trouve à 4.



OBSERVATIONS POSSIBLES	PISTES D'INTERVENTION
L'élève ne réussit pas à faire fonctionner son code.	<p>Quel bloc dois-tu utiliser pour commencer ton code?</p> 
L'élève utilise plusieurs blocs plutôt que de changer la variable de son bloc.	<p>Que peux-tu faire pour rendre ton code plus efficace? Remarques-tu des répétitions dans les blocs de ton code?</p> 
L'élève fait des déplacements qui ne représentent pas 10.	<p>Combien de déplacements la grenouille a-t-elle faits vers le haut? Représente ceux-ci en te servant des billes du Rekenrek. Combien de déplacements la grenouille a-t-elle faits vers la droite? Représente ceux-ci en utilisant les billes du Rekenrek.</p>  <p>Est-ce que le total des déplacements de la grenouille représente 10 billes, soit une rangée complète du Rekenrek? Sinon, combien de billes dois-tu ajouter?</p>

Objectivation (15 minutes)

L'évaluation
peut se faire par les...



Demander aux élèves de présenter leur code au groupe-classe. Inviter les autres élèves à les essayer.

Observer les codes avec les élèves et faire ressortir les événements séquentiels dans les codes.

Revoir avec les élèves les blocs et leur signification.

Consolidation

Proposer aux élèves de faire une chasse au trésor d'objets en utilisant des robots tels que Bee-Bot, Sphero, Ozobot ou autre. Si aucun appareil robotique n'est disponible, des versions interactives en ligne existent pour plusieurs robots éducatifs.

ÉLÉMENTS À CONSIDÉRER

Liens avec les autres domaines mathématiques (événements séquentiels)

Nombres

B2.2 Se rappeler les faits d'addition de nombres jusqu'à 10 et les faits de soustraction associés, et démontrer sa compréhension de ces faits.

En utilisant une variété de combinaisons de mouvements qui font 10, l'élève peut faire déplacer un *sprite* à des endroits précis. La soustraction peut être représentée par des déplacements dans le sens inverse (par exemple, 10 pas vers la gauche et 4 pas vers la droite, c'est un déplacement de 6 pas, ce qui représente $10 - 4 = 6$).

Algèbre

C1.2 Créer des suites à l'aide de mouvements, de sons, d'objets, de formes géométriques, de lettres et de nombres, et représenter les suites de différentes façons.

L'élève peut créer une suite à motif répété en utilisant les blocs dans le logiciel de codage, puis faire part de son code à une ou à un camarade de classe afin de déterminer la règle et prolonger la suite.

Données

D1.1 Trier et classer des ensembles de données portant sur des personnes ou des objets en fonction d'un attribut et décrire les critères de classement utilisés.

L'élève peut avoir une collection de *sprites* sur son écran et les classer selon leurs attributs en utilisant une séquence de déplacements afin que chaque *sprite* ait son endroit désigné.

Sens de l'espace

E1.5 Donner et suivre des directives pour se déplacer d'un endroit à un autre.

En utilisant le vocabulaire associé aux blocs de déplacement (à gauche, à droite, au haut, au bas), l'élève crée un parcours pour un *sprite* comportant une séquence de déplacements.

Différenciation pédagogique et conception universelle de l'apprentissage

- Donner une copie des blocs aux élèves et leur demander de résoudre le problème en codage débranché.
- Fournir un exemple déjà fait à l'élève pour pouvoir s'en inspirer.
- Proposer aux élèves d'utiliser leurs 10 déplacements afin d'amener le *sprite* à atteindre divers objets dans la forêt.
- Proposer aux élèves d'ajouter des pages à leur projet afin de faire déplacer leur *sprite* d'une page à l'autre.

Annexe – Mise en situation



Situation d'apprentissage – 2^e année

Titre : Une course d'obstacles

Durée : 75 minutes

Sommaire

Dans cette situation d'apprentissage, l'élève lit, modifie et crée des codes afin de faire naviguer un *sprite* au moyen d'une course d'obstacles. L'élève décrit également la position relative d'un objet et les déplacements requis pour passer d'un objet à un autre au moment de la création de la course d'obstacles.

Cette situation d'apprentissage comprend des activités de codage branché, c'est-à-dire du codage nécessitant un logiciel, une plateforme ou un appareil de codage. Il est possible, cependant, d'adapter les activités afin qu'elles soient débranchées.

Attentes et contenus d'apprentissage ciblés

Algèbre

C3. Résoudre des problèmes et créer des représentations de situations mathématiques de façon computationnelle à l'aide de concepts et d'habiletés en codage.

C3.1 Résoudre des problèmes et créer des représentations de situations mathématiques de façon computationnelle en écrivant et en exécutant des codes, y compris des codes comprenant des événements séquentiels et des événements simultanés.

C3.2 Lire et modifier des codes donnés, y compris des codes comprenant des événements séquentiels et des événements simultanés, et décrire l'incidence de ces changements sur les résultats dans divers contextes.

Sens de l'espace

E1. Décrire et représenter la forme, la position et le déplacement en se servant de propriétés géométriques et de relations spatiales pour s'orienter dans le monde qui l'entoure.

E1.5 Décrire la position relative d'objets divers et les déplacements requis pour passer d'un objet à l'autre.

Pratiques pédagogiques à fort impact en mathématiques à privilégier

Conversations mathématiques

Au moment de la mise en situation, une conversation mathématique est entamée avec les élèves du groupe-classe. Cette conversation les incitera à faire part aux autres de leurs idées, à montrer leur compréhension liée à la position relative d'un objet et aux déplacements, à raisonner et à prouver leur raisonnement. Les élèves pourront faire des liens avec des expériences vécues et communiquer leurs connaissances liées au sujet abordé. Des preuves peuvent être recueillies afin de guider l'enseignement.

Regroupements flexibles

Le travail en équipe de deux, au moment de la création et de la modification de codes, encourage notamment la communication et la collaboration. Les regroupements flexibles favorisent la collaboration et encouragent les élèves à prendre part à de riches conversations mathématiques. Les élèves peuvent apprendre les uns des autres. De cette façon, les élèves font évoluer leur réflexion mathématique. Tout en tenant compte du profil des élèves, les regroupements peuvent être aléatoires, hétérogènes ou homogènes, selon les besoins.

Enseignement en petits groupes

L'enseignement en petits groupes est une pratique pédagogique visant à faire progresser l'apprentissage des élèves. Tout le long de la création et de la modification d'un code, il est important de circuler parmi les groupes d'élèves afin de leur fournir des appuis spontanés nécessaires selon ce qui est observé et entendu. L'enseignement en petits groupes donne l'occasion de revoir avec des élèves des concepts mathématiques qui appuient leur nouvel apprentissage. Ces élèves pourront donc approfondir leur compréhension des concepts à l'étude (position et déplacement ou codage).

Enseignement explicite

Lorsqu'un code ne donne pas le résultat attendu, le personnel enseignant pourrait demander aux élèves de l'expliquer. C'est alors un moment propice pour offrir un enseignement explicite lié à la difficulté observée. Cet enseignement permet de vérifier la compréhension des élèves et de leur fournir une rétroaction immédiate.

Connaissance et habileté préalable

Il serait préférable que les élèves aient l'occasion d'explorer le logiciel de codage qui sera utilisé afin de connaître l'interface et les éléments de base. Cependant, cette situation d'apprentissage peut aussi servir d'exploration initiale d'un logiciel en allouant plus de temps, afin que les élèves se familiarisent avec celui-ci dans le déroulement.

Résultats d'apprentissage

À la fin de cette situation d'apprentissage, l'élève pourra :

- utiliser la terminologie propre à la position relative d'un objet (au-dessus, en dessous, sur, sous, à côté, à l'intérieur, à l'extérieur, à droite, à gauche) et au codage (*sprite*, événements séquentiels, événements simultanés);
- décrire les déplacements d'un *sprite* (vers la droite, vers la gauche, vers le haut, vers le bas);
- rédiger une séquence de code permettant à plus d'un *sprite* d'effectuer en même temps des déplacements;
- lire une séquence de code, puis la modifier afin de la rendre plus efficace.

Critères d'évaluation selon les grilles d'évaluation du rendement

Connaissance et compréhension

- L'élève connaît l'apparence et la fonction des blocs dans le logiciel de codage utilisé.
- L'élève décrit les déplacements d'un objet vers un autre en faisant référence à la direction et au nombre de déplacements.
- L'élève détermine les blocs et les actions de chaque bloc utilisé dans les événements séquentiels et simultanés.

- L'élève comprend ce que représentent deux séries d'instructions simultanées.

Habilités de la pensée

- L'élève planifie l'emplacement et la durée d'un bloc « pause » afin de bien synchroniser les événements simultanés dans son code.
- L'élève prédit le résultat d'un code et le modifie afin d'atteindre un but spécifique.
- L'élève montre une compréhension de la différence entre des événements séquentiels et des événements simultanés dans le contexte du codage.
- L'élève modifie un code et détermine l'effet des changements sur le comportement d'un *sprite*.

Communication

- L'élève explique sa séquence de code en utilisant la terminologie appropriée (à gauche, à droite, au haut, au bas, etc.).
- L'élève utilise le vocabulaire lié à la position relative d'un objet (au-dessus, en dessous, sur, sous, à côté, à l'intérieur, à l'extérieur, à droite, à gauche).

Mise en application

- L'élève met les blocs aux bons endroits et dans le bon ordre afin de créer des séquences qui contiennent des événements simultanés.

Matériel requis

- logiciel de programmation par blocs approprié selon l'habileté de l'élève
- papier quadrillé et crayons (optionnel)
- [annexe 1 \(Mise en situation\)](#)
- [annexe 2 \(Codes du papillon et de l'oiseau\)](#)

Note : Les exemples dans cette situation d'apprentissage ont été créés à partir de Scratch Jr.

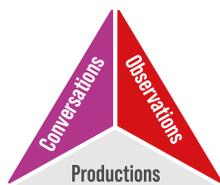
Vocabulaire mathématique

code, *sprite*, événements séquentiels, événements simultanés, à gauche, à droite, au-dessus, en dessous, devant, derrière, déplacements (x unités vers la gauche, vers la droite, vers le haut, vers le bas)

CONTENU

Mise en situation (10 minutes)

L'évaluation
peut se faire par les...



Montrer aux élèves la photo ci-dessous ([annexe 1](#)). Leur poser la question suivante : « Que remarques-tu ? » Utiliser la stratégie du PPP (Pense-Parle-Partage) afin d'amorcer une conversation mathématique avec les élèves.



P1 – Pense

Exiger une minute, au minimum, de réflexion en silence. Cela fait en sorte que les élèves réfléchissent sans la pression du temps ni l'influence des réponses des autres.

P2 – Parle

Demander aux élèves de discuter, en équipe de deux, de leurs premières impressions au sujet de la photo.

P3 – Partage

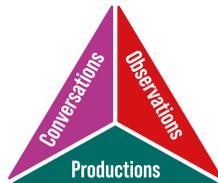
Ouvrir la conversation. Dans un premier temps, inviter les élèves à exprimer leurs impressions au sujet de la photo ainsi que leurs constats à la suite de leur conversation avec leur partenaire. Tenter ensuite de diriger le questionnement vers les concepts mathématiques qui seront à l'étude.

Pistes de questionnement et réponses possibles des élèves :

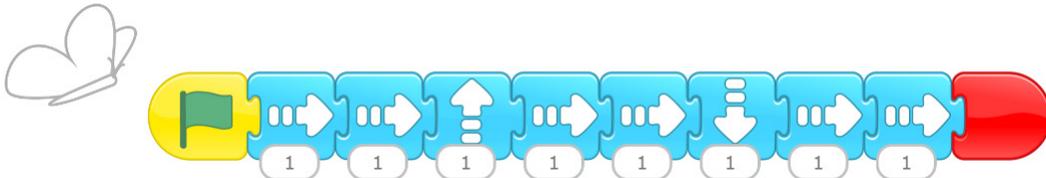
- Que font les enfants? Comment s'appelle ce type d'activité?
 - Une course d'obstacles.
- Pouvez-vous me décrire la course d'obstacles? Que voyez-vous?
 - Une enfant saute par-dessus une corde.
 - Il y a un tunnel derrière la corde. Il est à gauche des deux enfants debout en arrière. Il y a des drapeaux attachés aux arbres, au-dessus des enfants.
- Combien d'enfants font la course?
 - Trois enfants font la course.
 - Une enfant fait la course et les deux autres attendent leur tour.
- Comment peut-on déterminer qui gagnera?
 - Chronométrer les enfants pour déterminer qui a fait la course le plus rapidement possible.
 - Demander aux enfants de faire la course en même temps. L'élève qui arrive en premier gagne la course.

Déroulement (50 minutes en plus du temps alloué à l'exploration du logiciel de codage, au besoin)

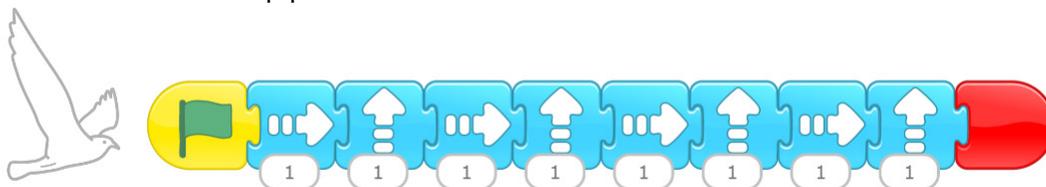
L'évaluation
peut se faire par les...



Présenter aux élèves le [code du papillon](#) ci-dessous, en l'affichant sur un écran ou en se servant des photocopies en couleurs.



Former des équipes de deux et inviter les élèves à étudier le code afin de décrire le mouvement du papillon. Lorsque les élèves sont satisfaits de leur réponse, leur demander de faire le même exercice en utilisant le [code de l'oiseau](#) ci-dessous. Cette fois-ci, étant donné que les équipes évolueront peut-être à des rythmes différents, il serait préférable de leur distribuer des photocopies en couleurs faites sur des bandes de papier.



De retour en groupe-classe, animer une conversation qui aura pour but de comparer les deux codes.

Pistes de questionnement et réponses possibles des élèves :

- Qui voyage le plus loin, le papillon ou l'oiseau?
 - Le papillon voyage le plus loin, car il se déplace de 6 cases vers la droite, alors que l'oiseau se déplace uniquement de 4 cases vers la droite.
 - L'oiseau voyage le plus loin, car il se déplace de 4 cases vers le haut, alors que le papillon se déplace d'une case vers le haut, puis d'une case vers le bas.
- Qui aura terminé son mouvement en premier?
 - Les deux animaux ont 8 blocs dans leur code, alors les deux termineront en même temps.
 - L'oiseau a uniquement 4 blocs de mouvement à droite, alors il terminera son mouvement avant le papillon.
- Que veut dire le drapeau vert dans le code?
 - C'est le signal de départ.

Permettre aux élèves d'accéder au logiciel de codage en blocs choisi afin de recréer le code et de vérifier leurs réponses aux questions.

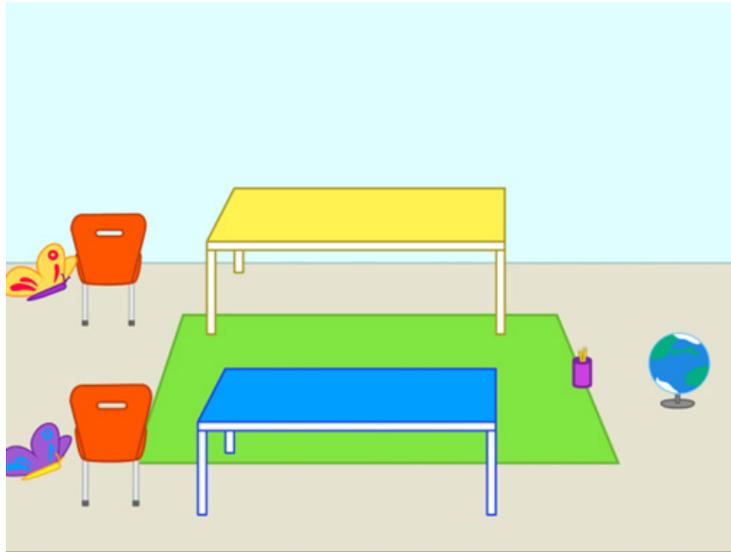
Par la suite, allouer aux élèves un peu de temps d'expérimentation avec le logiciel afin de modifier le code donné pour changer le comportement du papillon et celui de l'oiseau. Cela donne l'occasion aux élèves de découvrir l'effet de changer, d'ajouter et d'omettre certains blocs du code.

OBSERVATIONS POSSIBLES	PISTES D'INTERVENTION
Le code de l'élève ne ressemble pas du tout au code de départ présenté.	Demander à l'élève d'expliquer à voix haute les blocs de son code et leur effet. Fournir à l'élève une copie papier du code à dupliquer afin de pouvoir le comparer avec son code.
Le code de l'élève comprend des erreurs de logique (le code ne donne pas le résultat désiré).	Fournir à l'élève une feuille de papier quadrillé afin de pouvoir tracer les mouvements et déterminer les blocs causant l'erreur (cela ressemble à un pseudocode). Suggérer à l'élève de faire part de son code à une autre personne qui pourrait le relire et lui donner des pistes de solution (comme la révision dans le processus d'écriture).
Le code de l'élève comprend des erreurs de syntaxe (le code ne fonctionne pas du tout).	Revoir avec l'élève les situations de départ (par exemple, le drapeau vert). Demander à l'élève de vérifier si les blocs de son code sont bien connectés.

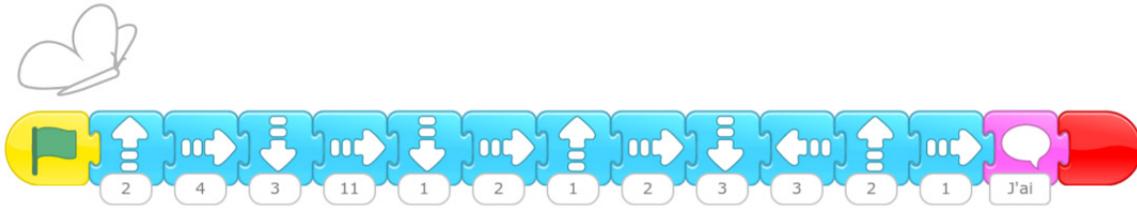
Donner maintenant aux élèves une intention plus spécifique pour leur code. Une course d'obstacles doit être créée, dans laquelle deux *sprites* font une compétition, un peu comme le papillon et l'oiseau. Les élèves choisissent les deux *sprites* ainsi que les obstacles de la course et décrivent les mouvements des *sprites* liés aux obstacles. Un des *sprites* devra atteindre la ligne d'arrivée avant l'autre.

Réponse possible

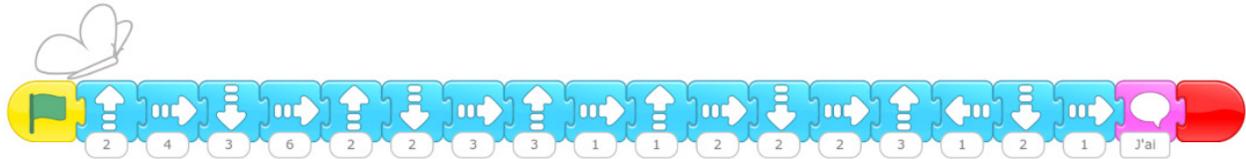
Trajet : Les papillons devront voler par-dessus la chaise, en dessous de la table, toucher le porte-crayon mauve et faire le tour du globe pour terminer devant celui-ci. Le papillon gagnant sera celui qui crie en premier « J'ai terminé! ».



Code possible du papillon jaune :



Code possible du papillon violet :



Note : Il est important d'encourager les élèves à exécuter leur code souvent. Une grande portion de l'apprentissage peut se faire par essai et erreur.

OBSERVATIONS POSSIBLES	PISTES D'INTERVENTION
L'élève ne sait pas par où commencer.	<p>Demander à l'élève de choisir les obstacles pour la course dans le menu de <i>sprites</i>, puis de déterminer le trajet.</p> <p>Fournir à l'élève du matériel physique avec lequel une représentation de la course d'obstacles pourrait être construite, afin d'avoir un repère visuel avant de commencer à coder.</p> <p>Demander à l'élève d'écrire, sur du papier, un pseudocode, c'est-à-dire une description ou un dessin de sa course d'obstacles. Du papier quadrillé pourrait être utilisé.</p>
L'élève ne dévie pas du tout du code fourni au début de l'activité.	<p>Demander à l'élève de décrire le trajet de la course d'obstacles. Est-ce que les mouvements des <i>sprites</i> reflètent la course? Comment peut-on rendre les mouvements plus réalistes?</p> <p>Encourager l'élève à modifier les blocs, un à la fois, et à exécuter le code à chaque changement afin d'en voir l'effet sur le comportement des <i>sprites</i>.</p>
L'élève crée deux codes identiques pour ses deux <i>sprites</i> .	<p>Note : Cela est très commun que l'élève crée deux codes identiques pour ses deux <i>sprites</i> lorsqu'on parle d'événements simultanés, croyant que les deux <i>sprites</i> doivent faire la même chose en même temps.</p> <p>Demander à l'élève si c'est possible de déterminer quel est le <i>sprite</i> gagnant lorsque les deux codes sont identiques.</p> <p>Utiliser l'exemple d'une autre sorte de course pour activer les connaissances antérieures de l'élève. Comment sait-on qu'une course est commencée? Un feu vert, un pistolet, un son? Une fois que la course est commencée, est-ce que les <i>sprites</i> bougent exactement de la même façon?</p>

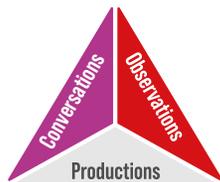
Note : Plusieurs logiciels de codage permettent aux élèves d'enregistrer leur voix, ce qui leur offre la possibilité de créer une narration pour la course et, aussi, de communiquer leur compréhension des concepts oralement plutôt que par écrit.



Exemple d'un bloc permettant l'enregistrement de la voix dans Scratch J'.

Objectivation (15 minutes)

L'évaluation
peut se faire par les...



- Choisir quelques codes à présenter au groupe-classe. Le choix peut être fait en demandant à des volontaires ou en sélectionnant de manière stratégique des codes mettant en valeur l'intention pédagogique, soit la familiarisation des élèves avec les déplacements et les positions relatives d'objets, ainsi que les événements simultanés.
- Demander aux équipes choisies de présenter leur code au groupe-classe. L'équipe doit expliquer son code ainsi que les règlements de sa course d'obstacles (les mouvements à faire et comment gagner). Ensuite, les élèves du groupe-classe analysent le code pour tenter de prédire qui gagnera la course. Finalement, le code est exécuté pour connaître le *sprite* qui gagnera.
- Poser aux élèves les questions suivantes :
 - Comment fait-on pour s'assurer que les deux codes commencent en même temps? (En cliquant sur le drapeau vert.)
 - Que se produirait-il si l'on cliquait sur chacun des *sprites* pour exécuter le code? (Il faudrait déplacer le pointeur de la souris d'un *sprite* à l'autre afin de les activer, alors ce serait impossible de les activer exactement en même temps.)



Exemple d'un bloc permettant de commencer une séquence en cliquant le *sprite*.

- Dans ce cas-ci, est-ce que les événements demeurent simultanés? (Même si les deux codes ne sont pas exécutés en même temps, il y a des moments durant lesquels les deux *sprites* bougent en même temps. Il s'agit d'événements simultanés à ces moments-là.)

Consolidation

La robotique est une excellente façon de consolider l'apprentissage en codage. Une course dans laquelle un robot, comme Sphero, Ozobot, LEGO ou autre, compétitionnerait avec un autre robot ayant un code différent pourrait rendre le concept d'événements simultanés encore plus clair pour les élèves.

Option débranchée

Si l'accès au matériel est limité aux plateformes sur ordinateur, il serait possible de mettre en valeur les événements simultanés en demandant aux élèves de faire déplacer des *sprites* sur des parcours en diagonale (par exemple, pour tracer un triangle). Le déplacement diagonal nécessite un déplacement horizontal et un déplacement vertical de façon simultanée, car il n'existe pas de bloc spécifique pour ce genre de déplacement. Le déplacement diagonal sera plus efficace, sur le plan du temps, qu'un déplacement horizontal suivi d'un déplacement vertical, même si la position finale est la même. Cela pourrait rendre d'éventuelles « courses d'obstacles » plus intéressantes.

ÉLÉMENTS À CONSIDÉRER

Liens avec les autres domaines mathématiques (événements simultanés)

Nombres

B1.5 Décrire les caractéristiques des nombres pairs et impairs.

En se servant d'une quantité inconnue de *sprites*, l'élève peut rédiger une séquence de codes permettant aux *sprites* de se placer, deux par deux (de façon simultanée), de manière organisée, à l'écran. L'élève peut déterminer si le nombre de *sprites* est pair ou impair en observant s'il en reste ou non après le jumelage.

B2.6 Représenter et résoudre des problèmes relatifs à la division de 12 éléments ou moins en tant que partage égal d'une quantité, à l'aide d'une variété d'outils et de schémas.

En utilisant un nombre de *sprites* de 1 à 12, l'élève peut rédiger un code dans lequel les objets se placent dans des groupes égaux; par exemple, s'il y a 6 *sprites* à l'écran et que le but est de créer 3 groupes égaux, l'élève crée un code pour que 3 *sprites* se déplacent de façon simultanée vers trois régions différentes de l'écran, puis fait de même pour les trois autres *sprites* afin de créer 3 groupes de 2 *sprites*.

Sens de l'espace

E1.5 Donner et suivre des directives pour se déplacer d'un endroit à un autre.

L'élève peut rédiger une séquence de codes afin qu'un *sprite* se déplace sur une carte. Les mouvements en diagonale nécessitent des événements simultanés, soit un déplacement horizontal en même temps qu'un déplacement vertical.

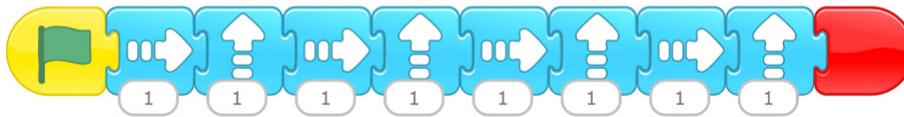
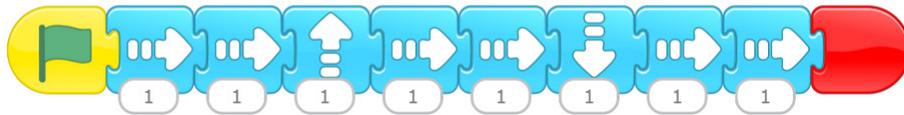
Différenciation pédagogique et conception universelle de l'apprentissage

- Un aide-mémoire des blocs et de leur fonction ou le code de couleurs utilisé par le logiciel de codage pourrait être utile pour les élèves qui n'ont pas encore une bonne connaissance du logiciel.
- Le travail individuel qui consiste à coder une course d'obstacles pourrait être un travail en équipe de deux ou de trois (éviter les équipes de plus de trois afin d'encourager la participation de chaque élève).
- Inviter l'élève à tenter de créer une course avec trois *sprites* ou une course où le code de chaque *sprite* est différent, mais les *sprites* terminent la course en même temps.

Annexe 1 – Mise en situation



Annexe 2 – Codes du papillon et de l'oiseau



Situation d'apprentissage – 3^e année

Titre : Les biscuits sans fin

Durée : de 140 à 150 minutes

Sommaire

Les élèves pourront mettre en pratique les faits de multiplication ainsi que la disposition rectangulaire en créant un code pour la machine du boulanger.

Attentes et contenus d'apprentissage ciblés

Nombres

- B2.** Utiliser ses connaissances des nombres et des opérations pour résoudre des problèmes mathématiques de la vie quotidienne.
 - B2.4** Démontrer sa compréhension des algorithmes de l'addition et de la soustraction de nombres naturels en établissant des liens avec les outils et les stratégies utilisés pour additionner et soustraire, et décrire ces liens.
 - B2.6** Représenter la multiplication de nombres jusqu'à 10×10 et la division de nombres jusqu'à $100 \div 10$ à l'aide d'une variété d'outils et de schémas, y compris des dispositions rectangulaires.

Algèbre

- C3.** Résoudre des problèmes et créer des représentations de situations mathématiques de façon computationnelle à l'aide de concepts et d'habiletés en codage.
 - C3.1** Résoudre des problèmes et créer des représentations de situations mathématiques de façon computationnelle en écrivant et en exécutant des codes, y compris des codes comprenant des événements séquentiels, simultanés et répétitifs.
 - C3.2** Lire et modifier des codes donnés, y compris des codes comprenant des événements séquentiels, simultanés et répétitifs, et décrire l'incidence de ces changements sur les résultats.

Pratiques pédagogiques à fort impact en mathématiques à privilégier

Résultats d'apprentissage, critères d'évaluation et rétroaction descriptive

Afficher, dans la salle de classe, les critères d'évaluation pour que les élèves s'y réfèrent pendant leur travail. Tout le long de l'activité, prendre le temps de circuler dans la salle de classe et d'observer ce que font les élèves. Les interroger au sujet de leur travail et leur donner une rétroaction descriptive liée aux critères d'évaluation affichés.

Conversations mathématiques

Dans la mise en situation, les élèves auront l'occasion de discuter des solutions possibles à apporter au code. Au cours de cette discussion, les élèves pourront écouter les opinions des autres élèves, expliquer leur raisonnement et ajouter des idées à celles des autres.

Outils et représentations

Les élèves mettent en pratique leur connaissance des multiplications et solidifient leur apprentissage en les représentant à l'aide de dispositions rectangulaires. Diverses situations pourront être modélisées de façon visuelle et comparées avec d'autres représentations. L'utilisation efficace d'outils et de représentations signifie que les différentes représentations sont valorisées et que des liens seront établis entre elles.

Connaissances et habiletés préalables

- Maîtriser les faits numériques de multiplication de 0 à 10.
- Être capable de représenter une multiplication à l'aide de la disposition rectangulaire.
- Être capable de représenter des groupes égaux avec un reste.
- Maîtriser les algorithmes d'addition et de soustraction de nombres naturels.
- Avoir une compréhension de base en codage du logiciel de son choix :
 - l'utilisation du bloc de répétition pour des boucles définies et indéfinies;
 - le clonage de *sprite*;
 - le retour au départ (sans connaître les coordonnées);
 - l'utilisation de variables.

Résultats d'apprentissage

À la fin de cette situation d'apprentissage, l'élève pourra :

- écrire et exécuter le code d'une représentation de disposition rectangulaire;
- ajouter une ou plusieurs boucles dans son code afin de le rendre plus efficace;
- utiliser les faits numériques de multiplication pour s'aider à disposer des pâtisseries de façon efficace sur une plaque à pâtisserie.

Critères d'évaluation selon les grilles d'évaluation du rendement

Connaissance et compréhension

- L'élève connaît les blocs d'un logiciel de programmation par blocs et comprend leur fonctionnement.
- L'élève utilise la disposition rectangulaire pour représenter les faits de multiplication jusqu'à 10×10 .

Habiletés de la pensée

- L'élève utilise un outil de codage pour représenter des multiplications à l'aide de la disposition rectangulaire.
- L'élève utilise les algorithmes d'addition et de soustraction pour créer des dispositions rectangulaires de nombres premiers.
- L'élève identifie des événements répétitifs et les place dans une boucle afin de rendre le code plus facile à lire et à modifier (efficacité).
- L'élève modifie un code erroné de sorte qu'il fonctionne.

Communication

L'élève utilise des blocs de répétition (boucles) afin de rendre sa programmation plus efficace.

Mise en application

- L'élève corrige un code erroné en utilisant ses connaissances des fonctions des blocs dans le logiciel de codage utilisé.
- L'élève fait des liens entre la disposition des blocs de codage et les faits numériques (par exemple, une boucle peut représenter une addition répétée ou une multiplication).

Matériel requis

- ordinateur et application ou logiciel de codage au choix
- [annexe \(Mise en situation\)](#)

Note : Les exemples dans cette situation d'apprentissage ont été créés à partir de Scratch.

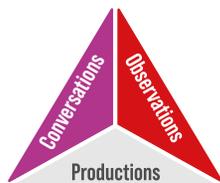
Vocabulaire mathématique

disposition rectangulaire, code, événements séquentiels, événements simultanés, événements répétitifs, boucle, variable

CONTENU

Mise en situation (30 minutes)

L'évaluation
peut se faire par les...



Présenter aux élèves la photo ci-dessous de la [mise en situation](#). Les inviter à communiquer leurs observations et leurs questions au sujet de la photo (je remarque, je me demande).



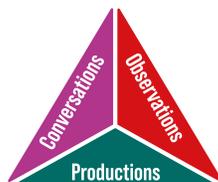
Exemples de questions ou d'observations :

- Combien de biscuits y a-t-il en tout?
- Comment peux-tu déterminer le nombre de biscuits sans les compter un par un?
 - Je vois une plaque à pâtisserie sur laquelle il y a 4 rangées de 5 biscuits et une plaque à pâtisserie sur laquelle il y a 2 rangées de 3 biscuits.
 - Je crois qu'il y a 20 biscuits sur la plaque à pâtisserie du bas, même si je ne vois pas la rangée du haut, car $4 \times 5 = 20$.

OBSERVATIONS POSSIBLES	PISTES D'INTERVENTION
L'élève ne sait pas par où commencer.	Que dois-tu faire actuellement? Que vois-tu?
L'élève n'arrive pas à faire des liens avec ses connaissances antérieures en mathématiques.	Que remarques-tu dans la photo qui te fait penser aux mathématiques? Est-ce que tu sais combien il y a de biscuits? Comment le sais-tu?
L'élève a de la difficulté à exprimer ses observations, à l'aide du vocabulaire mathématique.	Est-ce que le nombre de biscuits pourrait être représenté à l'aide d'une addition répétée? Est-ce que cette addition répétée pourrait être représentée par une multiplication? Sur la plaque à pâtisserie du bas, combien de rangées de biscuits y a-t-il? Combien de biscuits y a-t-il dans chaque rangée?

Déroulement (100 minutes)

L'évaluation peut se faire par les...



Présenter aux élèves la situation ci-dessous ainsi que le code :

À la boulangerie, Maxime doit planifier où mettre les pâtisseries sur les plaques à pâtisserie avant de les faire cuire. La machine qui place les pâtisseries sur une plaque à pâtisserie utilise un logiciel de codage en blocs. Maxime crée le code ci-dessous afin de mettre 24 pâtisseries sur une plaque.

Exemple du code de Maxime :



Questions à poser aux élèves :

- Que remarques-tu?
- Que cherche-t-on à faire avec ce code? (Mettre 24 pâtisseries sur une plaque à pâtisserie.)
- Combien de pâtisseries penses-tu qu'il y aura sur la plaque à pâtisserie? (Il y aura 8 pâtisseries sur la plaque à pâtisserie.)
- Quel bloc de code « met » une pâtisserie sur la plaque à pâtisserie?
Combien de fois est-il présent dans le code?

Le bloc « créer un clone de moi-même » met la pâtisserie sur la plaque à pâtisserie. Ce bloc est présent 4 fois, mais dans une boucle de 2 répétitions, donc $2 \times 4 = 8$.

- Quelles questions te poses-tu?
- Pourrais-tu me dire, dans tes propres mots, ce que dit ce code?

Inviter les élèves à créer un pseudocode qui représente le code de Maxime afin de pouvoir prédire ce qui arrivera lorsque le code sera exécuté. Puis, leur demander si le code de Maxime produira le résultat souhaité.

Exemples de réflexions :

- Je remarque une boucle, ce qui veut dire qu'une série de blocs est répétée.
- Je remarque que des blocs de variable sont utilisés. Je ne me suis jamais servi de variables.
- Je pense que 8 objets seront créés parce que la boucle se répète 2 fois. Dans la boucle, il y a l'événement « faire un clone » qui se répète 4 fois, $2 \times 4 = 8$.
- Je me demande comment je vais créer une variable.

Exemple de pseudocode possible :

Aller à la position de départ.
Mettre la variable « rangée » à zéro.
Montrer l'objet.
Répéter 2 fois.
Avancer de 70 pas. Faire une copie du <i>sprite</i> .
Avancer de 70 pas. Faire une copie du <i>sprite</i> .
Avancer de 70 pas. Faire une copie du <i>sprite</i> .
Avancer de 70 pas. Faire une copie du <i>sprite</i> .
Retourner à la position de départ.
Ajouter 1 à la variable de la rangée, donc monter de 1.
Ajouter $30 \times$ [valeur de « rangée »], donc $30 \times 1^*$.
Cacher l'objet.

* Cette partie du code déplace le *sprite* au début de la bonne rangée. Dans ce cas-ci, chaque rangée est espacée de 30 pixels. Donc, la première rangée (rangée = 0) commence à la position de départ, la deuxième rangée (rangée = 1) commence à la position de départ + 30, la troisième rangée (rangée = 2) commence à la position de départ + 60, etc.

Note : Des élèves remarqueront peut-être que la partie «Avancer de 70 pas, faire une copie du *sprite*» se répète 4 fois, et demanderont si ce serait possible d'utiliser une boucle pour simplifier le code. C'est en fait possible. Ce genre de structure est élaboré davantage en 4^e année (événements imbriqués). Cela étant dit, voici à quoi pourrait ressembler le code comportant cette structure, au cas où ce serait une des pistes de questionnement des élèves.



- Demander aux élèves de recréer le code en utilisant le logiciel choisi.
- Inviter les élèves à modifier le code de Maxime pour qu'il comprenne 24 pâtisseries, comme souhaité au départ.

Note : Cela sera une bonne occasion d'essayer différentes fonctions et de modeler l'approche essai et erreur en appliquant le processus d'élimination ou le processus d'essais systématiques. Encourager les élèves à sauvegarder le code aussitôt qu'il est fonctionnel et à en faire une copie à partir de laquelle différentes fonctions pourront être essayées et l'effet de la modification d'un bloc pourrait être observé. De cette façon, l'élève pourra toujours retourner à son code fonctionnel (le risque associé aux erreurs est donc très faible, ce qui encouragera l'expérimentation).

Réponse possible

Ce code met 8 biscuits sur la plaque à pâtisserie (2 répétitions de 4 biscuits). Je sais que chaque répétition de plus ajoutera 4 pâtisseries sur la plaque à pâtisserie. Je peux donc faire une addition répétée de 4 jusqu'à ce que j'atteigne 24 pâtisseries.

Alors, $4 + 4 + 4 + 4 + 4 + 4 = 24$, j'ai donc besoin de modifier la boucle de manière à avoir 6 répétitions ($6 \times 4 = 24$).

Exemple de code :



```
quand est cliqué
  aller à x: -200 y: -150
  mettre Rangée à 0
  montrer
  répéter 6 fois
    créer un clone de moi-même
    avancer de 70 pas
    créer un clone de moi-même
    avancer de 70 pas
    créer un clone de moi-même
    avancer de 70 pas
    créer un clone de moi-même
    avancer de 70 pas
  aller à x: -200 y: -150
  ajouter 1 à Rangée
  ajouter 30 * Rangée à y
  cacher
```

Demander aux élèves de circuler dans la salle de classe afin de voir les codes des autres élèves. Est-ce que les élèves ont choisi de faire les mêmes modifications? Est-ce que le résultat demeure le même?

Exemples de codes possibles (toujours pour 24 biscuits) :

The image displays three Scratch code snippets, each representing a different way to arrange 24 biscuits on a 200x150 coordinate grid. All scripts start with 'quand est cliqué', 'aller à x: -200 y: -150', 'mettre Rangée à 0', and 'montrer'.
1. The first script uses a 'répéter 8 fois' loop. Inside, it creates a clone, moves 70 pixels right, creates another clone, moves 70 pixels right, and repeats this for 8 clones. It then moves to x: -200 y: -150, increments 'Rangée' by 1, and adds $30 \times \text{Rangée}$ to 'y'. It ends with 'cacher'.
2. The second script uses a 'répéter 4 fois' loop. Inside, it creates a clone, moves 70 pixels right, creates another clone, moves 70 pixels right, and repeats this for 4 clones. It then moves to x: -200 y: -150, increments 'Rangée' by 1, and adds $30 \times \text{Rangée}$ to 'y'. It ends with 'cacher'.
3. The third script uses a 'répéter 12 fois' loop. Inside, it creates a clone, moves 70 pixels right, creates another clone, moves 70 pixels right, and repeats this for 12 clones. It then moves to x: -200 y: -150, increments 'Rangée' by 1, and adds $30 \times \text{Rangée}$ to 'y'. It ends with 'cacher'.

Représente $8 \times 3 = 24$

Représente $4 \times 6 = 24$

Représente $12 \times 2 = 24$

Exemples de réflexions :

- Non, les codes des autres élèves sont différents. J'ai choisi de faire 6 rangées de 4 pâtisseries, mais j'ai vu aussi 8 rangées de 3 pâtisseries et 4 rangées de 6 pâtisseries.
 - Il y a plusieurs multiplications qui font 24, mais je pense que 4 rangées de 6 pâtisseries est la meilleure multiplication pour bien utiliser l'espace.
 - Les résultats sont pareils, il y a toujours 24 pâtisseries sur la plaque à pâtisserie.
 - Les résultats sont différents, même s'il y a toujours 24 pâtisseries sur la plaque à pâtisserie, car la disposition des pâtisseries est différente.
- Demander aux élèves de créer un code pour les commandes d'aujourd'hui.

- 20 scones aux canneberges
- 81 biscuits aux raisins
- 47 chaussons aux framboises

- Encourager les élèves à comparer leurs représentations à celles des autres élèves.

Réponses possibles

20 scones aux canneberges :

- J'ai utilisé la multiplication 2×10 et Valérie a utilisé la multiplication 4×5 .
- Je crois que la disposition rectangulaire de 2×10 n'est pas aussi efficace que celle de 4×5 , puisqu'on gaspille beaucoup d'espace sur la plaque à pâtisserie.
- Je remarque qu'il y a plusieurs façons de représenter 20 biscuits en utilisant la disposition rectangulaire : 4×5 , 5×4 , 2×10 et 10×2 .

81 biscuits aux raisins :

- Il y a juste une façon de représenter ce nombre, puisqu'il y a juste une multiplication qui a un produit de 81, soit 9×9 .

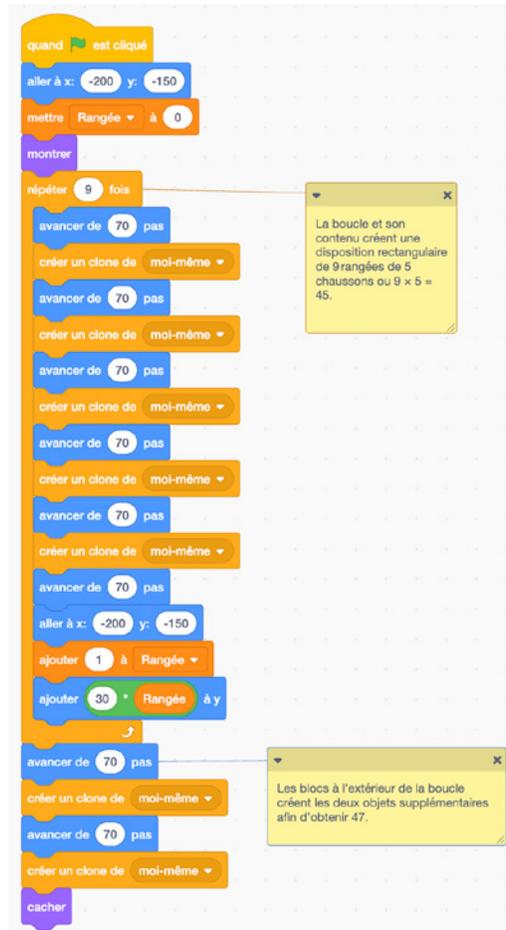
Note : Les faits de multiplication jusqu'à 10×10 sont priorités dans cette situation d'apprentissage. Toutefois, il est possible d'accepter les réponses 3×27 ou 27×3 si celles-ci sont proposées.

- Je remarque que les pâtisseries forment un carré.

47 chaussons aux framboises :

Il n'y a pas deux nombres qui, multipliés, donnent 47, mais je sais que $9 \times 5 = 45$. Je peux d'abord créer un code pour mettre 45 chaussons sur la plaque à pâtisserie (9 rangées de 5 chaussons) et ensuite en ajouter deux autres sur le côté.

Exemple de code :

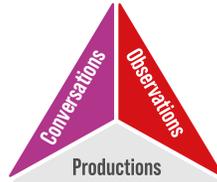


Note : Ce même raisonnement pourrait être utilisé avec d'autres faits numériques, comme $6 \times 7 = 42$, et 5 autres font 47.

OBSERVATIONS POSSIBLES	PISTES D'INTERVENTION
<p>L'élève ne réussit pas à utiliser correctement la boucle.</p>	<p>Utiliser différentes approches pour expliquer la structure. La boucle, par exemple, ressemble à une bouche, et tout ce qui est dans la bouche est mâchée un certain nombre de fois.</p> <p>Demander à l'élève si la boucle contient tous les éléments qui devront être répétés (parfois, il y a des oublis, surtout vers la fin de la boucle).</p> <p>Demander à l'élève de nommer la multiplication que l'on tente de représenter avec le code. Où peux-tu trouver les deux nombres multipliés dans ton code?</p> <p>Fournir du matériel pour que l'élève puisse « faire le robot » dans une activité de codage débranché en suivant le code tel qu'il est indiqué; par exemple, un jeton = une pâtisserie que l'on doit mettre sur une plaque à pâtisserie rectangulaire. L'élève peut ensuite traduire ses actions en code pour l'ordinateur.</p>
<p>L'élève ne modifie pas les déplacements, alors les pâtisseries ne vont pas toutes sur la plaque à pâtisserie ou ne la remplissent pas.</p>	<p>Demander à l'élève de démarrer son code, puis poser des questions sur ses observations.</p> <p>Demander à l'élève si la surface a été utilisée de façon efficace.</p> <p>Demander à l'élève s'il est possible de voir toutes les pâtisseries sur la plaque à pâtisserie.</p> <p>Revoir avec l'élève, dans le contexte du codage, la terminologie suivante : avancer, aller à et ajouter.</p>
<p>L'élève oublie de cacher l'objet à la fin du code. Exemple :</p> 	<p>Demander à l'élève de bien compter le nombre de pâtisseries sur la plaque à pâtisserie.</p> <p>Demander à l'élève d'expliquer son raisonnement pour résoudre son problème avant de chercher le bloc voulu.</p> <p>Note : Il existe plusieurs solutions à ce problème; par exemple, l'élève pourrait ajouter un bloc à la fin du code, qui renvoie le <i>sprite</i> à son endroit d'origine, ce qui crée le même résultat visuel que « cacher » le <i>sprite</i>.</p>
<p>L'élève oublie le bloc « créer un clone de moi-même ».</p>	<p>Demander à l'élève si les pâtisseries peuvent être comptées à la suite du démarrage de son code.</p> <p>Demander à l'élève d'expliquer son raisonnement pour résoudre son problème avant de chercher un bloc.</p> <p>Note : Il existe aussi un bloc nommé « estampiller » qui peut produire un effet semblable.</p>

Objectivation (15 minutes)

L'évaluation
peut se faire par les...



Choisir quelques élèves du groupe-classe et leur demander de présenter ou d'expliquer leur code (en utilisant un pseudocode). Pendant cette discussion, demander aux autres élèves du groupe-classe de faire ressortir des ressemblances et des différences entre les codes.

Faire un retour sur l'utilisation efficace de boucles dans un code.

Consolidation

Proposer aux élèves de créer des codes afin de représenter des formes géométriques régulières. Si le logiciel de codage utilisé a une option pour tracer des lignes (par exemple, le « stylo » dans Scratch), les élèves peuvent tracer des formes en utilisant des déplacements et des rotations de *sprite*. Les boucles permettront de réduire la quantité de code pour atteindre un but.

Pour coder une trajectoire carrée, par exemple, il faut connaître les propriétés du carré (4 angles droits et 4 côtés congrus). Le code pour la trajectoire pourrait donc ressembler à ceci :



L'ajout de boucles peut réduire le code à ceci :



Les élèves peuvent appliquer ce même processus afin de créer un code pour des rectangles.

ÉLÉMENTS À CONSIDÉRER

Liens avec les autres domaines mathématiques (événements répétitifs)

Nombres

B2.6 Représenter la multiplication de nombres jusqu'à 10×10 et la division de nombres jusqu'à $100 \div 10$ à l'aide d'une variété d'outils et de schémas, y compris des dispositions rectangulaires.

En utilisant une boucle, l'élève peut faire le lien entre l'addition répétée et la multiplication. En codant des déplacements et des reproductions de *sprites*, l'élève peut cheminer d'une représentation de groupes d'objets vers la disposition rectangulaire.

Algèbre

C1.1 Reconnaître et décrire les éléments et les opérations qui se répètent dans diverses suites (numériques et non numériques), y compris des suites trouvées dans la vie quotidienne.

Les séquences de code qui se trouvent à l'intérieur d'une boucle représentent le motif d'une suite à motif répétée. Les élèves peuvent donc utiliser du pseudocode pour décrire l'action à l'intérieur de la boucle.

Sens de l'espace

E2.8 Utiliser des unités de mesure non conventionnelles appropriées pour mesurer l'aire et expliquer l'incidence du chevauchement et des espaces entre les unités sur l'exactitude de la mesure.

En utilisant des événements répétitifs pour faire en sorte qu'une multiplication est représentée en disposition rectangulaire, l'élève peut faire le lien entre le nombre d'objets disposés en rectangle et l'aire.

Différenciation pédagogique et conception universelle de l'apprentissage

- Donner les blocs débranchés aux élèves pour les replacer de manière à reproduire le code.
- Donner un pseudocode aux élèves pour pouvoir s'en servir comme plan.
- Proposer aux élèves un code de base pour le modifier selon le nombre de pâtisseries souhaitées.
- Ce code se prête aussi bien aux événements imbriqués, une structure un peu plus complexe, mais qui peut rendre le code plus facile à interpréter. Les élèves qui sont plus à l'aise avec le code pourront utiliser cette structure en mettant des boucles à l'intérieur de boucles.
- Ajouter certains paramètres, par exemple, un maximum de pâtisseries pouvant aller sur une plaque à pâtisserie, ce qui crée une situation mathématique plus complexe à représenter à l'aide du code.

Annexe – Mise en situation



Situation d'apprentissage – 4^e année

Titre : À la poursuite des suites!

Durée : 100 minutes

Sommaire

Dans cette situation d'apprentissage, l'élève utilise la puissance des structures de contrôle imbriquées afin de créer et de prolonger des suites à motif répété et des suites croissantes.

Attentes et contenus d'apprentissage ciblés

Algèbre

- C1.** Reconnaître, décrire, prolonger et créer une variété de suites, y compris des suites trouvées dans la vie quotidienne, et faire des prédictions à leur sujet.
 - C1.1** Reconnaître et décrire des suites à motif répété et des suites croissantes, y compris des suites trouvées dans la vie quotidienne.
 - C1.2** Créer des suites à motif répété et des suites croissantes à l'aide d'une variété de représentations, y compris des tables de valeurs et des représentations graphiques, et établir des liens entre les différentes représentations.
- C3.** Résoudre des problèmes et créer des représentations de situations mathématiques de façon computationnelle à l'aide de concepts et d'habiletés en codage.
 - C3.1** Résoudre des problèmes et créer des représentations de situations mathématiques de façon computationnelle en écrivant et en exécutant des codes, y compris des codes comprenant des événements séquentiels, simultanés, répétitifs et imbriqués.
 - C3.2** Lire et modifier des codes donnés, y compris des codes comprenant des événements séquentiels, simultanés, répétitifs et imbriqués, et décrire l'incidence des changements sur les résultats.

Pratiques pédagogiques à fort impact en mathématiques à privilégier

Conversations mathématiques

La mise en situation ci-dessous est l'occasion d'avoir une conversation mathématique avec les élèves du groupe-classe. Cette occasion leur permettra d'échanger leurs idées, de montrer leur compréhension des suites, de raisonner et de prouver leur raisonnement quant au résultat prédit à la suite de la lecture d'un code. Les élèves pourront établir des liens avec des expériences vécues et échanger leurs connaissances relatives aux différentes suites possibles et aux divers événements possibles dans un code.

Regroupements flexibles

Les regroupements flexibles au cours de la création ou de la modification de codes facilitent, entre autres, la communication et la collaboration. Les élèves peuvent apprendre les uns des autres et ainsi faire évoluer leur réflexion mathématique. Tout en tenant compte du profil des élèves, les regroupements peuvent être aléatoires, hétérogènes ou homogènes, selon les besoins. Au cours du déroulement de cette situation d'apprentissage, des regroupements flexibles sont suggérés afin de rendre l'apprentissage optimal pour les élèves.

Enseignement en petits groupes

L'enseignement en petits groupes est une pratique pédagogique visant à faire progresser l'apprentissage des élèves. Tout le long de la création ou de la modification d'un code, il est important de former de petits groupes d'élèves afin de leur fournir l'enseignement nécessaire selon les besoins. L'enseignement en petits groupes permet de revoir des élèves des concepts mathématiques qui appuient leur nouvel apprentissage. Les élèves pourront donc renforcer leur compréhension des concepts à l'étude.

Enseignement explicite

Lorsqu'un code ne donne pas le résultat attendu, le personnel enseignant pourrait demander aux élèves de le lui expliquer. C'est alors un moment propice pour offrir un enseignement explicite relatif à la difficulté observée. Cet enseignement permet de vérifier la compréhension des élèves et de leur fournir une rétroaction immédiate.

Connaissances et habiletés préalables

- L'élève de 4^e année aura vu, dans les années précédentes, la façon de créer une séquence de code à l'aide d'un logiciel de codage par blocs. Si ce n'est pas le cas, une introduction au logiciel de codage et du temps d'exploration seront nécessaires pour le bon fonctionnement de cette activité.
- Plusieurs logiciels de codage utilisent le plan cartésien pour déterminer la position et les déplacements des objets. Il serait important de voir l'orientation des axes x et y , par exemple, si l'on ajoute à la valeur de x , l'objet avancera, tandis que, si l'on ajoute à la valeur de y , l'objet montera. Une présentation de ce concept pourrait être utile.

Résultats d'apprentissage

À la fin de cette situation d'apprentissage, l'élève pourra :

- utiliser la terminologie à l'étude telle que *code*, *sprite*, *événements séquentiels*, *événements simultanés*, *événements répétitifs*, *événements imbriqués*, *suite à motif répété*, *motif*, *suite croissante*, *table de valeurs*;
- décrire des suites;
- créer des suites à motif répété et des suites croissantes en écrivant et en exécutant des codes comprenant des événements séquentiels, des événements simultanés, des événements répétitifs et des événements imbriqués;
- lire et modifier des codes comprenant des événements séquentiels, des événements simultanés, des événements répétitifs et des événements imbriqués.

Critères d'évaluation selon les grilles d'évaluation du rendement

Connaissance et compréhension

- L'élève connaît les blocs d'un logiciel de programmation par blocs et comprend leur fonctionnement.
- L'élève reconnaît le motif dans une suite à motif répétée (avec du matériel ou dans une séquence de code).

Habiletés de la pensée

- L'élève planifie la rédaction ou la modification d'un code en rédigeant un pseudocode représentatif.
- L'élève prédit le résultat de l'exécution d'un code.
- L'élève organise une représentation d'une suite croissante dans un tableau.
- L'élève modifie un code pour intégrer une structure imbriquée (une boucle dans une boucle) de manière à rendre le code plus pertinent à une situation mathématique spécifique.

Communication

- L'élève utilise de manière appropriée le vocabulaire lié au codage (code, blocs, *sprite*, personnage, événements séquentiels, événements simultanés, événements répétitifs, événements imbriqués) et aux suites (suite à motif répété, suite croissante, rang, règle).
- L'élève rédige du code et du pseudocode en respectant le format et la syntaxe (par exemple, l'utilisation de retraits pour démarquer ce qui se trouve dans une boucle).

Mise en application

- L'élève modifie son code afin que celui-ci représente une nouvelle situation semblable au problème d'origine.
- L'élève utilise ses connaissances des suites à motif répété afin de déterminer le motif d'une suite créée par une séquence de code.

Matériel requis

- ordinateur muni d'un logiciel de codage par blocs approprié selon le niveau des élèves
- objets représentant des suites (jetons bicolores, tuiles de couleur, etc.)
- [annexe \[Mise en situation\]](#)

Note : Les exemples dans cette situation d'apprentissage ont été créés à partir de Scratch.

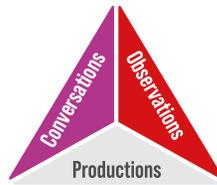
Vocabulaire mathématique

vocabulaire spécifique au codage (code, *sprite*, séquence, boucle), événements séquentiels, événements simultanés, événements répétitifs, événements imbriqués, suite à motif répété et suite croissante, motif, terme, rang, table de valeurs

CONTENU

Mise en situation (15-20 minutes)

L'évaluation
peut se faire par les...



Montrer l'image ci-dessous ([annexe Mise en situation](#)) au groupe-classe et animer une discussion de type « je remarque, je me demande ». Prendre en note les remarques et les questions des élèves.



Afficher la table de valeurs ci-dessous à l'écran et demander aux élèves si l'ajout de celle-ci affecte les remarques et les questions échangées lorsqu'on ne voyait que l'image. Ajouter les nouvelles remarques et les nouvelles questions à la liste.

MOIS	ARGENT AJOUTÉ À LA TIRELIRE	SOMME D'ARGENT DANS LA TIRELIRE
Janvier	5 \$	5 \$
Février	5 \$	10 \$
Mars	5 \$	15 \$
Juin	5 \$?

Exemples de réflexions :

- Je remarque que l'on ajoute toujours la même somme à la tirelire.
- Je remarque que l'on saute directement de mars à juin. Il manque des mois.
- Je remarque que la somme d'argent augmente avec le temps.
- Je me demande combien d'argent sera dans la tirelire au mois de juin.
- Je me demande si la somme d'argent restera la même tout le long de l'année.
- Je me demande si la personne dépensera de l'argent de sa tirelire avant le mois de juin.

Fournir aux élèves du matériel de manipulation et leur demander de représenter les données afin de déterminer la somme d'argent dans la tirelire au mois de juin.

Exemple de représentation possible avec des jetons :



1	2	3	4	5	6
Janvier	Février	Mars	Avril	Mai	Juin

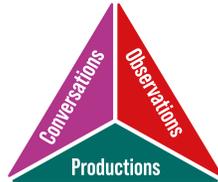
Vérifier si les prédictions des élèves quant à la somme d'argent dans la tirelire étaient justes. Si des élèves ont obtenu des résultats différents, vérifier si c'est à cause des suppositions qui ont été faites (par exemple, un mois, la personne reçoit plus de 5 \$, ou un mois, la personne décide de faire une dépense).

Expliquer aux élèves que la représentation avec du matériel fonctionne relativement bien dans cette situation. Mais, si l'on voulait déterminer les épargnes sur une plus longue durée, cela pourrait prendre beaucoup de temps à représenter. Afin de rendre ce processus plus rapide et convivial, on peut rédiger un code qui généralisera la situation et **qui nous permettra de voir** les épargnes plus loin dans l'avenir.

Note : Il n'est pas tant question ici de la « réponse », mais plutôt de la représentation de la situation mathématique. Des élèves remarqueront peut-être rapidement que l'on peut multiplier le nombre de mois (le rang) par 5 pour obtenir la somme d'argent épargnée au total. Cependant, cela ne nous montre pas la croissance de la somme d'argent dans la tirelire.

Déroulement (60 minutes, en plus du temps alloué à l'exploration du logiciel, au besoin)

L'évaluation
peut se faire par les...



Présenter aux élèves une séquence de code qui représente la situation de la tirelire. Ne pas leur montrer le résultat de l'exécution du code (ne pas cliquer sur le drapeau vert).

Note : Dans les exemples fournis, les notes en jaune sont des explications du fonctionnement du code et devraient être omises lorsque le code est présenté aux élèves.

Demander aux élèves d'étudier le code afin de prédire ce qu'il fera lorsqu'on l'exécute. La rédaction d'un pseudocode pourrait aussi aider l'élève à prédire le résultat de l'exécution et à mieux comprendre à quoi servent les divers blocs.

Exemple de code représentant le problème de la tirelire à l'aide de jetons :

CODE	RÉSULTAT DE L'EXÉCUTION
<pre>quand est cliqué effacer tout aller à x: -200 y: -160 attendre 1 secondes estampiller ajouter 20 à y attendre 1 secondes estampiller</pre>	

Exemple d'un pseudocode représentatif du code présenté :

Condition de départ – Drapeau vert
Objet : jeton qui se rend au coin inférieur gauche
Pause de 1 seconde
La position du jeton est estampillée.
Objet : jeton qui se déplace vers le haut
Pause de 1 seconde
La position du jeton est estampillée.
Objet : jeton qui se déplace vers le haut
Pause de 1 seconde
La position du jeton est estampillée.
Objet : jeton qui se déplace vers le haut
Pause de 1 seconde
La position du jeton est estampillée.
Objet : jeton qui se déplace vers le haut
Pause de 1 seconde
La position du jeton est estampillée.

Après avoir alloué suffisamment de temps afin que les élèves fassent leurs prédictions, le code peut être exécuté. Ensuite, les élèves pourront comparer leurs prédictions avec le résultat obtenu.

Demander aux élèves ce qui ressort le plus du code qui vient d'être exploré. Il existe plusieurs réponses possibles, mais tenter de diriger le questionnement vers les éléments qui se répètent. Le code, tel qu'il est présenté, est une suite à motif répété. Le motif de la suite est le suivant :

[Déplacement du jeton], [Pause de 1 seconde], [Estampiller]

Ce motif se répète tout le long du code. Celui-ci peut donc être réduit avec une boucle.

Note : Si les élèves ne connaissent pas le concept de boucle, il serait avantageux de leur faire explorer le logiciel de codage en utilisant la boucle afin que les élèves se familiarisent avec cette structure avant de continuer.

Exemple de code réduit avec la boucle :

The image shows a Scratch script on a grid background. The script starts with an orange 'when clicked' block, followed by a purple 'show' block, a green 'clear all' block, a blue 'go to x: -200 y: -160' block, an orange 'repeat 5 times' loop containing an orange 'wait 1 seconds' block, a green 'stamp' block, a blue 'add 20 to y' block, and a purple 'hide' block. Three yellow callout boxes with arrows point to specific parts of the code:

- The first callout points to the 'go to x: -200 y: -160' block and contains the text: "Le déplacement à la position initiale n'est pas dans la boucle. Ce déplacement se produit une seule fois, au début du code."
- The second callout points to the 'repeat 5 times' loop and contains the text: "Cette section à l'intérieur de la 'boucle' de la boucle sera répétée 5 fois."
- The third callout points to the 'hide' block and contains the text: "L'objet demeure sur l'écran après la fin de l'exécution du code, ce qui donne l'impression d'un objet de trop. En le cachant, on obtient le bon nombre d'étapes pour représenter la situation. N'oubliez pas d'ajouter un bloc 'montrer' au début du code, sinon l'objet demeurera caché."

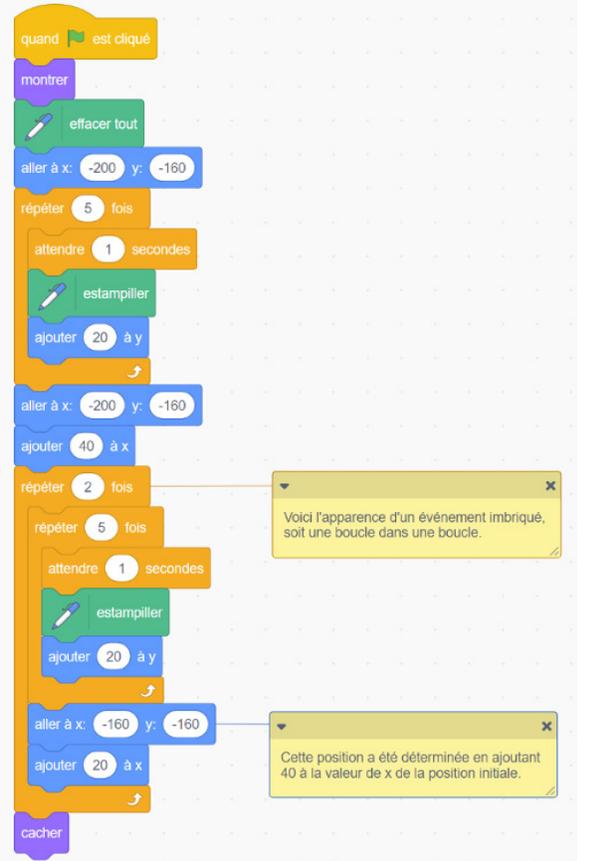
À noter que le résultat de ce code donne le même résultat que le code plus long sans la boucle. Faire remarquer aux élèves que le code produit aussi une représentation identique à la figure au rang 1 – Janvier, créée lors de la mise en situation. Leur demander de modifier le code afin qu'il reproduise la situation mathématique de la tirelire. Il existe plusieurs façons de poursuivre.

1. Si les élèves commencent leur exploration du codage, il serait avantageux de leur montrer, par essais et erreurs, la façon de modifier le code pour insérer des boucles. En modélisant leurs suggestions, les élèves peuvent travailler ensemble afin de trouver une solution au problème.
2. Si les élèves ont un peu plus d'expérience avec le logiciel de codage, des équipes de deux ou même du travail individuel seraient avantageux pour que les élèves découvrent à leur manière des séquences de code qui représenteront la situation mathématique.
3. Si le degré d'expérience et d'aisance des élèves varie beaucoup dans le groupe-classe, des regroupements aléatoires ou réfléchies pourraient les mettre dans une situation d'apprentissage par les pairs. Le personnel enseignant pourrait également profiter de cette occasion pour faire de l'enseignement explicite en petits groupes selon les besoins des élèves.

Voici un exemple de pseudocode et de code qui représente la situation de la tirelire avec des boucles :

Condition de départ – Drapeau vert
<i>Sprite</i> : jeton qui se rend au coin inférieur gauche (position initiale 1)
Répéter 5 fois.
Pause de 1 seconde
La position du jeton est estampillée.
<i>Sprite</i> : jeton qui se déplace vers le haut de 20 unités
<i>Sprite</i> : jeton qui retourne à la position initiale 1
<i>Sprite</i> : jeton qui se déplace vers la droite de 30 unités (position initiale 2)
Répéter 2 fois.
Répéter 5 fois.
Pause de 1 seconde
La position du jeton est estampillée.
<i>Sprite</i> : jeton qui se déplace vers le haut de 20 unités
<i>Sprite</i> : jeton qui retourne à la position initiale 2
<i>Sprite</i> : jeton qui se déplace vers la droite de 2 unités

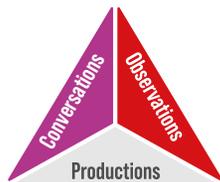
Ce pseudocode représente 2 mois (position initiale 1 et position initiale 2). En déterminant la tendance, on peut prolonger cette séquence de code, qui ressemble, encore une fois, à une suite, pour représenter plusieurs mois.

CODE	RÉSULTAT DE L'EXÉCUTION
 <p>The code consists of the following blocks:</p> <ul style="list-style-type: none">when green flag clickedshowcleargo to x: -200 y: -160repeat 5 times:<ul style="list-style-type: none">wait 1 secondstampadd 20 to ygo to x: -200 y: -160add 40 to xrepeat 2 times:<ul style="list-style-type: none">repeat 5 times:<ul style="list-style-type: none">wait 1 secondstampadd 20 to ygo to x: -160 y: -160add 20 to xhide <p>Two callouts are present:</p> <ul style="list-style-type: none">Voici l'apparence d'un événement imbriqué, soit une boucle dans une boucle.Cette position a été déterminée en ajoutant 40 à la valeur de x de la position initiale.	 <p>The execution result shows a 3x3 grid of blue hearts on a white background.</p>

OBSERVATIONS POSSIBLES	PISTES D'INTERVENTION
L'élève ne comprend pas le fonctionnement de la boucle.	<p>Montrer à l'élève un exemple de code plus simple avec une boucle.</p> <p>Demander à l'élève de rédiger un code avec un seul événement dans une boucle et d'observer le comportement.</p>
L'élève ne comprend pas le fonctionnement d'une boucle imbriquée dans une boucle.	<p>Exécuter le code avec chaque boucle seule afin que l'élève puisse voir son effet sans qu'elle soit imbriquée.</p> <p>Rappeler à l'élève l'effet multiplicatif des boucles imbriquées pour donner un repère plus accessible.</p>
L'élève ne reconnaît pas la règle de la suite (rang 1 = une colonne de 5, rang 2 = 2 colonnes de 5, etc.).	<p>Remettre à l'élève du matériel concret afin de représenter la situation et d'observer la règle.</p> <p>Demander à l'élève de verbaliser ses observations quant à la règle de la suite. Parfois, c'est en conversant que vient la compréhension.</p>
Le code de l'élève ne ressemble aucunement au pseudocode rédigé.	<p>Les logiciels de codage utilisent souvent des codes de couleur pour les blocs. Demander à l'élève d'ajouter de la couleur, à l'aide de surligneurs, à son pseudocode en respectant les couleurs utilisées dans le logiciel en question. Le pseudocode et le code peuvent ensuite être comparés afin de trouver les inconsistances.</p>
Le code de l'élève ne fonctionne pas.	<p>Demander à l'élève s'il s'agit d'une erreur de syntaxe (le code ne fonctionne pas du tout) ou d'une erreur de logique (le code fonctionne, mais ne produit pas le résultat désiré).</p> <p>Créer des partenariats de vérification de codes entre les élèves du groupe-classe. Parfois, il est plus facile de trouver une erreur si l'on n'a pas créé le code.</p> <p>Demander à l'élève d'expliquer le rôle de chaque séquence de code, surtout les séquences d'événements imbriqués, avec du pseudocode.</p>

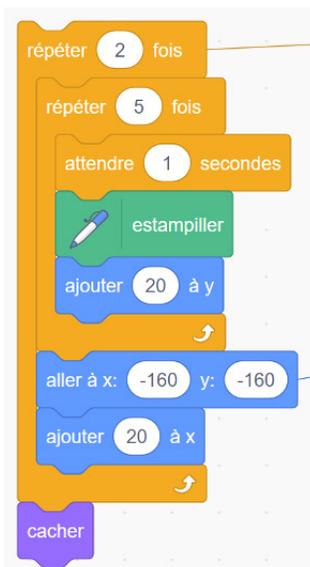
Objectivation (20 minutes)

L'évaluation
peut se faire par les...



Questionner les élèves sur l'utilisation de la boucle dans la boucle (événements imbriqués).

- Quelles parties du visuel sont le résultat de la boucle interne?
Les cinq objets dans chaque colonne.
- Quelles parties du visuel sont le résultat de la boucle externe?
Le nombre de colonnes d'objets par figure.
- Qu'arriverait-il si l'on inversait les nombres des répétitions dans la boucle interne et externe de la séquence suivante?



Au lieu de deux colonnes de cinq objets, on aurait cinq colonnes de deux objets.

Demander aux élèves si cette situation représente un revenu, des épargnes ou un investissement. Les réponses peuvent varier selon l'interprétation du problème. Une discussion sur la terminologie serait utile si ces termes n'ont pas encore été vus en classe.

Réponses possibles

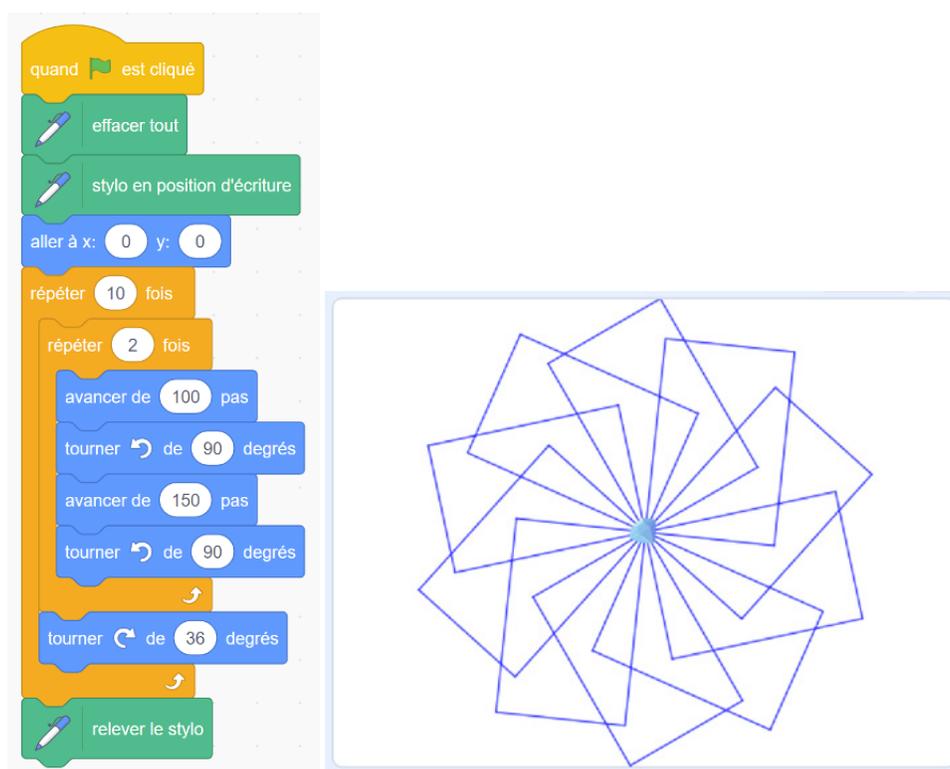
- C'est un revenu mensuel, car la personne s'est fait payer 5 \$ par mois, la somme ajoutée à la tirelire.
- Ce sont des épargnes, car la personne choisit de mettre de côté une certaine somme d'argent par mois.
- Ce n'est pas un investissement, la seule chose qui affecte la valeur de la tirelire, ce sont les dépôts de la personne.

Consolidation

- Proposer aux élèves différentes situations à résoudre à l'aide d'événements imbriqués.

Exemples :

- La personne fait les mêmes versements qu'à la situation initiale (versements réguliers de 5 \$ par mois), mais dépense 10 \$ tous les trois mois.
 - La personne dépose plutôt 1 \$ de plus par mois (5 \$, 6 \$, 7 \$, 8 \$, etc.).
 - La personne double la valeur de sa tirelire chaque mois (somme totale dans la tirelire : 5 \$, 10 \$, 20 \$, 40 \$).
- Inviter les élèves à retourner dans le logiciel de codage afin de représenter une nouvelle situation de leur choix en se basant sur leur code existant.
 - Demander aux élèves de créer des motifs géométriques en utilisant des boucles imbriquées. En utilisant les propriétés d'un rectangle, les élèves peuvent créer un code qui dessine un rectangle, puis le fait tourner légèrement avant d'en dessiner un autre, un certain nombre de fois. Le code et le motif résultant pourraient ressembler à ceci :



- Demander aux élèves d'expérimenter en modifiant le nombre de répétitions de la boucle extérieure et le nombre de degrés de rotation à chaque répétition afin de créer de nouveaux motifs.

ÉLÉMENTS À CONSIDÉRER

Liens avec les autres domaines mathématiques

Nombres

B2.2 Se rappeler les faits de multiplication de 1×1 à 10×10 et les faits de division associés, et démontrer sa compréhension de ces faits.

Les boucles imbriquées créent des représentations de multiplications plus complexes qu'avec des boucles simples. On peut, par exemple, obtenir 24 répétitions d'une séquence de code en le plaçant dans une boucle de 4 répétitions, et placer la boucle de 4 répétitions dans une boucle de 6 répétitions.

Algèbre

C1.1 Reconnaître et décrire des suites à motif répété et des suites croissantes, y compris des suites trouvées dans la vie quotidienne.

C1.2 Créer des suites à motif répété et des suites croissantes à l'aide d'une variété de représentations, y compris des tables de valeurs et des représentations graphiques, et établir des liens entre les différentes représentations.

C2.1 Déterminer et utiliser des symboles comme variables dans des expressions et des équations.

Les événements imbriqués créent des suites croissantes. Une première boucle définit le motif de la suite, et on la place dans une deuxième boucle qui détermine le nombre de répétitions. Par la suite, il s'agit d'ajouter des blocs de code et des variables pour modifier la règle de la suite.

Sens de l'espace

E2.5 Utiliser la structure en rangées et en colonnes d'une disposition rectangulaire pour mesurer l'aire d'un rectangle et pour démontrer que l'aire d'un rectangle peut être calculée en multipliant sa base par sa hauteur.

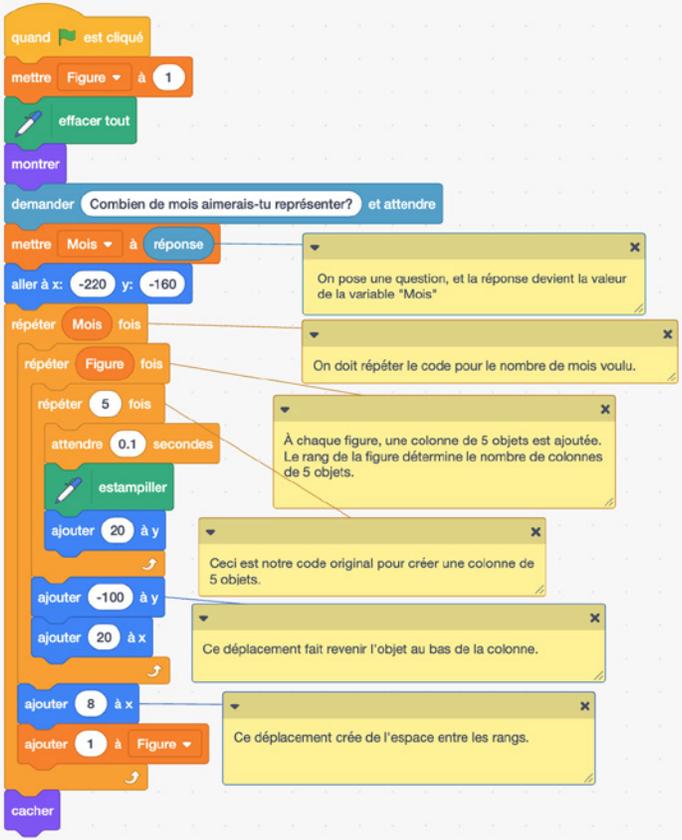
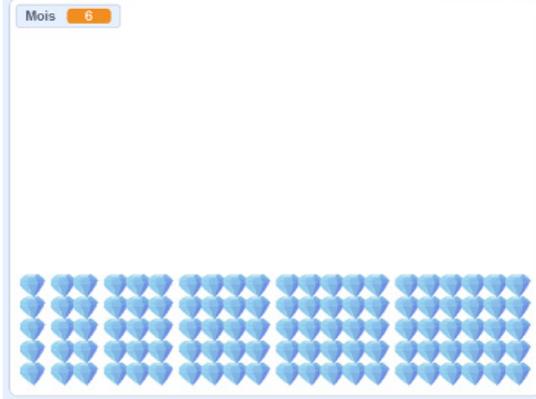
Les boucles imbriquées peuvent aider à représenter la disposition rectangulaire. La boucle interne représente le nombre d'éléments dans une rangée, et la boucle externe, le nombre de rangées.

Littératie financière

F1.3 Expliquer les concepts de dépense, d'épargne, de revenu, d'investissement et de don, et déterminer les principaux éléments à considérer dans la prise de décisions simples.

Différenciation pédagogique et conception universelle de l'apprentissage

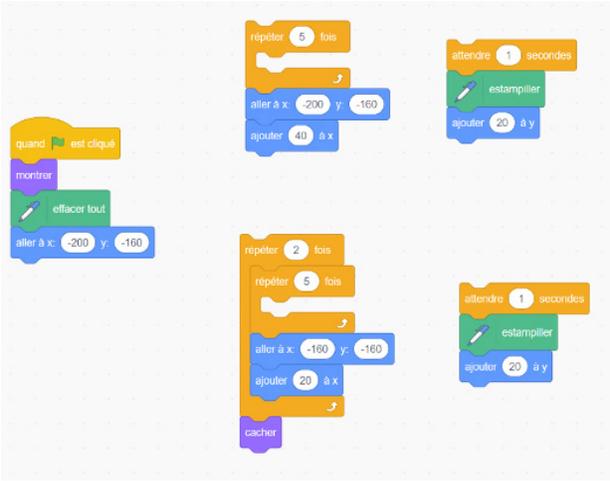
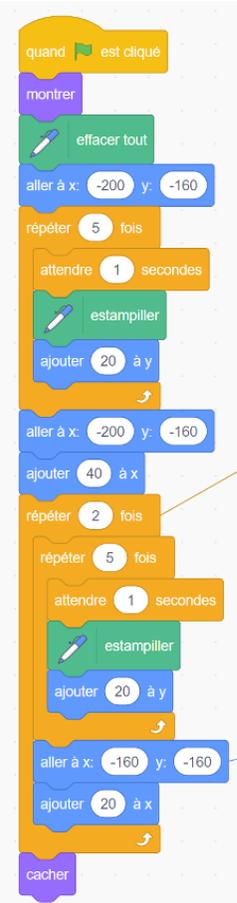
L'élève à l'aise avec le codage en blocs pourrait utiliser des variables afin d'automatiser encore plus le code. Voir ci-dessous un exemple de code qui demande le nombre de mois à représenter. L'élève utilise la réponse afin de représenter la situation mathématique. Le code pourrait aussi être étudié afin de prédire le résultat ou fourni afin que l'élève le modifie selon ses objectifs personnels.

CODE	RÉSULTAT DE L'EXÉCUTION
 <p>The code block contains the following logic:</p> <ul style="list-style-type: none">When clicked, set Figure to 1, clear the stage, and show it.Ask the user: "Combien de mois aimerais-tu représenter?" and wait for a response.Set the variable "Mois" to the user's response.Move to coordinates (-220, -160).Repeat "Mois" times:<ul style="list-style-type: none">Repeat "Figure" times:<ul style="list-style-type: none">Repeat 5 times:<ul style="list-style-type: none">Wait 0.1 seconds.Stamp a heart.Move 20 units up (y).Move -100 units up (y).Move 20 units right (x).Move 8 units right (x).Move 1 unit down (Figure). Hide the stage. <p>Annotations in the image:</p> <ul style="list-style-type: none">"On pose une question, et la réponse devient la valeur de la variable 'Mois'" points to the 'demander' block."On doit répéter le code pour le nombre de mois voulu." points to the outer 'répéter' block."À chaque figure, une colonne de 5 objets est ajoutée. Le rang de la figure détermine le nombre de colonnes de 5 objets." points to the inner 'répéter' block."Ceci est notre code original pour créer une colonne de 5 objets." points to the 'répéter 5 fois' block."Ce déplacement fait revenir l'objet au bas de la colonne." points to the 'ajouter -100 à y' block."Ce déplacement crée de l'espace entre les rangs." points to the 'ajouter 1 à Figure' block.	 <p>The execution result shows a window with a variable "Mois" set to 6. Below it, a grid of 30 blue hearts is displayed, arranged in 6 rows and 5 columns.</p>

L'élève qui a de la difficulté à utiliser le logiciel de codage pourrait faire équipe avec des élèves plus à l'aise afin d'apprendre de ses partenaires. Un groupe d'élèves ayant le même degré d'expérience avec le logiciel faciliterait aussi l'enseignement explicite selon les besoins du groupe.

Plusieurs logiciels de codage permettent le partage de projets. Les élèves pourraient recevoir à l'avance certaines séquences de blocs de code assemblés afin de les mettre aux bons endroits pour faire fonctionner le code.

Exemple :

L'ÉLÈVE COMMENCE PAR CECI...	... EN VUE DE CRÉER CECI
	

Annexe – Mise en situation



MOIS	ARGENT AJOUTÉ À LA TIRELIRE	SOMME D'ARGENT DANS LA TIRELIRE
Janvier	5 \$	5 \$
Février	5 \$	10 \$
Mars	5 \$	15 \$
Juin	5 \$?

Situation d'apprentissage – 5^e année

Titre : La machine à probabilité

Durée : 125 minutes

Sommaire

Dans cette situation d'apprentissage, l'élève crée une machine à utiliser le lancer aléatoire d'un dé afin de montrer que plus le nombre d'essais est élevé, plus la probabilité expérimentale se rapproche de la probabilité théorique.

Attentes et contenus d'apprentissage ciblés

Algèbre

C3. Résoudre des problèmes et créer des représentations de situations mathématiques de façon computationnelle à l'aide de concepts et d'habiletés en codage.

C3.1 Résoudre des problèmes et créer des représentations de situations mathématiques de façon computationnelle en écrivant et en exécutant des codes, y compris des codes comprenant des instructions conditionnelles et d'autres structures de contrôle.

C3.2 Lire et modifier des codes donnés, y compris des codes comprenant des instructions conditionnelles et d'autres structures de contrôle, et décrire l'incidence de ces changements sur les résultats.

Données

D2. Décrire la probabilité que des événements se produisent et utiliser cette information pour faire des prédictions.

D2.2 Déterminer et comparer les probabilités théoriques et expérimentales qu'un événement se produise.

Pratiques pédagogiques à fort impact en mathématiques à privilégier

Résultats d'apprentissage, critères d'évaluation et rétroaction descriptive

Affichez, dans la salle de classe, les critères d'évaluation pour que les élèves s'y réfèrent pendant leur travail. Tout le long de la leçon, prenez le temps de circuler et d'observer ce que font les élèves. En profitez pour les interroger sur leur travail et leur donner une rétroaction descriptive relative aux critères d'évaluation affichés.

Conversations mathématiques

Dans la mise en situation, les élèves discutent des probabilités possibles en lançant le dé. Pendant leur exploration des résultats possibles, les élèves écoutent les différentes opinions et expliquent leur raisonnement à leurs pairs. Prenez le temps de circuler et de questionner les élèves sur leurs observations, leurs connaissances ou sur leurs nouveaux objectifs d'apprentissage.

Regroupements flexibles

Les élèves travaillent en groupe afin de favoriser la collaboration. Les expériences en petits groupes sécurisent les élèves et les motivent souvent à prendre des risques. Les élèves peuvent travailler en petits groupes pour créer leur machine à probabilité qui facilitera, en temps et en précision, les 1 000 lancers demandés.

Connaissances et habiletés préalables

Pour être en mesure de réaliser cette situation d'apprentissage, l'élève doit :

- avoir une compréhension de la probabilité expérimentale et de la probabilité théorique;
- pouvoir organiser des données dans un tableau de fréquences relatives (fractions);
- être capable d'utiliser un logiciel de programmation par blocs;
- connaître les multiples des nombres.

Résultats d'apprentissage

À la fin de cette situation d'apprentissage, l'élève pourra :

- écrire et exécuter le code d'un lancer de dé aléatoire dans un logiciel de programmation par blocs;
- intégrer une ou des instructions conditionnelles dans son code;
- comparer les probabilités théoriques et expérimentales d'un événement.

Critères d'évaluation selon les grilles d'évaluation du rendement

Connaissance et compréhension

- L'élève comprend la signification des blocs d'instruction conditionnelle.
- L'élève connaît la différence entre la probabilité théorique et la probabilité expérimentale.
- L'élève détermine les blocs et les actions de chaque bloc utilisé dans les événements répétitifs et imbriqués.
- L'élève utilise un logiciel de codage afin de représenter un résultat aléatoire.

Habiletés de la pensée

- L'élève écrit un pseudocode avec exactitude.
- L'élève compare les probabilités théoriques et expérimentales d'obtenir un nombre pair.
- L'élève modifie/débogue son code avec précision lorsqu'il ne fonctionne pas.

Communication

- L'élève explique avec clarté l'incidence de certains blocs tels que les blocs de contrôle.
- L'élève utilise le vocabulaire lié au codage (instructions conditionnelles, variables) et aux probabilités (impossible, peu probable, équiprobable, très probable, certain).

Mise en application

- L'élève crée un code efficace pour comptabiliser les lancers aléatoires d'un dé.
- L'élève crée un code efficace pour comptabiliser les nombres pairs tirés aléatoirement.

Matériel requis

- logiciel de programmation par blocs
- dés numérotés de 1 à 6
- annexe A (tableau dans lequel compiler les résultats expérimentaux)
- crayon
- [annexe \[Dés\]](#)

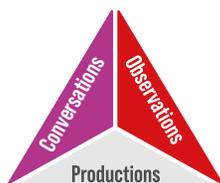
Note : Les exemples dans cette situation d'apprentissage ont été conçus avec Scratch.

Vocabulaire mathématique

probabilité expérimentale, probabilité théorique, événements répétitifs, événements imbriqués, instruction conditionnelle, structure de contrôle

CONTENU

L'évaluation
peut se faire par les...



Mise en situation (10 minutes)

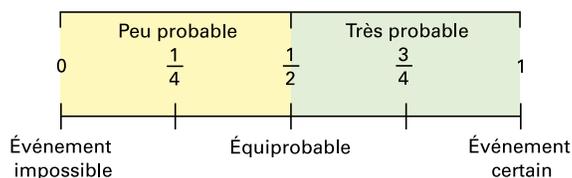
En groupe-classe, montrer l'image des dés ([annexe](#)). Demander aux élèves de trouver des questions à se poser relativement à cette image.



Exemples de questions :

- Combien de points y a-t-il en tout?
- Combien de fois voit-on le nombre 6?
- Quelle est la probabilité d'obtenir un 3?

Grouper les élèves en petits groupes et remettre un dé numéroté de 1 à 6 à chacun. Leur demander de trouver la probabilité théorique d'obtenir un nombre pair. Au besoin, revoir la ligne de probabilité et la terminologie associée.



Exemple de réflexion recherchée :

- Il y a trois nombres pairs, soit 2, 4 et 6. Il y a six résultats possibles lorsqu'on lance un dé. La probabilité d'avoir un nombre pair est donc de $\frac{3}{6}$.
- La fraction $\frac{3}{6}$ peut être simplifiée à $\frac{1}{2}$, qui représente aussi la moitié. Sur la ligne de probabilité, cela nous indique qu'il est équiprobable que l'on obtienne un nombre pair ou un nombre impair.

OBSERVATIONS POSSIBLES	PISTES D'INTERVENTION
L'élève ne sait pas par où commencer.	Que dois-tu faire en premier? Quels sont tous les résultats possibles à la suite d'un lancer de dé? Quels sont les nombres pairs sur le dé? Qu'est-ce qu'une probabilité?
L'élève a de la difficulté à exprimer une probabilité sous forme de fraction.	Que représente le numérateur en probabilité? Que représente le dénominateur en probabilité? Combien de côtés le dé a-t-il au total? Ce total représente-t-il le numérateur ou le dénominateur?
L'élève a de la difficulté à exprimer la probabilité d'obtenir un nombre pair sous forme de fraction.	Que représente la fraction $\frac{3}{3}$? (un entier, $\frac{1}{1}$ ou 100 %) Est-ce vrai que chaque lancer du dé aboutira à un nombre pair?

Demander aux élèves de trouver la probabilité théorique d'obtenir un multiple de 3, puis la probabilité d'obtenir le nombre 4. Les élèves devront ensuite placer les probabilités sur la ligne de probabilité, comme dans le premier exemple.

Exemples de réflexion recherchée :

- Il y a deux multiples de 3 sur un dé à six faces (3 et 6), et six résultats possibles, donc la probabilité d'obtenir un multiple de 3 est de $\frac{2}{6}$.
- La fraction $\frac{2}{6}$ est équivalente à $\frac{1}{3}$.

ou

- Je sais que $\frac{1}{3}$ se situe entre $\frac{1}{4}$ et $\frac{1}{2}$.
- Il y a un seul nombre 4 sur un dé à six faces, et six résultats possibles, donc la probabilité d'obtenir un 4 est de $\frac{1}{6}$.

Demander aux élèves de lancer le dé 20 fois et de compiler les résultats dans un tableau de fréquences relatives comme celui-ci.

ÉVÉNEMENT	DÉNOMBREMENT	FRÉQUENCE	FRÉQUENCE RELATIVE EXPRIMÉE EN FRACTION
J'obtiens un nombre pair.			
J'obtiens un multiple de 3.			
J'obtiens un 4.			

Demander aux élèves ce que l'on remarque dans les résultats obtenus.

Les résultats représentent-ils la probabilité théorique? Pourquoi? (Écrire les réponses des élèves à cette question pour s'y référer plus tard.)

Discuter des résultats obtenus et les comparer avec les probabilités trouvées plus tôt. Encourager les élèves à se questionner sur l'effet du nombre de lancers sur le résultat.

Déroulement (100 minutes)



Demander aux élèves de déterminer la probabilité expérimentale d'obtenir un nombre pair en lançant le dé 1 000 fois.

Exemple de réflexion recherchée :

- 1 000 lancers de dé, c'est beaucoup, et ça prendra beaucoup de temps.
- Un lancer de dé dure environ 5 secondes, donc 1 000 lancers équivalent approximativement à 5 000 secondes (5 000 secondes, c'est environ 83 minutes!).
- Avec autant de lancers, on pourrait faire plus d'erreurs, par exemple oublier de mettre un résultat dans le tableau de fréquences ou perdre le compte du nombre de lancers.

Questionner les élèves à savoir s'il y a une façon plus efficace de déterminer les résultats de 1 000 lancers (on cherche à nommer le code comme outil de représentation).

Demander d'abord aux élèves ce que devrait faire l'ordinateur pour compter le nombre de fois où l'on obtient un nombre pair après 1 000 lancers d'un dé à six faces. Cela deviendra le pseudocode qui dirigera le codage par blocs.

Exemple de pseudocode :

Répéter 1 000 fois.
Générer un nombre aléatoire de 1 à 6.
Ajouter le nombre à la liste « La liste des lancers »
Si le nombre = 2 ou le nombre = 4 ou le nombre = 6,
ajouter 1 au compteur « Nombres pairs ».

Le code est constitué de trois parties :

- Une séquence de code pour lancer un dé à six faces et compiler les résultats.
- Une séquence de code avec un événement conditionnel visant de déterminer si le résultat du lancer est pair ou non.
- Une séquence de code qui compilera le total des lancers aboutissant à un nombre pair.

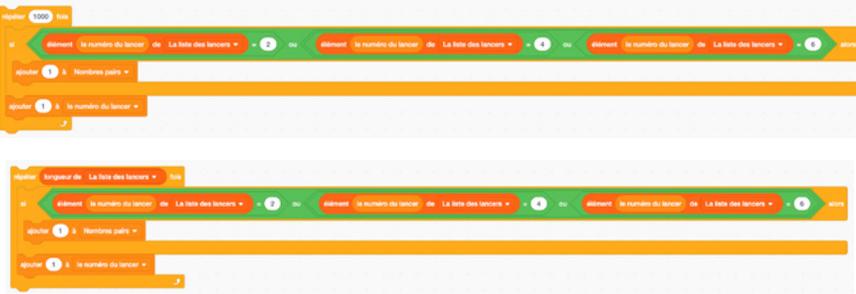
Voici un exemple de code qui détermine le nombre de fois où l'on obtient un nombre pair à la suite des 1 000 lancers du dé, basé sur le pseudocode ci-dessus.

```
quand est cliqué
mettre Nombres pairs à 0
mettre le numéro du lancer à 1
supprimer tous les éléments de la liste La liste des lancers
répéter 1000 fois
mettre face du dé à nombre aléatoire entre 1 et 6
ajouter face du dé à La liste des lancers
répéter longueur de La liste des lancers fois
si élément le numéro du lancer de La liste des lancers = 2 ou élément le numéro du lancer de La liste des lancers = 4 ou élément le numéro du lancer de La liste des lancers = 6 alors
ajouter 1 à Nombres pairs
ajouter 1 à le numéro du lancer
```

Voici le résultat généré par ce code. Le code des élèves pourrait être différent tout en demeurant fonctionnel.



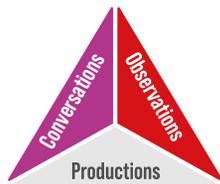
OBSERVATIONS POSSIBLES	PISTES D'INTERVENTION
<p>L'élève ne réussit pas à compiler les résultats du lancer du dé.</p>	<p>Demander à l'élève de trouver le nombre de faces du dé qui lui a été remis.</p> <ul style="list-style-type: none"> • Quel bloc te permet de représenter le dé avec ses six faces? • Combien de lancers dois-tu faire? • Quel bloc te permet de faire plusieurs lancers? 
<p>L'élève oublie de remettre ses variables et ses listes à zéro.</p>	 <p>Demander à l'élève de relire son code à voix haute pour voir s'il manque des consignes.</p> <p>Demander à l'élève d'expliquer les trois blocs identifiés.</p> <p>Demander à l'élève ce qui se produit lorsque les variables ne sont pas remises à zéro.</p> <p>Quel est l'impact sur les résultats des essais?</p>
<p>L'élève n'utilise pas correctement les blocs opérateurs.</p>	 <p>Demander à l'élève s'il y a une façon de faire faire davantage ce qui est voulu à l'ordinateur (utiliser le bloc « ou »). Lui faire comprendre que l'ordinateur lit une donnée à la fois, donc la virgule n'est pas utile en programmation. L'utilisation du bloc opérateur « ou » est utile pour suggérer plusieurs réponses possibles, comme 2 ou 4 ou 6 en ce qui concerne les nombres pairs.</p>

OBSERVATIONS POSSIBLES	PISTES D'INTERVENTION
<p>L'élève répète 1 000 fois dans son code de nombre pair plutôt que de répéter la longueur de la liste.</p>	<p>Demander à l'élève où sont compilés les résultats des lancers de dé (les résultats des lancers de dé devraient être compilés dans « La liste des lancers »).</p> <p>Dans quelle liste dois-tu trouver les nombres pairs?</p> <p>Quelle variable dois-tu ajouter pour rechercher les nombres pairs? (Je dois ajouter la « longueur de la liste des lancers », de sorte que toute la liste des lancers va être répétée.)</p> 
<p>L'élève n'utilise pas les variables adéquatement.</p>	<p>Demander à l'élève ce que signifient les variables. La signification des variables utilisées doit être connue afin de bien comprendre le code.</p> 

Note : Dans les exemples fournis, les notes en jaune sont des explications du fonctionnement du code et devraient être omises lorsque le code est présenté aux élèves.

Objectivation (15 minutes)

L'évaluation
peut se faire par les...



Faire un retour sur l'activité.

- Quels sont les constats?
- Les résultats de la probabilité expérimentale à 20 lancers diffèrent-ils des résultats de la probabilité expérimentale à 1 000 lancers?
- Que remarquez-vous entre la probabilité théorique et la probabilité expérimentale?
- Dans votre code, quels blocs représentent des instructions conditionnelles?
- Pourriez-vous utiliser votre machine à lancers aléatoires ou une machine semblable dans d'autres situations de probabilité? Lesquelles? Pourquoi?

Consolidation

- Proposer aux élèves de modifier leur code pour vérifier la probabilité expérimentale d'obtenir un multiple de 3 ou le nombre 4.
- Proposer aux élèves de faire un code pour comparer la probabilité théorique et la probabilité expérimentale de tirer une bille rouge dans un sac de 10 billes en sachant que 3 billes sont rouges (on aborde les instructions conditionnelles *si/alors/sinon* dans cet exemple).

ÉLÉMENTS À CONSIDÉRER

Liens avec les autres domaines mathématiques

Nombres

B1.4 Comparer et ordonner des fractions à partir des demis jusqu'aux douzièmes, y compris des fractions impropres et des nombres fractionnaires, dans divers contextes.

B1.5 Lire, représenter, comparer et ordonner des nombres décimaux jusqu'aux centièmes, dans divers contextes.

Les instructions conditionnelles [*si/alors/sinon*] permettent de faire des comparaisons entre des nombres en les utilisant en combinaison avec les blocs « opérateurs » [$>$, $<$, $=$].

Données

D2.1 Utiliser des fractions pour exprimer la probabilité que des événements se produisent, la représenter sur une ligne de probabilité et s'appuyer sur cette probabilité pour faire des prédictions et prendre des décisions éclairées.

D2.2 Déterminer et comparer les probabilités théoriques et expérimentales qu'un événement se produise.

Les instructions conditionnelles permettent de coder dans des situations expérimentales avec les probabilités. La terminologie *si/alors/sinon* est utile pour déterminer certains événements d'une expérience aléatoire.

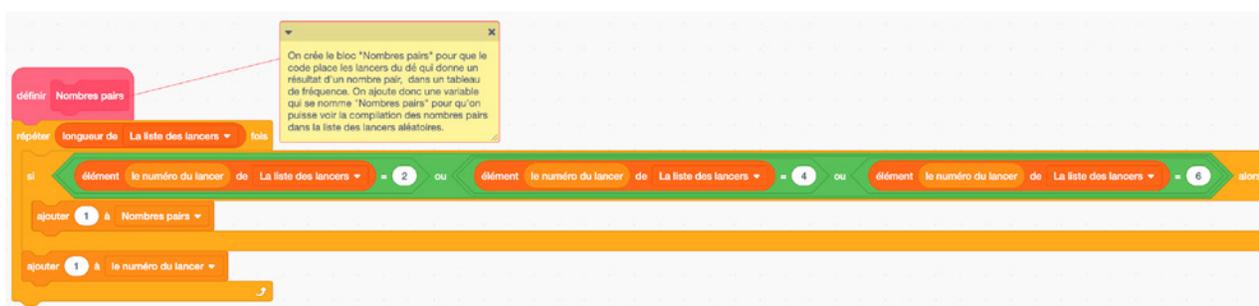
Littératie financière

F1.2 Estimer et calculer le coût de transactions comprenant plusieurs articles dont le prix est exprimé en dollars et en cents, en incluant les taxes de vente, à l'aide de diverses stratégies.

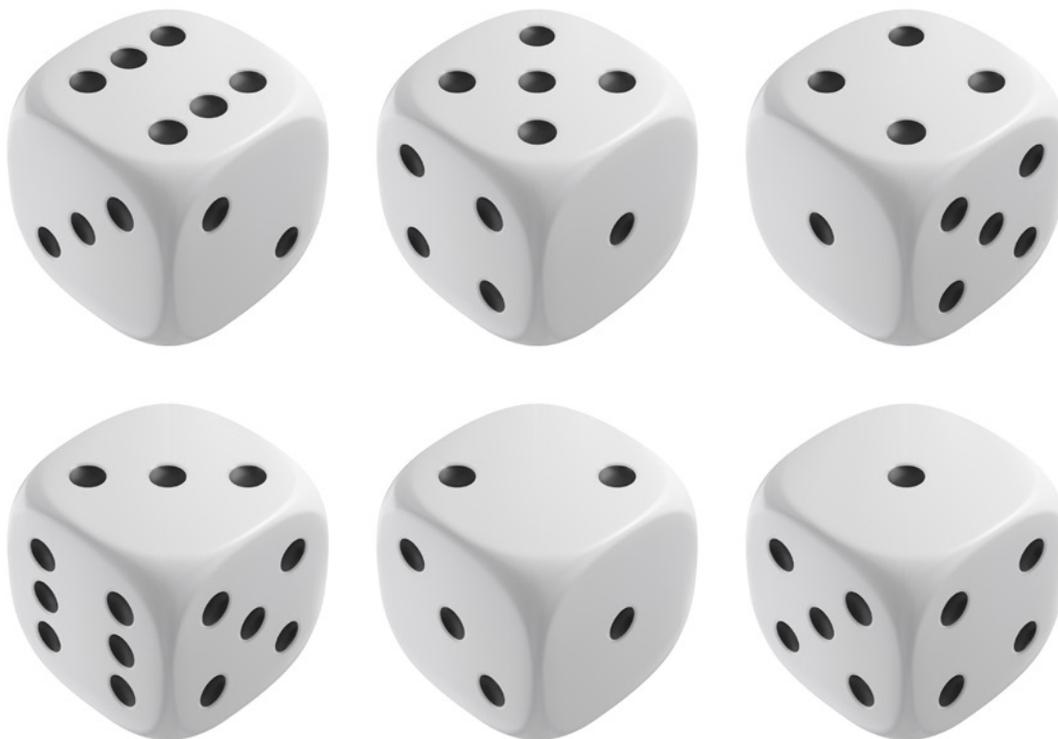
Avec l'élimination de la pièce de 1 cent, le coût total est arrondi au 5 cents près. En utilisant les instructions conditionnelles (*si/alors/sinon*), il est possible de créer des règles à suivre pour arrondir (par exemple, Si le nombre à la position des centièmes est 1 ou 2, ALORS changer la valeur à 0. Si le nombre à la position des centièmes est 3 ou 4, ALORS changer la valeur à 5, etc.).

Différenciation pédagogique et conception universelle de l'apprentissage

- Donner aux élèves les blocs débranchés afin de pouvoir les replacer et en faire un code.
- Donner aux élèves les variables avant de commencer le code.
- Proposer un début de code pour que l'élève le termine.
- Créer certains blocs pour l'élève, comme le bloc « multiple de 3 » et le bloc « nombre pair ».



Annexe – Dés



Situation d'apprentissage – 6^e année

Titre : Petit poisson, petit poisson

Durée : 120 minutes

Sommaire

Les élèves devront coder un jeu vidéo qui consiste à donner des directives à un *sprite* afin d'effectuer des translations sur un plan cartésien dans le but « d'attraper un autre *sprite* ».

Attentes et contenus d'apprentissage ciblés

Nombres

B1 Démontrer sa compréhension des nombres et établir des liens avec leur utilisation dans la vie quotidienne.

B1.1 Lire et représenter des nombres entiers à l'aide d'outils et de stratégies, y compris des droites numériques horizontales et verticales.

Algèbre

C3 Résoudre des problèmes et créer des représentations de situations mathématiques de façons computationnelles à l'aide de concepts et d'habiletés en codage.

C3.1 Résoudre des problèmes et créer des représentations de situations mathématiques de façons computationnelles en écrivant et exécutant des codes efficaces, y compris des codes comprenant des instructions conditionnelles et d'autres structures de contrôle.

C3.2 Lire et modifier des codes donnés, y compris des codes comprenant des instructions conditionnelles et d'autres structures de contrôle, et décrire l'incidence de ces changements sur les résultats et l'efficacité.

Sens de l'espace

E1 Décrire et représenter la forme, la position et le déplacement en se servant de propriétés géométriques et de relations spatiales pour s'orienter dans le monde qui l'entoure.

Pratiques pédagogiques à fort impact en mathématiques à privilégier

Tâches et expériences de résolution de problèmes

En utilisant un contexte qui est authentique aux élèves, tel que la création d'un jeu vidéo, les élèves font preuve de motivation et d'engagement dans l'activité. Les résolutions de problèmes demandent souvent beaucoup d'essais et d'erreurs, tout en faisant des liens avec l'expérience personnelle de l'élève. On peut par la suite inviter les élèves à comparer les tâches et à déterminer ce qui est semblable et différent entre elles.

Conversations mathématiques

En ayant des conversations mathématiques au sujet d'un concept qui est à la fois bien connu (les jeux vidéo) et nouveau (le codage) pour les élèves, le personnel enseignant est en mesure de voir ce que l'élève comprend réellement. Les conversations permettent d'approfondir la réflexion de l'élève et lui demandent d'aller au-delà de sa connaissance et de sa compréhension de base. Des liens et des transferts vers d'autres domaines d'étude sont remarqués au cours des conversations mathématiques.

Connaissances et habiletés préalables

- Les connaissances de base d'un logiciel de programmation par blocs;
- La façon de bien utiliser les coordonnées;
- Le mouvement du *sprite* en utilisant des mouvements positifs (vers la droite ou vers le haut) et négatif (vers la gauche ou vers le bas);
- La création ou la modification de l'apparence d'un *sprite*;
- Les blocs de conditions;
- Le changement de taille des *sprites*;
- Le changement d'arrière-plan;
- Le déplacement continu d'un objet en utilisant la boucle « répéter infiniment »;
- Les coordonnées d'un plan cartésien (quatre quadrants);
- La droite numérique;
- Les nombres entiers négatifs et leur effet sur une opération ou un déplacement.

Résultats d'apprentissage

À la fin de cette situation d'apprentissage, l'élève pourra :

- créer et organiser un code en utilisant ses connaissances des conventions et de l'efficacité de celui-ci dans le but d'en faire un jeu vidéo;
- lire et représenter des nombres entiers en utilisant la droite numérique;
- situer et lire des coordonnées dans les quatre quadrants d'un plan cartésien afin de bien positionner les personnages de son jeu.

Critères d'évaluation selon les grilles d'évaluation du rendement

Connaissance et compréhension

- L'élève connaît les blocs d'un logiciel de programmation par blocs et comprend leur fonctionnement.
- L'élève comprend l'effet du signe négatif sur l'emplacement d'un nombre sur la droite numérique.
- L'élève utilise les coordonnées sur un plan cartésien afin de déterminer l'emplacement d'un *sprite* à l'écran.

Habiletés de la pensée

- L'élève trouve des solutions aux problèmes du code (débugage).
- L'élève analyse ou planifie un projet de codage en utilisant du pseudocode.
- L'élève organise son code en respectant les conventions et en tenant compte de l'efficacité (utiliser un minimum de blocs afin d'atteindre le but).

Communication

- L'élève respecte la syntaxe du logiciel de codage choisi (met les blocs dans le bon ordre et utilise des structures de contrôle qui créent un code efficace).

Mise en application

- L'élève utilise les propriétés des entiers négatifs afin de programmer le mouvement d'un *sprite*.
- L'élève utilise ses connaissances en codage afin de créer un jeu qui comprend l'utilisation d'un plan cartésien et de nombres entiers.

Matériel requis

- ordinateur avec accès au logiciel choisi
- [annexe Plan cartésien](#)

Note : Les exemples dans cette situation d'apprentissage ont été conçus avec Scratch.

Vocabulaire mathématique

codage, code, événement séquentiel, événement simultané, événement répétitif, boucle, condition (*si, alors, sinon*), événement imbriqué, plan cartésien, nombre entier, coordonnée, efficacité (dans un contexte de codage).

CONTENU

Mise en situation (30 minutes)

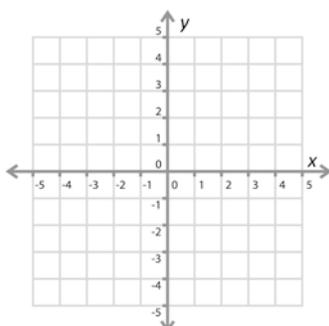


Activité débranchée préparatrice au codage

Demander aux élèves de se tenir debout et de s'imaginer au point $(0, 0)$ d'un plan cartésien. Si c'est la première fois que les élèves travaillent avec les quatre quadrants d'un plan cartésien, il serait profitable de commencer par une minileçon qui montre aux élèves que les axes du plan cartésien sont simplement deux droites numériques qui se croisent à 0, donc le point $(0, 0)$.



Si les coordonnées du plan cartésien n'ont pas encore été enseignées, c'est un bon moment de les présenter, car les élèves pourront profiter de ce nouvel apprentissage immédiatement. Comme appui, afficher, au besoin, un [plan cartésien](#) avec des nombres sur les axes des x et des y , qui servira d'aide-mémoire tout le long de l'activité.



Demander aux élèves de déterminer, dans le plan, l'endroit où se trouve le point $(2, 3)$ par rapport à leur position actuelle $(0, 0)$. Leur demander les directives qui permettraient de se rendre au point $(2, 3)$.

Exemples de réponses :

- On peut faire deux pas à droite et trois pas vers le haut.
- On peut faire trois pas vers le haut et deux pas à droite.
- On peut indiquer le point et glisser en diagonale de $(0, 0)$ à $(2, 3)$.

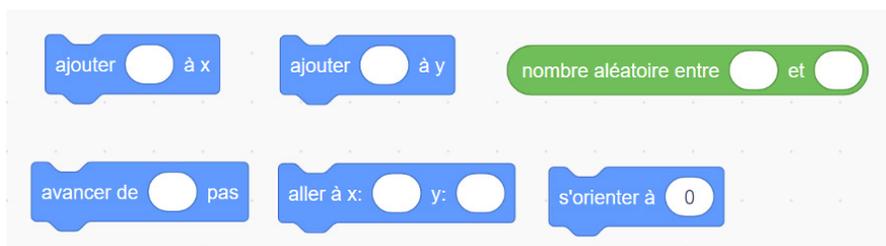
Répéter l'exercice, mais, cette fois, avec le point $(-2, -3)$ par rapport à leur position initiale $(0, 0)$.

Exemples de réponses :

- On peut faire deux pas à gauche et trois pas vers le bas.
- On peut faire trois pas vers le bas et deux pas à gauche.
- On peut indiquer le point et glisser en diagonale de $(0, 0)$ à $(-2, -3)$

Faire remarquer aux élèves que, dans la vie de tous les jours, on peut utiliser des mots tels *à droite* et *vers le haut*, mais qu'il est aussi possible de décrire des déplacements en utilisant des nombres positifs et négatifs.

Présenter aux élèves des blocs de code semblables à ceux-ci.



Note : Pour un groupe d'élèves qui est très à l'aise avec le codage et le logiciel en question, les blocs pourraient être présentés en même temps. Si c'est une première exploration du codage ou du logiciel, la séquence ci-dessous est suggérée.

- Avancer de () pas ET s'orienter à ().
- Ajouter () à x ET ajouter () à y ET aller à x: () y: ()
- Nombre aléatoire entre () et ().

Demander aux élèves de partager leurs observations au sujet des blocs. Les réponses peuvent varier, mais la remarque importante à soulever est que la terminologie utilisée dans le codage est additive (avancer, ajouter, etc.). Demander aux élèves la façon dont on peut déplacer un *sprite* sur tout le plan cartésien si l'on peut uniquement « avancer » ou « ajouter ».

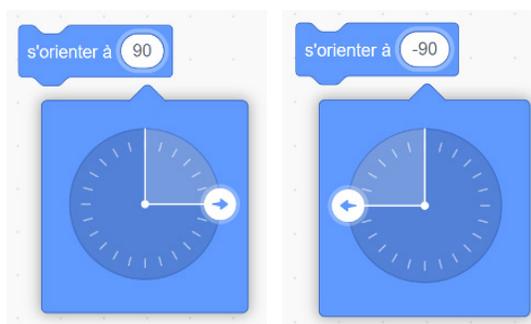
Réponses possibles

- Il faut toujours commencer le *sprite* dans le coin inférieur gauche afin d' « avancer ».

Cette réponse est intéressante, car l'élève réalise que les déplacements vers le haut et vers la droite nécessitent des additions. On peut utiliser les connaissances antérieures des élèves au sujet des opérations inverses afin de leur faire découvrir que les déplacements à gauche et vers le bas nécessitent des soustractions.

- Il faut toujours utiliser un bloc « s'orienter vers » afin de connaître la direction du déplacement.

Cette option est fonctionnelle, mais est beaucoup moins efficace et risque de créer des erreurs dans le code. Cela étant dit, l'élève qui a de la difficulté avec les entiers négatifs pourrait utiliser cette stratégie afin de mieux comprendre le fonctionnement de la droite numérique. Le fonctionnement du bloc « s'orienter vers », par sa nature même, initie les élèves à l'effet du signe négatif en utilisant 90 degrés pour orienter vers la droite et -90 degrés pour orienter vers la gauche.



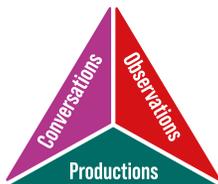
Note : Il faut cliquer sur 90 afin d'entrer une direction sur la roue.

- Il faut utiliser des déplacements positifs pour « en haut » et « à droite », et des déplacements négatifs pour « en bas » et « à gauche ».

Cette réponse témoigne d'une excellente compréhension de l'effet du signe négatif et correspond à la façon la plus efficace de programmer un déplacement dans un plan cartésien. Dans le monde du codage, il faut utiliser des nombres négatifs pour représenter les déplacements vers le bas et vers la gauche.

Déroulement (60 minutes)

L'évaluation
peut se faire par les...

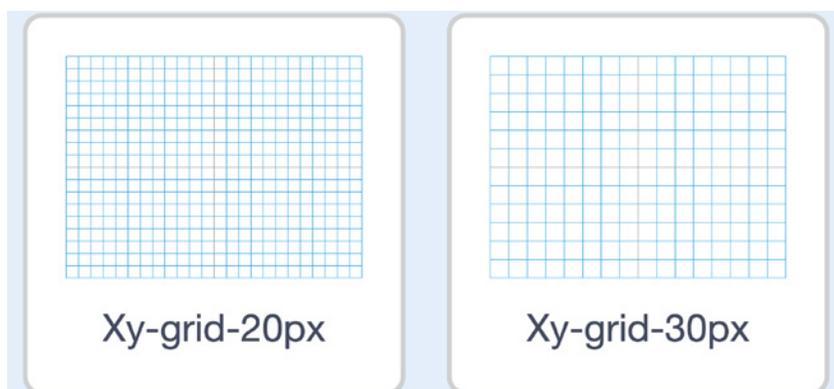


Exploration guidée – Les positions sur le plan cartésien

Afin que la tâche puisse exploiter les apprentissages ciblés, le logiciel de codage choisi doit avoir certaines fonctionnalités. Le logiciel doit :

- permettre un changement d'arrière-plan;
- fonctionner avec des coordonnées dans les quatre quadrants du plan cartésien;
- permettre à l'élève d'effectuer des translations en isolant la composante horizontale (x) et verticale (y).

Lorsque les élèves sont devant leur ordinateur avec un nouveau projet vierge à l'écran, leur demander de choisir un arrière-plan qui représente un plan cartésien. Il est possible que certains plans cartésiens utilisent différentes échelles.

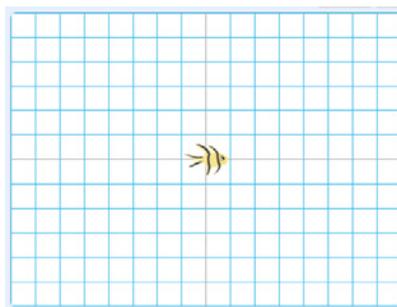


Voici deux arrière-plans qui ont deux échelles différentes, soit des graduations tous les 20 pixels ou tous les 30 pixels. On voit aussi que l'axe des x et l'axe des y sont mis en évidence avec une couleur différente.

Une fois l'arrière-plan choisi, poser aux élèves des questions liées à l'échelle afin d'assurer leur compréhension.

- Quelle est l'unité de mesure type des éléments graphiques sur un écran? (des pixels)
- Combien de « pas » (pixels) seront nécessaires afin que le *sprite* se déplace d'une case? [La réponse dépend de l'échelle. Dans l'exemple ci-dessus, la réponse pourrait être 20 pas (pixels) ou 30 pas (pixels). Cette réponse peut être découverte par expérimentation.]

Demander aux élèves de choisir un *sprite* et de le placer aux coordonnées (0, 0) du plan cartésien. Afin qu'il soit aux coordonnées (0, 0) en début de jeu, le bloc « aller à » pourrait être ajouté au début du code (après le drapeau vert).



Un *sprite* est placé aux coordonnées (0, 0) sur un plan cartésien dans lequel les graduations se trouvent à 30 pixels les unes des autres.

Fournir à l'élève une liste de coordonnées auxquelles leur *sprite* devra se rendre en utilisant uniquement les blocs « ajouter à x » et « ajouter à y ». Les coordonnées devraient se trouver dans les quatre quadrants du plan; par exemple, en partant de (0, 0), le *sprite* devra se rendre de façon successive aux coordonnées suivantes :

[-3, 3]

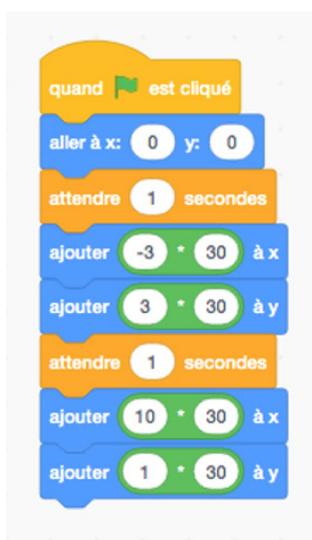
[7, 4]

[-7, -4]

[2, -1]

[0, -3]

La tâche de l'élève est de déterminer la translation nécessaire pour que le *sprite* se rende aux nouvelles coordonnées. La nature de la tâche fait en sorte que la position du *sprite* change, donc le point de départ du *sprite* change perpétuellement. Voici un exemple de code qui respecte les deux premières translations de la liste.



On voit, dans cette séquence de code, deux translations. Les coordonnées du premier point de départ sont (0, 0). La première translation ressemble aux coordonnées de la position finale, soit [-3, 3].

Pour la deuxième translation, il faut prendre en considération le nouveau point de départ [-3, 3]. Donc, le déplacement en x sera de 10 et le déplacement en y sera de 1.

Ce dernier déplacement se compare aux nombres entiers situés sur une droite numérique; par exemple, pour passer de -3 à +7, un déplacement de 10 unités vers la droite est nécessaire (droite numérique horizontale). Le même principe s'applique pour un déplacement en y , qui sera vers le haut (droite numérique verticale). Encourager l'élève à utiliser une droite numérique afin de pouvoir mieux définir le déplacement voulu.

À noter que les blocs d'opération qui multiplient chaque nombre par 30 sont essentiels afin de respecter l'échelle du plan cartésien qui, dans cet exemple, comprend des graduations de 30 pixels.

Aider les élèves à coder les déplacements du *sprite* selon diverses coordonnées d'un plan cartésien.

À la suite de l'exploration guidée, demander aux élèves s'il est efficace (et réaliste) de coder toutes les positions possibles du *sprite* sur l'écran dans le contexte d'un jeu. La réponse est non. Leur demander les modifications qui pourraient rendre le code plus efficace ou plus fonctionnel.

Réponses possibles

- On pourrait utiliser le bloc « aller à [x] [y] », qui fera la translation pour nous sans qu'on ait à le calculer.
- On pourrait utiliser le bloc « aller à position aléatoire », ce qui élimine la prédictibilité de l'emplacement du *sprite*.
- L'échelle de 30 pixels est utile pour l'arrière-plan du plan cartésien, mais avec un arrière-plan différent, on peut utiliser les coordonnées actuelles au lieu de toujours multiplier par 30, ce qui permettrait des positions plus précises.



Ce bloc permettrait à un *sprite* de faire des déplacements selon des coordonnées aléatoires sur un plan quadrillé ayant une échelle de 30 pixels.

Dans l'exemple de jeu ci-dessous, certaines réponses pourront être trouvées dans le code.

La programmation d'un jeu interactif

Donner aux élèves le défi de créer un jeu qui sera ensuite montré à un pair. Le but du jeu est de tester les connaissances du pair sur la description de translations dans les quatre quadrants du plan cartésien. Le jeu consistera à déplacer un *sprite* à l'aide de vecteurs de translation (coordonnées) afin qu'il touche des objets divers.

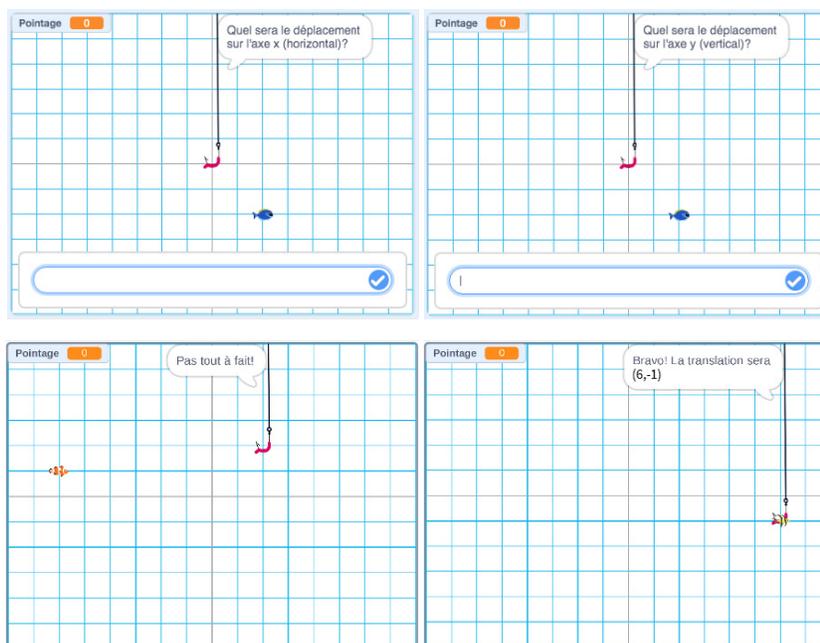
Il existe plusieurs possibilités de jeux qui pourraient vérifier la compréhension de ces concepts. Voici un exemple de jeu de pêche que pourrait produire l'élève :

CODE DU <i>SPRITE</i> « HAMEÇON »	CODE DU <i>SPRITE</i> « POISSON »

Note : Dans ce code, on voit l'utilisation de variables qui détermineront les déplacements nécessaires afin que l'hameçon aille « attraper » le poisson. On voit aussi l'utilisation d'un « message ». Les messages permettent aux *sprites* d'interagir davantage et peuvent être très utiles dans le contexte du codage d'un jeu. Dans ce cas, lorsque le message « Déplace-toi, petit poisson! » est envoyé, le *sprite* du poisson reçoit le message et exécute le code.

On voit aussi l'utilisation du bloc « regrouper », qui rend le code moins encombrant en évitant d'utiliser quatre différents blocs « dire » pour communiquer avec l'utilisatrice ou l'utilisateur.

Voici ce à quoi l'interface du jeu pourrait ressembler :



Note : Les captures d'écran ont été prises de l'interface du programme en plein écran à différents moments du jeu.

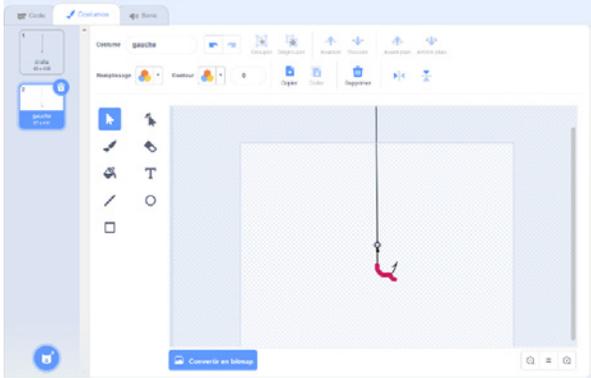
Aller plus loin – Le jeu de pêche

Il existe plusieurs façons de rendre le jeu plus interactif et plus amusant. Encourager les élèves à rendre le jeu aussi interactif que possible. Voici quelques suggestions :

- Ajouter une variable qui pourra compter le pointage et faire répéter le code jusqu'à ce qu'un certain nombre de points soit atteint.
- Utiliser un plan cartésien avec une différente échelle pour créer un niveau plus facile ou plus difficile.
- Ajouter un élément de temps, comme un chronomètre qui ferait un décompte.
- Coder les poissons pour qu'ils demeurent en place pendant un nombre de secondes précises.

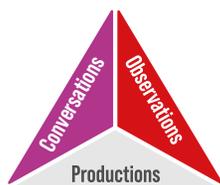
Voici un exemple de code qui incorpore les éléments de pointage et de répétition :

```
quand cliqué
mettre déplacement en X à 0
mettre déplacement en Y à 0
mettre Pointage à 0
aller à x: 0 y: 0
envoyer à tous déplace-toi, poisson!
répéter jusqu'à ce que Pointage = 10
  dire "Quelle translation sera nécessaire pour attraper le poisson?" pendant 4 secondes
  demander "Quel sera le déplacement sur l'axe x (horizontal)?" et attendre
  mettre déplacement en X à réponse
  demander "Quel sera le déplacement sur l'axe y (vertical)?" et attendre
  mettre déplacement en Y à réponse
  ajouter déplacement en X * 30 à x
  ajouter déplacement en Y * 30 à y
  si touche le Poisson ? alors
    dire regrouper "Bravo! La translation sera " et regrouper déplacement en X et " " et déplacement en Y pendant 10 secondes
    ajouter 1 à Pointage
    envoyer à tous déplace-toi, poisson!
  sinon
    dire "Pas tout à fait!" pendant 10 secondes
    envoyer à tous déplace-toi, poisson!
envoyer à tous cache-toi, petit poisson!
dire "Bravo! Tu as réussi 10 translations!"
```

OBSERVATIONS POSSIBLES	PISTES D'INTERVENTION
L'élève n'arrive pas à déplacer son <i>sprite</i> aux coordonnées indiquées.	<p>Faire un rappel à l'élève qu'un pixel est très petit et qu'il faudra employer une échelle pour obtenir la translation voulue.</p> <p>Vérifier que l'élève a bien codé la position de départ comme étant (0, 0), et que le <i>sprite</i> soit replacé à cet endroit pour reprendre le code.</p>
L'élève ne connaît pas la façon de modifier ou de créer un <i>sprite</i> .	<p>Décortiquer avec l'élève le fonctionnement du logiciel et revoir la façon d'ajouter un <i>sprite</i>.</p> <p>Demander à l'élève si un élément sur la page pourrait aider.</p>  <p>L'hameçon utilisé dans l'exemple est dessiné à la main. Les élèves peuvent choisir des <i>sprites</i> existants ou créer ou modifier un <i>sprite</i> dans le menu « costumes ».</p>
L'élève n'arrive pas à bien utiliser les variables.	<p>Questionner l'élève afin de vérifier sa compréhension du bloc « variable ».</p> <ul style="list-style-type: none"> • Pourrais-tu me décrire ce qu'est une variable? • Y a-t-il une ou des valeurs qui changent dans ton code? <p>Il peut aussi être profitable de rendre toutes les variables visibles sur l'interface afin que l'élève puisse tenir compte de la valeur associée à chacune d'elles.</p>
L'élève n'utilise pas le bloc « regrouper » pour créer des phrases complexes ayant des variables.	<p>Le bloc « regrouper » peut être un peu intimidant au début. Un pseudocode peut venir appuyer la compréhension de son utilité. Dans l'exemple du code ci-dessus, le bloc « regrouper » est utilisé afin de communiquer la translation à l'élève. Le pseudocode pourrait donc ressembler à ceci :</p> <p>Dire : « Bravo! La translation sera (variable déplacement en x), (variable déplacement en y). »</p> <p>Il faut d'abord créer un bloc « joindre » ayant quatre éléments (ovales). Exemple :</p> 
L'élève a de la difficulté à utiliser les structures conditionnelles.	<p>Demander à l'élève de verbaliser l'intention de son code ou d'écrire un pseudocode. Trouver les endroits où l'on voit des mots indiquant des conditions (<i>si, lorsque, jusqu'à, quand</i>).</p>

Objectivation (30 minutes)

L'évaluation
peut se faire par les...



Faire une foire avec les élèves pour essayer les jeux. Leur demander de noter les ressemblances et les différences entre les jeux essayés et le leur. Encourager la rétroaction des pairs au cours de l'essai du jeu.

Guider la réflexion par le questionnement.

- Quelles sont les principales différences entre les codes? Y a-t-il plus d'une façon de coder le comportement voulu? (Oui, il existe plusieurs façons d'écrire un code. Il est probable que les codes dans la classe seront similaires, mais pas identiques.)
- Est-ce qu'il serait possible de rendre le jeu plus facile (par exemple, ajouter des nombres aux axes) ou plus difficile (par exemple, enlever le quadrillé et exiger la prédiction de la position)?
- As-tu remarqué des éléments, dans les autres codes, que tu aimerais intégrer au tien (par exemple, des éléments visuels ou textuels plus précis, des séquences plus efficaces, des blocs spécialisés, telles des booléennes ou des boucles conditionnelles)?

Consolidation

Proposer aux élèves de programmer un jeu-questionnaire dans lequel un *sprite* doit se rendre à des coordonnées précises. L'élève aurait d'abord à entrer les translations nécessaires afin de passer d'une coordonnée à l'autre. L'ajout d'un bloc permettant de tracer le trajet pourrait dévoiler une forme ou un message mystère indiquant que l'élève a réussi à déplacer son *sprite* aux bons endroits. Au cours de l'essai du jeu-questionnaire, encourager la rétroaction des pairs.



Le bloc « stylo en position d'écriture » est un exemple de bloc qui permet de tracer le parcours du *sprite*. Ce bloc fait partie du logiciel de codage Scratch. Dans d'autres logiciels, ce type de bloc pourrait porter un nom différent.

ÉLÉMENTS À CONSIDÉRER

Liens avec les autres domaines mathématiques

Nombres

B2.1 Utiliser les propriétés des opérations et les relations entre les opérations pour résoudre des problèmes comportant des nombres naturels, des nombres décimaux, des fractions, des rapports, des taux et des pourcentages, y compris des problèmes à plusieurs étapes ou plusieurs opérations.

Un problème à plusieurs étapes peut rapidement devenir un code à plusieurs lignes. L'efficacité peut donc venir en appui afin de simplifier un problème plus complexe, par exemple, en utilisant des boucles, des variables ou des blocs personnalisés.

Sens de l'espace

F1.4 Situer et lire des coordonnées dans les quatre quadrants d'un plan cartésien, et décrire les déplacements d'une coordonnée à l'autre à l'aide de translations. (puisque nous parlons tout de suite après du plan cartésien.

Le plan cartésien est souvent utilisé pour montrer la position et les déplacements dans des contextes de codage visuel. Il est donc possible de demander à l'élève de faire plusieurs déplacements en utilisant les coordonnées du plan cartésien. L'élève peut par la suite rendre son code plus efficace en organisant les déplacements et en utilisant des boucles et des blocs personnalisés.

Littératie financière

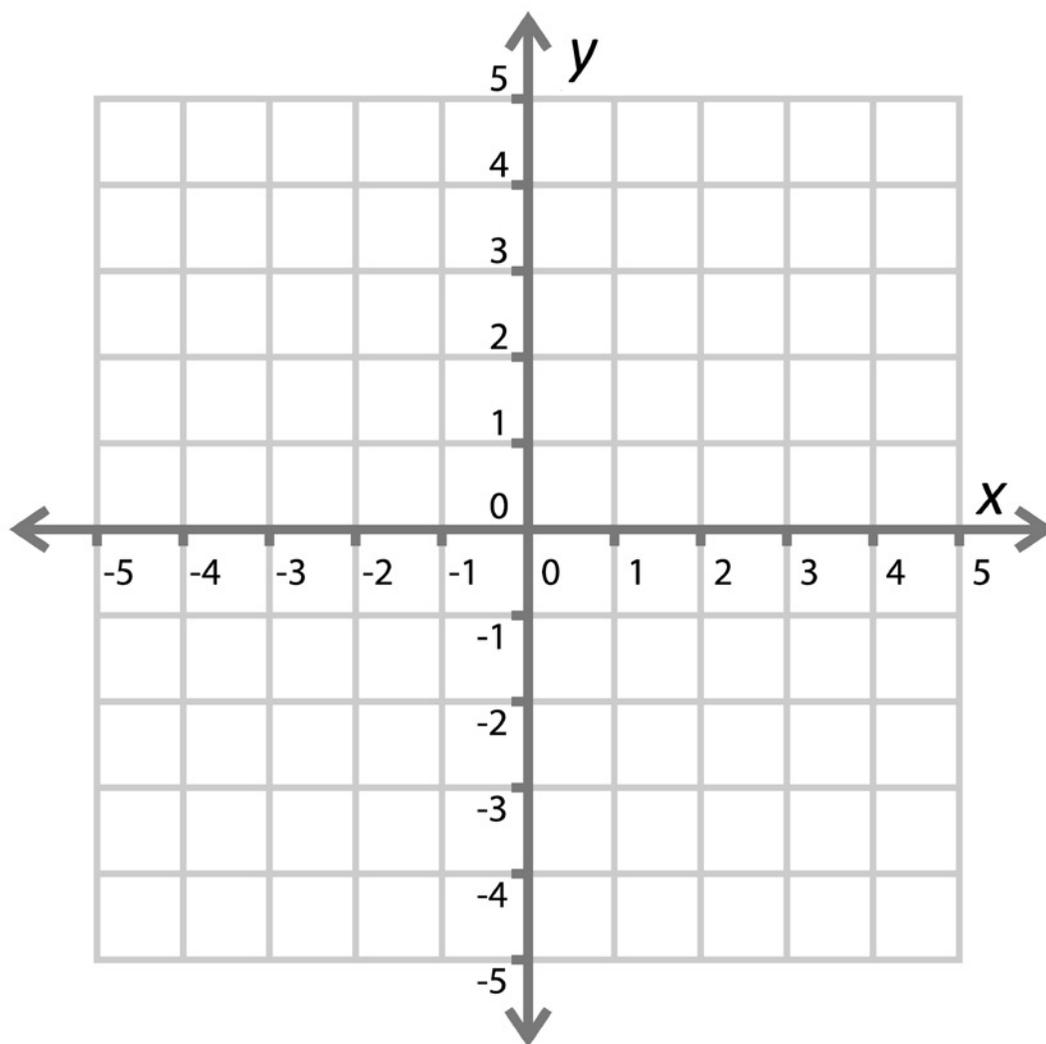
F1.4 Expliquer le concept des taux d'intérêt et déterminer les types de taux d'intérêt et de frais associés à différents comptes et prêts offerts par diverses banques et autres institutions financières.

Les scénarios sur l'intérêt, surtout l'intérêt composé, peuvent facilement devenir plusieurs lignes de code. L'utilisation de variables, de blocs personnalisés et de généralisations permet de simplifier le code (et de rendre une situation financière complexe plus facile à comprendre).

Différenciation pédagogique et conception universelle de l'apprentissage

- Demander à l'élève de créer un pseudocode avant même de commencer afin de créer un plan pour sa programmation.
- Déterminer dès le début les variables avec l'élève.
- Fournir un code incomplet pour que l'élève le modifie.
- Fournir des aide-mémoire avec les fonctions des différents blocs.
- Inviter les élèves à travailler en équipes, soit pour encourager l'apprentissage par les pairs, soit pour favoriser l'enseignement explicite et personnalisé selon leurs besoins.
- Encourager les élèves à créer des niveaux à leur jeu pour que la difficulté augmente graduellement (par exemple, les poissons ne sont pas stationnaires, il y a des objets à éviter).

Annexe – Plan cartésien



Situation d'apprentissage – 7^e année

Titre : Les peintures des souliers

Durée : 180 minutes

Sommaire

Dans cette situation d'apprentissage, l'élève recueille les peintures des souliers des élèves de son groupe-classe afin de suggérer un inventaire de souliers pour préadolescentes et préadolescents au magasin de chaussures de la communauté. À partir des données, l'élève crée des codes à l'aide d'un logiciel de programmation par blocs en vue de trouver la moyenne et la médiane des peintures des souliers des élèves du groupe-classe. L'élève observe qu'en ajoutant ou en retirant des données les résultats peuvent avoir une incidence sur les mesures de tendances centrales.

Attentes et contenus d'apprentissage ciblés

Données

- D1.** Traiter, analyser et utiliser des données pour formuler des arguments persuasifs et prendre des décisions éclairées dans divers contextes de la vie quotidienne.
 - D1.5** Déterminer l'incidence de l'ajout ou de la suppression de données sur les mesures de tendance centrale et décrire comment ces changements modifient la représentation et la distribution des données.

Algèbre

- C3.** Mettre en application ses habiletés en codage pour résoudre des problèmes et créer des représentations de situations mathématiques de façon computationnelle à l'aide de concepts et d'habiletés en codage.
 - C3.1** Résoudre des problèmes et créer des représentations de situations mathématiques de façon computationnelle en écrivant et en exécutant des codes efficaces, y compris des codes comprenant des événements influencés par un dénombrement prédéfini ou un sous-programme et d'autres structures de contrôle.
 - C3.2** Lire et modifier des codes donnés, y compris des codes comprenant des événements influencés par un dénombrement prédéfini ou un sous-programme et d'autres structures de contrôle, et décrire l'incidence de ces changements sur les résultats et l'efficacité.

Pratiques pédagogiques à fort impact en mathématiques à privilégier

Enseignement explicite

Dans la mise en situation, le personnel enseignant questionne les élèves au sujet du mode, de la médiane et de la moyenne afin d'activer leurs connaissances antérieures. À la lumière des réponses des élèves, le vocabulaire à l'étude est introduit tel que *mesures de tendance centrale* et *sous-programme*. Le personnel enseignant montre un sous-programme (pseudocode pour le mode) et interroge les élèves au sujet du pseudocode. Les élèves émettent leurs opinions/réflexions au cours d'un pense-parle-partage et travaillent ensuite en pratique guidée afin de créer des pseudocodes pour la médiane et la moyenne.

Enseignement en petits groupes

Au cours du déroulement, l'enseignement en petits groupes se fait, à raison d'une dizaine de minutes à la fois, avec des groupes prédéterminés afin de converser et de faire progresser les élèves dans la création de leur code. C'est le moment idéal pour parler d'instructions conditionnelles et de l'incidence de ces blocs sur un code.

Regroupements flexibles

La communication et la collaboration sont des atouts dans des tâches comme celle-ci. En travaillant en groupes, les élèves se déplacent et se trouvent un endroit pour travailler entre pairs. Le personnel enseignant circule parmi les élèves et les appuie au besoin. Le personnel enseignant questionne les élèves et pousse leur raisonnement pendant leur création pour collecter des données, calculer la moyenne et trouver la médiane.

Connaissances et habiletés préalables

Pour être en mesure de réaliser cette situation d'apprentissage, l'élève doit :

- comprendre ce qu'est un pseudocode;
- connaître ce que sont les mesures de tendance centrale;
- être capable d'utiliser un logiciel de programmation par blocs.

Résultats d'apprentissage

À la fin de cette situation d'apprentissage, l'élève pourra :

- déterminer les mesures de tendance centrale à l'aide d'un logiciel de programmation par blocs;
- utiliser de façon répétée des blocs d'instructions conditionnelles ainsi que des blocs opérateurs;
- écrire des pseudocodes efficaces;
- reconnaître l'incidence de l'ajout ou du retrait de certaines données sur les mesures de tendance centrale.

Critères d'évaluation selon les grilles d'évaluation du rendement

Connaissance et compréhension

- L'élève comprend la catégorie des blocs « mes blocs » afin de créer des blocs personnalisés* et de les utiliser.
- L'élève utilise des variables dans son code.

Habiletés de la pensée

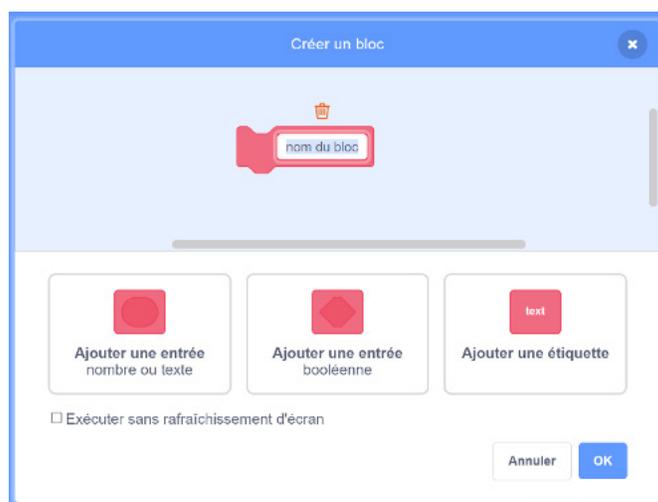
- L'élève écrit des pseudocodes avec précision pour recueillir des données, calculer la moyenne et trouver la médiane des données recueillies.
- L'élève corrige avec exactitude les erreurs de syntaxe et de logique dans ses codes (débogage).
- L'élève explique l'incidence de l'ajout ou du retrait de données dans le calcul de la moyenne et de la médiane.

Communication

- L'élève explique de façon cohérente l'incidence de certains blocs, tels les blocs opérateurs et les blocs de contrôle.
- L'élève explique avec clarté le code de ses pairs.

Mise en application

- L'élève crée un code efficace pour recueillir les pointures des souliers des élèves du groupe-classe.
- L'élève ajoute avec cohérence des blocs d'instructions conditionnelles dans ses codes.



* **Blocs personnalisés** : Cette capture d'écran montre le menu de création d'un nouveau bloc dans le logiciel de codage Scratch. En créant un nouveau bloc, on peut rendre le code plus efficace en résumant une séquence de code qui sera répétée à divers endroits dans le code en un seul bloc. La création de blocs dans d'autres logiciels de codage par blocs pourrait avoir une interface qui diffère de celle-ci.

Matériel requis

- logiciel de programmation par blocs
- papier
- crayon
- [annexe 1 \[Mise en situation\]](#)
- [annexe 2 \[Pseudocode\]](#)

Note : Les exemples dans cette situation d'apprentissage ont été conçus avec Scratch.

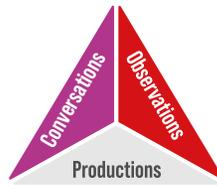
Vocabulaire mathématique

mesures de tendance centrale, mode, médiane, moyenne, sous-programme [codage], pseudocode

CONTENU

Mise en situation (15 minutes)

L'évaluation
peut se faire par les...



Demander aux élèves d'observer, en groupe-classe, l'image ci-dessous ([annexe \[Mise en situation\]](#)) en essayant d'imaginer le problème mathématique qui pourrait y être associé.



Présenter aux élèves l'image ci-dessous. Les informer que les deux images sont liées. Leur demander de nouveau de poser des questions et de tenter de déterminer le problème associé à l'image. Guider le questionnement afin que les élèves se posent des questions au sujet des mesures de tendance centrale (moyenne, médiane et mode).



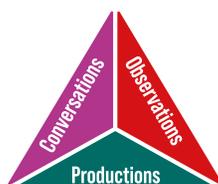
Exemple de questions :

- Pourquoi croyez-vous que certains nombres sont en caractères gras?
- Quelles mathématiques pourraient être liées à des souliers?
- Est-ce que tous les souliers dans l'image sont semblables ou différents? Comment cela affecte-t-il ton interprétation du problème?

OBSERVATIONS POSSIBLES	PISTES D'INTERVENTION
L'élève ne parle pas des mesures de tendance centrale.	Que représente la formule? As-tu déjà utilisé une formule comme celle-ci? Dans quel contexte? Pourquoi penses-tu que le chiffre 2 est écrit en gras?
L'élève n'a aucune idée de ce que peuvent représenter les souliers.	Combien de souliers vois-tu? Selon toi, quelles sont les pointures des souliers? Comment les souliers peuvent-ils différer les uns des autres?
L'élève ne connaît pas les mesures de tendance centrale.	Revoir avec les élèves les concepts de mode, de médiane et de moyenne.

Déroulement (150 minutes)

L'évaluation
peut se faire par les...



Se préparer au codage

Expliquer aux élèves que des données concernant la pointure des souliers des élèves du groupe-classe seront collectées. À l'aide de ces données, les élèves créeront des codes qui aideront à trouver la moyenne et la médiane des pointures des souliers dans un logiciel de programmation par blocs.

Pistes de réflexion

- Quelles données cherche-t-on?
On cherche la pointure des souliers des élèves du groupe-classe.
- Comment peut-on automatiser la collecte de données?
À l'aide d'un code (par blocs ou dans un chiffrier), en consultant un site Web où l'on peut créer des sondages, en remplissant un formulaire, etc.

- Serait-il possible de créer un code pour la collecte de données? Qu'est-ce que le code devra faire?

Le code devra demander la pointure des souliers d'une personne, ajouter cette pointure à une liste, puis poser la question à quelqu'un d'autre jusqu'à ce que toutes les données soient recueillies (un pseudocode pourrait aider à cette étape).

- Est-ce plus facile de coder pour la moyenne, la médiane ou le mode?

Des élèves diront que la médiane ou le mode sont plus faciles à coder, car il s'agit simplement de trouver un nombre dans une liste, tandis que la moyenne nécessite une opération mathématique. En réalité, le code pour trier la liste des données en ordre croissant est beaucoup plus exigeant que le code pour faire des opérations mathématiques. Il serait intéressant et pertinent de reposer cette question au moment de l'objectivation.

- Qu'est-ce que le code devra faire pour calculer ou reconnaître chacune des mesures de tendance centrale?

- Pour la moyenne, le code devra calculer la somme de toutes les données, puis diviser cette somme par le nombre de données dans la liste.
- Pour la médiane, le code devra classer les données en ordre croissant et reconnaître la valeur centrale, soit en trouvant la valeur au centre de la liste (nombre impair de données) ou en calculant la moyenne des deux valeurs au centre de la liste (nombre pair de données).
- Pour le mode, le code devra classer les données en ordre croissant, puis comparer les valeurs afin de déterminer celle que l'on trouve le plus souvent dans la liste.

En groupe-classe, présenter le [pseudocode](#) ci-dessous afin d'aider les élèves à bien comprendre ce qu'est un pseudocode.

Variables : nb1
nb2
somme
Départ
Mettre nb1 = 0.
Mettre nb2 = 0.
Mettre somme = 0.
Définir la valeur de nb1.
Poser la question « Quelle est la valeur de nb1? ».
La réponse devient la valeur de nb1.
Définir la valeur de nb2.
Poser la question « Quelle est la valeur de nb2? ».
La réponse devient la valeur de nb2.
Somme = nb1 + nb2
Fin

- Quelles sont les variables? Que représentent-elles?

Les variables sont :

- nb1 : la valeur de la première donnée recueillie;
- nb2 : la valeur de la deuxième donnée recueillie;
- somme : la somme de toutes les données.

- Quel est le rôle des trois blocs « mettre à 0 » au début du code?

Sans les blocs « mettre à 0 », toutes les variables qui ont été définies lors de la dernière exécution du code demeureront actives. Cela pourrait avoir des effets indésirables sur le comportement du code. C'est un peu comme appuyer sur la touche « CE » d'une calculatrice.

- Quel serait le code, dans un logiciel de programmation par blocs, pour ce pseudocode?

Inviter les élèves à créer le code pour la situation que représente le pseudocode. Les codes peuvent varier. Il existe plusieurs façons de représenter ce pseudocode au moyen du codage par blocs.

Exemple de code :



Note : Dans cette partie de la situation d'apprentissage, les élèves s'approprient les blocs au moyen desquels les questions sont posées et la valeur des variables est modifiée. Si les élèves ont déjà utilisé le logiciel de codage en question, il serait possible de passer directement à la prochaine partie de l'activité, soit la collecte des données. Cette activité peut aussi servir d'évaluation diagnostique pour déterminer si les élèves ont tout ce qu'il leur faut pour passer à la collecte des données.

Prendre le temps de circuler parmi les élèves et d'écouter leurs conversations. À partir de leurs observations et de votre écoute active, les questionner, au besoin, pour les guider ou mieux comprendre leur raisonnement.

Coder pour recueillir des données

Demander aux élèves de créer un pseudocode pour faire la collecte des données de toutes et de tous les élèves du groupe-classe. Les élèves peuvent partir du code préparé à l'étape précédente ou commencer un nouveau code.

Exemple de pseudocode :

Variables : * nombres de données à entrer

* peinture des souliers

* liste de données des peintures de souliers

Départ
Répéter pour chaque donnée.
Demander la peinture des souliers.
Ajouter la réponse à la liste des peintures des souliers.
Demander s'il reste des données à entrer. Si oui – répéter Si non – fin

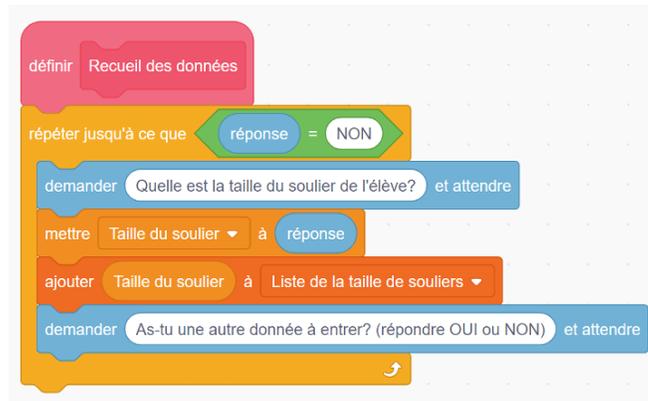
Une fois le pseudocode terminé, inviter les élèves à créer leur code pour recueillir les données liées à la peinture des souliers des élèves, puis à les compiler dans une liste.

Exemple de code :



Note : Plusieurs logiciels de codage par blocs rendent possible la personnalisation des blocs. Si c'est le cas, les élèves peuvent créer un bloc (bloc personnalisé) et le définir avec leur code. De cette façon, le code principal devient moins encombré.

Exemple de bloc personnalisé :



Une fois ce bloc créé, le code principal ressemblerait à ceci :



Coder pour analyser des données

Demander aux élèves ce que devra faire l'ordinateur pour trouver la moyenne des pointures des souliers parmi les données recueillies. Au besoin, l'image de départ avec la formule de la moyenne pourrait être réaffichée afin d'activer les connaissances antérieures des élèves. La formule de la moyenne peut aussi être transformée en pseudocode. Les inviter à rédiger les directives pour l'ordinateur afin de calculer la moyenne des données recueillies.

Voici un exemple de pseudocode :

Variables : * numéro de la liste (ou élément de la liste)

* somme

* liste des pointures de souliers

Départ
Mettre numéro de liste à 1 (on commence par le premier élément de la liste).
Mettre somme à 0.
Répéter pour la longueur de la liste (nombre d'éléments).
Ajouter (élément 1) de la liste à somme.
Ajouter 1 à numéro de la liste.
Dire moyenne = somme/longueur de liste.
Fin

Exemple de code :

```
quand ce sprite est cliqué
mettre Numéro de la liste à 1
mettre SOMME à 0
répéter longueur de Liste de la taille de souliers fois
ajouter élément Numéro de la liste de Liste de la taille de souliers à SOMME
ajouter 1 à Numéro de la liste
dire regrouper La moyenne de la taille de souliers de la classe est et SOMME / longueur de Liste de la taille de souliers
```

Poser aux élèves les questions suivantes :

- Qu'arrivera-t-il si j'ajoute des pointures de souliers dans ma liste?
- La moyenne sera-t-elle la même? Pourquoi?
- Pourrait-on rendre ce code encore plus efficace?

Note : Ce code est structuré ainsi afin de séparer les étapes et d'encourager le questionnement et l'exploration de l'élève. Une version plus simple de ce code verrait ajouter les éléments de la liste à la valeur de la variable « somme » au moment de la collecte des données, éliminant toute cette étape. Ce code ressemblerait à ceci :

```
répéter jusqu'à ce que réponse = NON
demander Quelle est la taille du soulier de l'élève? et attendre
mettre Taille du soulier à réponse
ajouter Taille du soulier à Liste de la taille de souliers
ajouter Taille du soulier à SOMME
demander As-tu une autre donnée à entrer? (répondre OUI ou NON) et attendre
```

L'ajout de ce bloc à notre code de collecte de données permet de compiler la somme au fur et à mesure que les données sont recueillies.

Inviter les élèves à créer un pseudocode et un code afin de trouver la médiane des pointures des souliers des élèves du groupe-classe.

Questions possibles :

- Quelles sont les étapes à suivre pour déterminer la médiane?
- Existe-il un bloc pouvant faire en sorte que les données soient ordonnées en ordre croissant? Peut-on en créer un? (Les élèves vont mentionner qu'il faut trier par ordre croissant les éléments de la liste des pointures de souliers afin de trouver la médiane.)

Exemple de pseudocode :

Variables : * valeur centrale 1

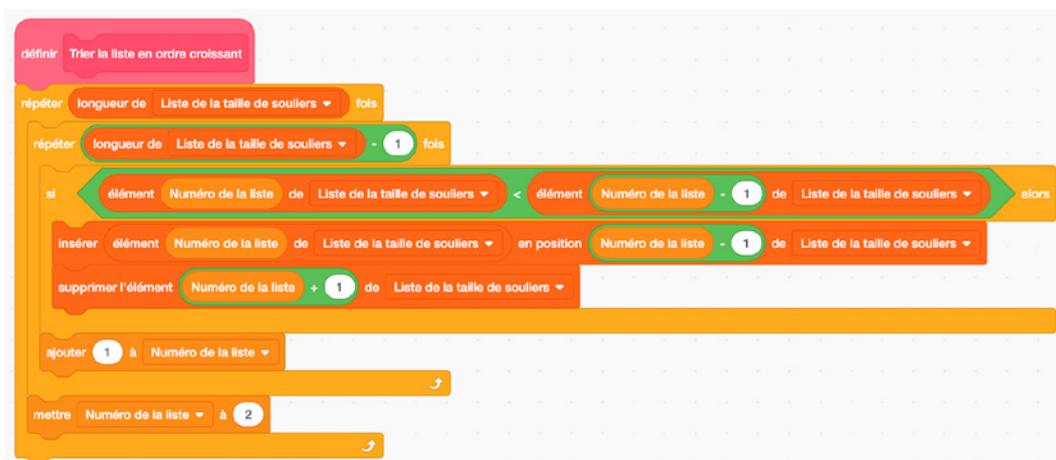
* valeur centrale 2

* liste des pointures de souliers

Départ
Mettre « liste des pointures de souliers » en ordre croissant.
Si la liste a un nombre pair d'éléments
Identifier l'élément correspondant à $(\text{longueur de la liste} / 2)$.
Mettre la valeur de l'élément à [« valeur centrale 1 »].
Identifier l'élément correspondant à $((\text{longueur de la liste} / 2) + 1)$.
Mettre la valeur de l'élément à [« valeur centrale 2 »].
Mettre médiane à $([\text{« valeur centrale 1 »} + \text{« valeur centrale 2 »}] / 2)$.
SINON
Mettre médiane à l'élément correspondant à $((\text{longueur de la liste} + 1) / 2)$.
Fin

Note : Si la liste contient un nombre pair de données, faire remarquer aux élèves que le calcul de la médiane prend la même forme que le calcul de la moyenne. On trouve essentiellement la moyenne des deux valeurs centrales.

Voici un exemple de code pour ordonner les données de la liste en ordre croissant. Encore une fois, le code a été transformé en bloc personnalisé. Cette étape est utile, mais pas nécessaire.

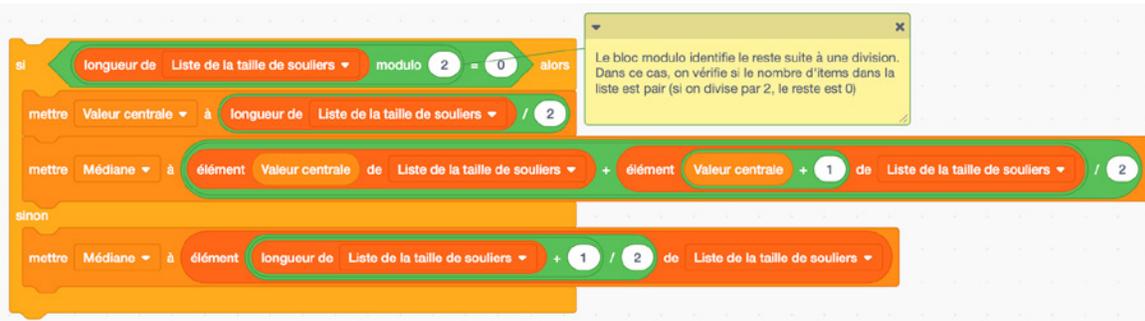


Courte explication de la logique du code :

Ce code fonctionne en comparant une valeur de la liste avec la valeur avoisinante. Si la valeur est plus petite, les positions des éléments de la liste sont inversées. Le code est répété le même nombre de fois qu'il y a d'éléments dans la liste afin de s'assurer que chaque valeur est plus petite que la valeur suivante.

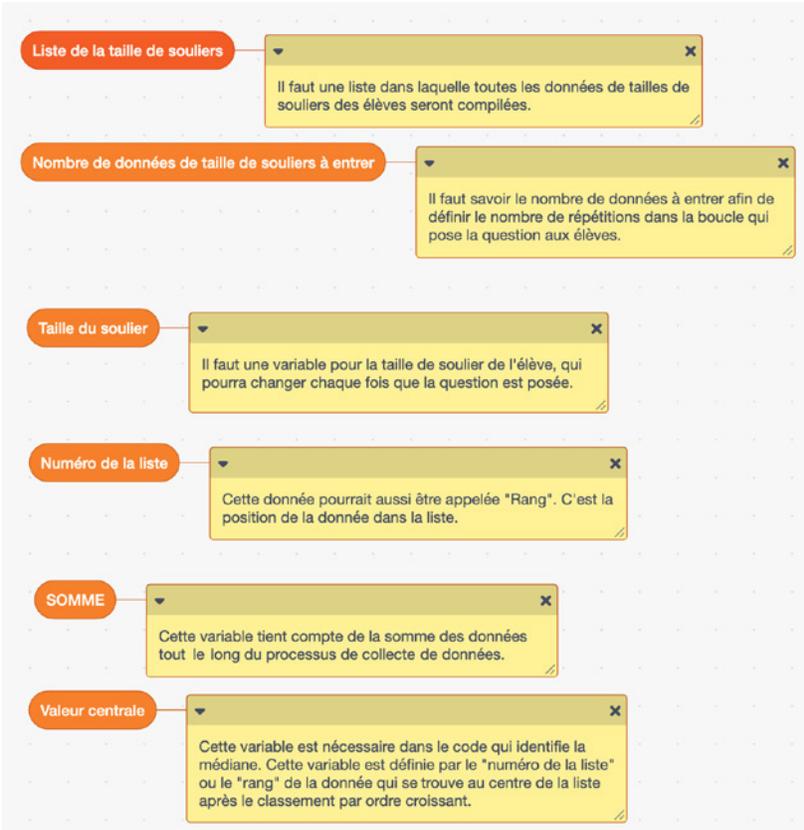
Note : Ordonner des nombres en ordre croissant est un problème complexe pour l'ordinateur, qui doit comparer les valeurs des données, puis les ordonner. Ce genre de problème se prête très bien à une recherche dans Internet. Il existe plusieurs communautés et une panoplie de projets existants qui peuvent répondre aux questions des élèves et des membres du personnel. En modélisant ce genre de débrouillardise, les élèves seront plus à l'aise à faire de même, et en apprendront davantage sur ce qui est possible avec les logiciels de codage par blocs.

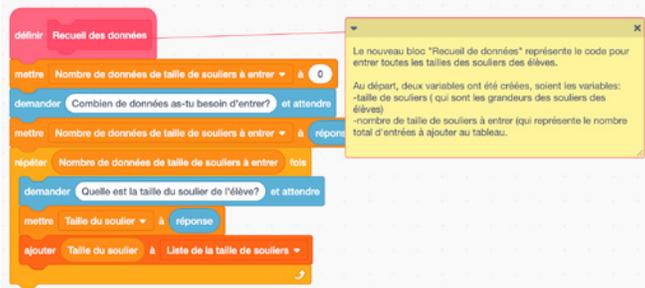
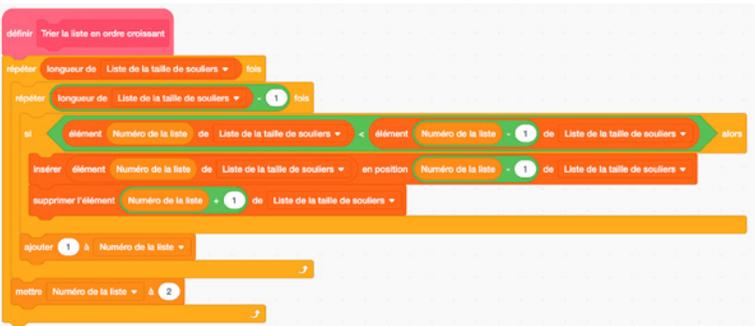
Exemple de code pour trouver la médiane :



Poser aux élèves les questions suivantes : Qu'arrivera-t-il si j'ajoute des pointures de souliers dans ma liste? La médiane restera-t-elle la même? Pourquoi?

OBSERVATIONS POSSIBLES	PISTES D'INTERVENTION
<p>L'élève ne réussit pas à entrer toutes les pointures des souliers des élèves dans une liste.</p>	<p>Dans quel bloc peux-tu poser une question?</p> <p>Quelle question peux-tu poser ici?</p> <p>De quelles variables as-tu besoin?</p> 

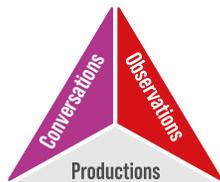
OBSERVATIONS POSSIBLES	PISTES D'INTERVENTION
<p>L'élève oublie de remettre les variables à zéro.</p>	<p>Que se produit-il lorsque tes variables ne sont pas remises à zéro en début de code? Quel est l'effet sur l'entrée des données dans la liste?</p>  <p>Demander à l'élève de retourner voir le pseudocode qui a été analysé au début du déroulement. Ce pseudocode contient aussi des blocs « mettre à 0 ». L'encourager à trouver le lien entre ces blocs dans les deux codes.</p>
<p>L'élève n'utilise pas les variables correctement.</p>	<p>Demander à l'élève ce que veulent dire les variables. La compréhension de la signification des variables utilisées est nécessaire à la compréhension du code.</p> <p>Il est possible que l'élève ne comprenne pas le nom des variables. L'encourager à renommer ses variables avec des termes familiers dans le contexte du problème.</p> 

OBSERVATIONS POSSIBLES	PISTES D'INTERVENTION
<p>L'élève ne réussit pas à créer de nouveaux blocs (blocs personnalisés).</p>	<p>Quelle catégorie de blocs peut t'aider à créer de nouveaux blocs? Quelle est sa couleur?</p> <p>Demander à l'élève les raisons pour lesquelles on crée de nouveaux blocs (pour rendre les codes plus efficaces en étant plus courts ou pour rendre le code plus facile à analyser ou à déboguer en isolant certaines séquences du code principal).</p>  
<p>L'élève ne réussit pas à créer un code pour calculer la moyenne.</p>	<p>Demander à l'élève de revoir son pseudocode pour la moyenne. L'encourager à repérer le calcul de la moyenne dans son pseudocode, puis identifier les blocs correspondants dans son code. S'il manque des blocs, ce serait la première étape.</p> <p>Revoir avec l'élève la signification des blocs et des variables. Est-ce que les bonnes variables sont aux bons endroits dans la formule?</p> <p>S'il y a lieu, faire en sorte que les valeurs des variables soient affichées à l'écran. De cette façon, l'élève aura un repère visuel des valeurs et pourra détecter s'il y a une erreur.</p>

OBSERVATIONS POSSIBLES	PISTES D'INTERVENTION
<p>L'élève ne réussit pas à créer un code afin d'ordonner les données en ordre croissant.</p>	<p>Ordonner des données en ordre croissant à l'aide de blocs peut demander de la logique. Demander à l'élève ce qu'il faut faire pour ordonner une liste en ordre croissant (par exemple, déterminer si chaque valeur est plus petite que la prochaine). Comment pourrait-on le faire comprendre à l'ordinateur?</p> <p>Montrer à l'élève les opérateurs « plus grand que » et « plus petit que » à titre d'indice.</p> <p>Inviter l'élève à effectuer une recherche dans Internet pour voir si ce problème a déjà une solution au lieu de créer tout un nouveau code.</p>
<p>L'élève ne réussit pas à créer un code pour trouver la médiane.</p>	<p>Demander à l'élève la signification de certains blocs opérateurs.</p> <p>Que signifie le bloc d'instructions conditionnelles SI, ALORS, SINON?</p> <p>Comment trouves-tu la médiane des nombres habituellement?</p> <p>Dans ce code, quelles sont les variables nécessaires pour trouver la médiane?</p> 

Objectivation (15 minutes)

L'évaluation
peut se faire par les...



Demander aux élèves de circuler dans la salle de classe et de regarder les pseudocodes et les codes des autres élèves.

- Que remarques-tu?
- Les pseudocodes de tes pairs sont-ils les mêmes que les tiens?
- Les codes de tes pairs sont-ils les mêmes que les tiens?
- Que changerais-tu dans tes codes? Quelles suggestions donnerais-tu à tes pairs?
- Quels sont les apprentissages faits aujourd'hui?

Afin de poursuivre les apprentissages en codage, continuer à s'exercer à passer d'un pseudocode à un code, ou d'un code à un pseudocode.

Demander aux élèves s'il y a d'autres façons de coder pour analyser des données (par exemple, tableur, tels Sheets et MS Excel).

Demander aux élèves d'expliquer l'avantage de créer un code lié aux mesures de tendance centrale dans un logiciel de codage par blocs, au lieu d'utiliser des commandes =MOYENNE et =MÉDIANE dans un tableur. (Le codage par blocs nous donne l'occasion de voir ce qui se passe en arrière-scène lorsqu'on utilise des raccourcis dans un tableur. La création de ce code permet d'explorer la puissance de blocs conditionnels [SI, ALORS, SINON] et de boucles [répéter, répéter jusqu'à]).

Consolidation

Proposer aux élèves une activité dans laquelle on peut générer une table de valeurs en utilisant un dénombrement prédéfini, c'est-à-dire une boucle dans laquelle on génère une liste de valeurs selon le rang et le terme.

Quelques exemples d'activités qui utilisent le bloc « répéter jusqu'à ce que... » :

- Créer un code qui fera le recueil d'un nombre exact de données (par exemple, répéter jusqu'à ce que le nombre de réponses = 10) afin de faciliter le calcul de certaines mesures de tendance centrale. Ici, le nombre pourrait aussi correspondre au nombre d'élèves du groupe-classe afin que chaque élève ait le droit à une seule réponse.
- Créer un code qui recueille des données jusqu'à ce qu'une somme maximale soit atteinte (par exemple, répéter jusqu'à ce que la somme des valeurs = 250). Ce genre de code pourrait être utile pour une entrée cumulative de points afin d'atteindre une cible. Il serait possible, par la suite, d'utiliser les mesures de tendance centrale pour déterminer la valeur moyenne ou la médiane de points pour chaque entrée.

ÉLÉMENTS À CONSIDÉRER

Liens avec les autres domaines mathématiques

Nombres

B2.1 Utiliser les propriétés et la priorité des opérations, et les relations entre les opérations pour résoudre des problèmes comportant des nombres naturels, des nombres décimaux, des fractions, des rapports, des taux et des pourcentages, y compris des problèmes à plusieurs étapes ou à plusieurs opérations.

En utilisant les instructions conditionnelles (SI, ALORS, SINON), il serait possible de créer un code qui peut analyser un problème mathématique et de déterminer les processus à suivre pour le résoudre en faisant appel à des sous-programmes (par exemple, des sous-programmes pour les priorités d'opérations, pour l'addition de fractions).

Algèbre

C4. Mettre en application le processus de modélisation mathématique pour représenter et analyser des situations de la vie quotidienne, ainsi que pour faire des prédictions et fournir des renseignements à leur sujet.

L'utilisation de sous-programmes et de blocs personnalisés permet de généraliser une situation en créant des modèles et en faisant appel aux modèles à divers moments dans le code.

Données

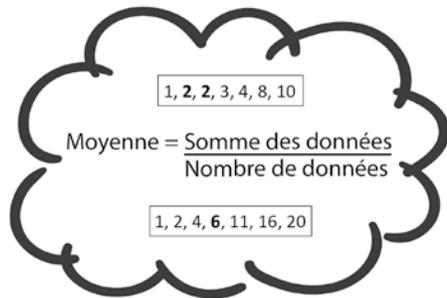
D1.5 Déterminer l'incidence de l'ajout ou de la suppression de données sur les mesures de tendance centrale et décrire comment ces changements modifient la représentation et la distribution des données.

En créant un sous-programme pour chaque mesure de tendance centrale, il serait possible de rédiger un code qui fait appel à chaque sous-programme à différents moments, créant un code plus efficace et facile à lire/à modifier.

Différenciation pédagogique et conception universelle de l'apprentissage

- Suggérer à l'élève qui cherche à relever un défi de créer un code pour trouver le mode de la liste de peintures de souliers des élèves du groupe-classe. Ce code utilisera la liste des données triées en ordre croissant, mais les conditions à déterminer seront très différentes.
- En utilisant un logiciel de codage où le partage de projets est possible, une partie du code pourrait déjà être en place pour que les élèves ne partent pas toujours d'un canevas blanc.
- Travailler au rythme des élèves. Transférer les connaissances et les concepts mathématiques à du codage peut prendre du temps. Allouer le temps nécessaire à chaque partie de la situation d'apprentissage.
- Fournir des pseudocodes aux élèves afin de pouvoir les transformer en codes. Le pseudocode peut aussi avoir un code de couleurs qui guidera l'élève vers la bonne catégorie de blocs dans le logiciel choisi.
- Fournir un aide-mémoire relatif à la fonction de certains blocs plus complexes, tels les opérateurs et les conditions.

Annexe 1 – Mise en situation



Annexe 2 – Pseudocode

Variables : nb1
nb2
somme
Départ
Mettre nb1 = 0.
Mettre nb2 = 0.
Mettre somme = 0.
Définir la valeur de nb1.
Poser la question « Quelle est la valeur de nb1? ».
La réponse devient la valeur de nb1.
Définir la valeur de nb2.
Poser la question « Quelle est la valeur de nb2? ».
La réponse devient la valeur de nb2.
Somme = nb1 + nb2
Fin

Situation d'apprentissage – 8^e année

Titre : Savez-vous planter des choux?

Durée : 150 minutes

Sommaire

Le terrain à côté de l'école, qui appartient à la municipalité, est à louer. Les élèves de 8^e année décident de faire une demande afin de planter un jardin communautaire et de vendre des pousses et des produits au marché.

Les élèves aimeraient amasser assez d'argent pour payer les coûts liés à l'achat de cotons ouatés pour toute la population étudiante de l'école. Le coût total serait de 10 000 \$.

Les élèves devront utiliser un tableur afin d'équilibrer le budget en tenant compte des dépenses fixes et des dépenses variables, ainsi que des revenus possibles. Les élèves devront faire des prédictions, prendre des décisions éclairées et les communiquer au reste de l'école.

Note : Le projet ne doit pas nécessairement être un jardin communautaire. Questionner les élèves afin de trouver une idée entrepreneuriale qui les intéresse.

Attentes et contenus d'apprentissage ciblés

Algèbre

- C3.** Résoudre des problèmes et créer des représentations de situations mathématiques de façons computationnelles à l'aide de concepts et d'habiletés en codage.
 - C3.1** Résoudre des problèmes et créer des représentations de situations mathématiques de façon computationnelle en écrivant et en exécutant des codes, y compris des codes comprenant l'analyse de données, afin de prendre des décisions éclairées et de les communiquer.

Littératie financière

- F1.** Démontrer les connaissances et les habiletés nécessaires pour prendre des décisions financières éclairées.
 - F1.2** Établir un plan visant à atteindre un objectif financier à long terme en tenant compte du revenu, des dépenses et des répercussions fiscales.
 - F1.3** Déterminer différentes façons de maintenir un budget équilibré et utiliser les outils appropriés pour faire le suivi de toutes les dépenses et de tous les revenus dans diverses situations.

Pratiques pédagogiques à fort impact en mathématiques à privilégier

Résultats d'apprentissage, critères d'évaluation et rétroaction descriptive

Avant de commencer à créer le tableur, le personnel enseignant détermine avec l'aide des élèves les résultats d'apprentissage et les critères d'évaluation. Cela rend l'intention de l'activité plus évidente, de sorte que les élèves peuvent avoir une vision plus claire de ce qui est attendu. Les élèves peuvent ainsi créer plus facilement des liens avec leur expérience personnelle et d'autres domaines ou matières.

Outils et représentations

L'utilisation d'outils dans cette activité aide l'élève à visualiser un concept qui est souvent très abstrait, soit un budget. L'utilisation d'un outil tel qu'un tableur électronique aide l'élève à voir les changements dans le budget et la somme d'argent qui manque avant d'atteindre son objectif financier.

Enseignement en petits groupes

Le personnel enseignant peut se servir de l'enseignement en petits groupes pour répondre plus facilement aux besoins de chaque élève. Les élèves en ressentent plus de confort lors de nouvelles situations. Le personnel enseignant pourra aussi renforcer la compréhension des diverses fonctions employées au cours de la préparation du budget.

Connaissances et habiletés préalables

Pour être en mesure de réaliser cette situation d'apprentissage, l'élève doit :

- maîtriser l'addition, la soustraction et la multiplication de sommes d'argent;
- avoir une certaine compréhension des mesures de tendance centrale, surtout de la moyenne;
- connaître l'interface d'un tableur électronique ou d'un logiciel à l'aide duquel on peut créer un tableur;
- avoir certaines connaissances propres à la littératie financière telles que l'importance d'équilibrer un budget et les différents éléments d'un plan financier.

Résultats d'apprentissage

À la fin de cette situation d'apprentissage, l'élève pourra :

- créer un tableur afin de représenter de façon efficace un budget équilibré;
- utiliser les fonctions du tableur afin d'automatiser certains calculs;
- représenter les dépenses et les revenus de la vente de plantes et de légumes;
- créer un plan visant à atteindre un objectif financier à long terme, soit l'achat de cotons ouatés pour toutes et tous les élèves de l'école.

Critères d'évaluation selon les grilles d'évaluation du rendement

Connaissance et compréhension

- L'élève représente un plan budgétaire en utilisant un tableur électronique.
- L'élève comprend des situations mathématiques de façon computationnelle.
- L'élève utilise des fonctions telles que la somme, la moyenne et le produit afin de calculer les dépenses et les revenus.

Habiletés de la pensée

- L'élève fait des prédictions et prend des décisions éclairées en utilisant un diagramme représentant son plan budgétaire.
- L'élève crée un plan financier en ayant en tête un objectif.

Communication

- L'élève communique clairement et efficacement avec ses pairs afin de justifier des ajouts et des adaptations au chiffré du projet.
- L'élève crée des représentations en utilisant un tableur et des diagrammes.

Mise en application

- L'élève établit des liens avec ses expériences personnelles afin de créer un budget.
- L'élève utilise des fonctions telles que la somme, la moyenne et le produit afin de calculer les dépenses et les revenus.

Matériel requis

- ordinateur qui a accès à un tableur électronique
- [annexe \(Mise en situation\)](#)

Note : Les exemples dans cette situation d'apprentissage ont été faites avec Google Sheets.

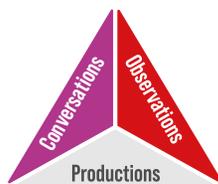
Vocabulaire mathématique

objectif financier, revenu, dépenses, budget équilibré, fixe, variable (dans le contexte financier), somme, produit, moyenne

CONTENU

Mise en situation (30 minutes)

L'évaluation
peut se faire par les...



Présenter aux élèves la photo ci-dessous [\[annexe\]](#) et leur demander de formuler une question mathématique liée à celle-ci. Les élèves peuvent aussi faire des observations relativement à la photo.

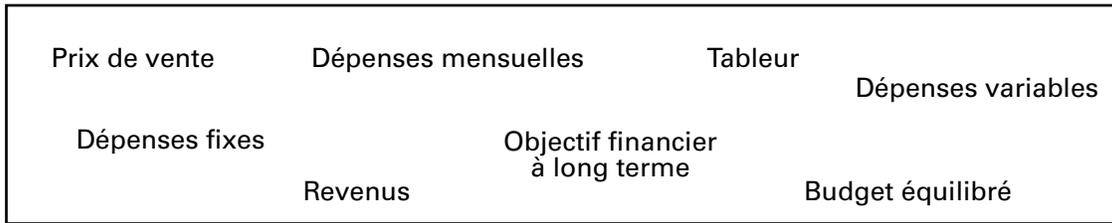


Réflexions recherchées :

- Combien d'argent gagneraient les élèves si leurs légumes étaient vendus au marché?
- À quels prix leurs produits devraient être vendus pour faire un profit?
- Comment bien tenir compte des ventes et des dépenses du jardin?

Proposer l'élaboration d'un plan budgétaire concernant la vente de produits, dont l'objectif financier est l'achat de cotons ouatés pour les élèves de l'école.

Avoir une discussion avec les élèves au sujet de l'importance d'un plan budgétaire. Discuter des composantes importantes d'un plan financier. Voici des exemples de sujets que pourraient cibler les élèves :



Questions d'élèves possibles :

- Que doit-on ajouter au tableur, autres que les dépenses et les revenus?
- Comment va-t-on tenir compte des dépenses et des revenus de mois en mois?
- Peut-on déterminer la somme d'argent qu'il nous reste à amasser pour atteindre notre objectif?

Établir un plan avec les élèves afin d'énumérer les dépenses fixes et variables qui seront dans leur budget. Il serait important de compter à part les dépenses uniques, qui ne seront pas répétées chaque mois, tel l'achat de graines et de terre.

OBSERVATIONS POSSIBLES	INTERVENTIONS POSSIBLES
L'élève ne sait pas par où commencer.	Demander à l'élève d'établir des liens avec son expérience personnelle. <ul style="list-style-type: none"> • Es-tu déjà allée ou allé au marché? • As-tu déjà fait un jardin?
L'élève ne peut pas relever les éléments importants d'un budget.	Quel est le but d'un budget? Qu'est-ce qu'on cherche à faire en préparant un budget? Pourquoi un budget est-il important?
L'élève ne peut pas différencier une dépense variable d'une dépense fixe.	Établir des liens avec l'expérience personnelle des élèves. <ul style="list-style-type: none"> • Quelle dépense tes parents et toi faites-vous quelques fois seulement [dépense variable]? • Quelle dépense tes parents et toi faites-vous de façon régulière [dépense fixe]?

Déroulement (60 minutes)

L'évaluation peut se faire par les...



Exemple de budget fait avec Google Sheets : [Savez-vous planter des choux?](#)

Demander aux élèves de rédiger, sur papier, un brouillon de leur budget et d'y insérer toutes les informations pertinentes relevées lors de la mise en situation. Leur demander ensuite de déterminer les calculs qui seront nécessaires au budget.

Réponses possibles

- Je dois additionner toutes les dépenses afin de connaître la somme d'argent que je dépense initialement et par mois. (Le code « =SOMME » sert à additionner une série de cellules au lieu de les additionner une à la fois.)
- Il faut déterminer le coût unitaire des plantes qui seront vendues, puis le multiplier par la quantité. (Dans un chiffrier, l'astérisque [*] représente souvent la multiplication. On peut aussi utiliser le code « =PRODUIT ».)
- Il faut déterminer la différence entre l'argent amassé et l'argent dépensé (revenus – dépenses).
- Il faut déterminer la somme d'argent qu'il nous reste à recueillir afin d'atteindre notre objectif (10 000 \$ – total amassé).
- J'aimerais que les cellules changent de couleur pour indiquer des problèmes potentiels; par exemple, si l'on dépense plus qu'on gagne dans un mois. (La mise en forme conditionnelle utilise le langage SI, ALORS pour changer les propriétés des cellules dans une zone donnée.)

Demander aux élèves de déterminer les fonctions nécessaires à la création de leur tableau.

Voici quelques exemples de fonctions présentes dans le tableau Google Sheets. Ces fonctions demeurent accessibles dans différents tableurs, mais l'endroit exact où les trouver dans les menus peut varier :

The image displays four screenshots of the Google Sheets interface, each showing a different menu with various functions highlighted.

- SOMME:** The 'Fonction' (Function) menu is open, showing a list of mathematical functions including SOMME, SOMME.CARRÉS, SOMME.SÉRIES, SOMME.SI, SOMME.SI.ENS, SOUS.TOTAL, TAN, and TANH.
- PRODUIT:** The 'Fonction' (Function) menu is open, showing a list of mathematical functions including PRODUIT, MULTINOMIALE, MUNIT, NB.SI, NB.SI.ENS, NB.VIDE, PAIR, PERCENTIF, PGCD, PHI, PI, PLAFOND, PLANCHER, PPCM, PRODUIT, PUISSANCE, QUOTIENT, RACINE, RACINE.PI, RADIANS, and RANDARRAY.
- MOYENNE:** The 'Fonction' (Function) menu is open, showing a list of mathematical functions including MOYENNE, NB, MAX, MIN, and various other functions like TOUTES, ANALYSEUR, BASE DE DONNÉES, DATE, FILTRER, FINANCES, GOOGLE, INFOS, INGÉNIERIE, LOGIQUE, MATHS, and OPÉRATEUR.
- MISE EN FORME CONDITIONNELLE:** The 'Mise en forme conditionnelle' (Conditional formatting) menu is open, showing options like 'Mise en forme conditionnelle', 'Validation des données', 'Obtenir le lien vers cette page', 'Trier une plage', 'Plage aléatoire', 'Définir une plage nommée', and 'Protéger la plage'.

Note : L'élève peut changer l'apparence des cellules dans le tableur, selon les données qui s'y trouvent, au moyen de la mise en forme conditionnelle. Cela pourrait être très utile afin de visualiser les données dans un budget, même s'il ne s'agit pas d'une formule mathématique comme telle.

Lancer aux élèves le défi de planifier un budget pour le projet de jardin communautaire. Les encourager à explorer le logiciel et à tenter d'appliquer les fonctions mentionnées ci-dessus.

Exemples de tableaux et de fonctions relatifs au budget :

Exemples de dépenses uniques :

Dépense unique	Montant
Achat de graines	189,38 \$
Achat de pousses	170,58 \$
Achat de terre	600,00 \$
Compost	150,00 \$
Bannière promotionnelle	1 164,96 \$ × 55,00 \$
Total	=SOMME(B7:B11)

Exemples de dépenses mensuelles fixes :

Dépense mensuelle fixe	Montant
Location du terrain	300,00 \$
Location du stand	80,00 \$
Location de l'autobus	680,00 \$ × 300,00 \$
Total	=SOMME(B15:B17)

Exemples de dépenses mensuelles variables :

Dépense mensuelle variable	Montant
Paieiment d'eau	
Achat de boîtes pour vente	
Achat de sacs en papier	0,00 \$ ×
Total	=SOMME(B22:B25)

Note : Les fonctions les plus utilisées seront SOMME pour l'addition et la soustraction (lorsque le signe négatif est présent pour une donnée, le logiciel soustraira cette donnée dans la fonction « SOMME »). La fonction PRODUIT sera aussi utilisée au cours du calcul des revenus.

Exemple de SOMME :

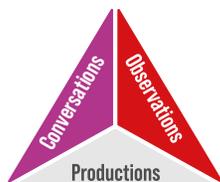
Sommaire du mois	
Revenus	0,00 \$
Dépenses	1 844,96 \$
	-1 844,96 \$ ×
Total	=SOMME(K5-K6)

OBSERVATIONS POSSIBLES	INTERVENTIONS POSSIBLES
L'élève n'utilise pas la fonction « SOMME ».	Demander à l'élève s'il y a une fonction relevée au début qui pourrait être utilisée pour simplifier la tâche. Rappeler à l'élève que l'utilisation du tableur simplifie la tâche de budgéter.
L'élève n'arrive pas à organiser le tableur de façon efficace.	Demander à l'élève de se référer au croquis du début. Proposer à l'élève d'utiliser des couleurs pour indiquer les différentes dépenses et les divers revenus, puis les organiser.
L'élève ne parvient pas à faire une soustraction.	Souligner que l'on peut soustraire deux cellules en insérant un calcul de base. Note : Les logiciels en français n'ont pas la fonction « MINUS » comme en anglais. Il suffit d'insérer le symbole d'égalité, suivi du calcul désiré.

Sommaire du mois	
Revenus	0,00 \$
Dépenses	1 844,96 \$
	-1 844,96 \$ x
Total	=SOMME(K5-K6)

Objectivation (60 minutes)

L'évaluation
peut se faire par les...



Regrouper les élèves et leur demander de partager le tableur créé avec les autres élèves du groupe-classe.
Faire ressortir les éléments qui se ressemblent et ceux qui diffèrent.

Réflexions possibles :

- Certains budgets sont plus ou moins réalistes que d'autres.
- Mon plan tient compte de certaines dépenses que d'autres plans ont omises (la réflexion contraire pourrait aussi être pertinente).
- En faisant un budget, on se rend compte que lancer une entreprise peut être complexe!

Demander aux élèves s'il est possible de créer un plan pour le groupe-classe en regroupant certains éléments de tous les plans.

- Ce nouveau plan est-il réaliste?
- Ce nouveau plan permettra-t-il d'atteindre l'objectif d'amasser 10 000 \$?

Consolidation

Demander aux élèves de faire un budget en utilisant un tableur pour une étudiante ou un étudiant de première année d'un apprentissage, du postsecondaire ou du marché de travail. Ce tableur devra comprendre les dépenses liées à la vie professionnelle ou à la vie étudiante (frais de scolarité, frais d'apprentissage, achat d'outils, transport, formation, etc.) et à la vie personnelle (loyer, factures, épicerie, etc.), ainsi que les sources de revenus possibles (emploi, bourses d'études, prêts, dons, etc.). Les élèves peuvent utiliser le plan budgétaire du jardin comme point de départ.

ÉLÉMENTS À CONSIDÉRER

Liens avec les autres domaines mathématiques

Données

D1.2 Collecter des données continues pour répondre à des questions d'intérêt concernant deux variables et organiser les ensembles de données de façon appropriée dans une table de valeurs.

Il serait possible de recueillir des données, soit manuellement ou avec un logiciel de sondage, puis de transférer les données dans un tableur pour les analyser et les interpréter au moyen de séquences de code.

Sens de l'espace

E2.4 Expliquer le théorème de Pythagore en utilisant divers modèles géométriques et se servir du théorème pour calculer la mesure de longueur manquante d'un côté d'un triangle rectangle donné.

Il serait possible de créer un code pour tracer des triangles rectangles en utilisant une table de valeurs qui donne deux des trois longueurs de côtés. Alternativement, l'élève pourrait créer un code qui génère une table de valeurs liée aux longueurs mesurées des côtés a et b , puis générer la longueur de l'hypoténuse.

Littératie financière

F1.4 Déterminer la valeur croissante d'intérêts simples et composés à divers taux à l'aide d'outils technologiques, et expliquer l'incidence des intérêts dans la planification financière à long terme.

En utilisant les opérations liées aux intérêts simples et composés, les élèves pourront générer des listes de valeurs d'investissement ou de prêt sur un temps donné afin de les comparer.

Différenciation pédagogique et conception universelle de l'apprentissage

- Offrir un exemple de budget aux élèves. L'exemple pourrait être entièrement vide afin de leur donner une idée de la structure d'un budget. Il pourrait aussi comprendre certaines données visant à activer les connaissances des élèves.
- Inviter les élèves à travailler en équipes, soit pour encourager l'apprentissage par les pairs ou pour favoriser l'enseignement explicite et personnalisé selon leurs besoins.
- Créer des référentiels des différentes fonctions mathématiques à l'étude et des façons de les insérer dans le tableur.
- Présenter certaines données sous forme de représentations graphiques. Lui demander d'explorer les outils de représentation graphique afin de trouver des représentations visuelles qui pourraient être utiles et pertinentes.

Annexe – Mise en situation



4. ET ENSUITE...

L'expérience de l'élève

Les liens entre le codage et l'enseignement des STIM (des problèmes modernes nécessitent des solutions modernes)

Afin de faire vivre aux élèves l'expérience mathématique, il est important de tenir compte de l'omniprésence de la technologie dans notre société et de son évolution constante. Dans la salle de classe, il est possible d'intégrer plusieurs matières au programme-cadre de mathématiques afin de les contextualiser et de les rendre plus accessibles et pertinentes.

L'enseignement des STIM vise un apprentissage interdisciplinaire des sciences, de la technologie, de l'ingénierie et des mathématiques. Étant donné qu'il est difficile, voire impossible, d'appliquer les disciplines STIM à la résolution de problèmes sociétaux sans tenir compte de l'aspect esthétique et artistique des solutions proposées, la lettre « a », pour arts, est ajoutée à l'acronyme, soit STIAM. En intégrant les différentes disciplines STIM dans l'enseignement, les élèves évoluent dans des contextes authentiques pour trouver des solutions à des problèmes réels et modernes.

Le codage peut servir de point de départ et de fil conducteur à l'enseignement des STIM en salle de classe. Puisque le codage n'est pas un domaine en soi, mais plutôt une approche servant à interpréter un problème, à le représenter et à le résoudre, il peut être utilisé pour faire des liens entre les domaines mathématiques et les autres matières. Cette intégration de concepts mathématiques ciblés à un autre programme-cadre, par exemple, à celui de sciences et technologie, permet de contextualiser les mathématiques pour rendre l'apprentissage de la matière plus accessible aux élèves et évaluer plusieurs connaissances et habiletés en même temps. Voici quelques exemples d'intégration du codage dans les disciplines STIM.

S - Sciences



Les sciences comprennent plusieurs sous-domaines, comme la biologie, la chimie, la physique, l'astronomie et les sciences de la Terre. Les avancements réalisés dans tous ces domaines ont été possibles grâce notamment à la robotique et au codage. Les élèves ne sont peut-être pas conscientes et conscients de l'importance qu'occupent le codage et la robotique en sciences. Les satellites dans l'espace en sont un excellent exemple. Étant donné que les satellites n'ont pas d'équipage, plusieurs processus importants doivent être automatisés afin que ces engins spatiaux puissent parcourir des distances exceptionnelles ou maintenir une orbite autour d'un astre.

En salle de classe, les élèves peuvent utiliser un logiciel de codage pour tracer une forme ronde autour d'un objet, comme si c'était un satellite en orbite. Pour un plus grand défi, l'objet autour duquel le satellite fait son orbite pourrait être en mouvement. Les élèves pourraient aussi créer un code pour programmer un robot qui ferait une collecte de données, comme des données météorologiques, et les retournerait pour l'analyse. Des robots, comme LEGO Mindstorm, LEGO WeDo et Dash et Dot, ou des microcontrôleurs, comme micro:bit et Arduino, pourraient servir de capteurs à cet effet.

Les outils utilisés en médecine, comme les moniteurs de fréquence cardiaque ou les glycomètres dont se servent les gens souffrant de diabète, sont un autre exemple de codage en sciences. Ces outils sont essentiels à la survie de plusieurs personnes et fonctionnent à partir d'instructions conditionnelles.

En salle de classe, les élèves peuvent créer leur propre code pour effectuer la collecte de données liées à la fréquence cardiaque en utilisant des microcontrôleurs, comme Arduino, micro:bit et Grove Zero. Les élèves peuvent ensuite se servir d'un tableur pour y entrer leurs données et les représenter à l'aide d'une variété de diagrammes.



T – Technologie



La technologie est la discipline STIM qui est la plus souvent et la plus facilement associée au codage. Elle est utilisée dans plusieurs milieux de travail, entre autres l'agriculture et la médecine.

Les fermières et les fermiers utilisent la technologie afin d'exploiter efficacement de grandes terres agricoles. Les systèmes industriels d'irrigation et de semence sont souvent partiellement automatisés. Ils utilisent des capteurs pour déterminer la température, l'humidité et la composition chimique du sol. Les capteurs permettent aussi l'autoguidage de la machinerie agricole et l'utilisation du GPS afin d'assurer le bon positionnement des systèmes en question.

En salle de classe, les élèves peuvent effectuer des simulations de systèmes d'irrigation, comme ceux qu'utilisent les agricultrices et les agriculteurs, ou créer une version miniature d'un appareil d'irrigation à l'aide d'Arduino ou d'autres microcontrôleurs.

Les fauteuils roulants motorisés sont un atout pour les personnes à mobilité réduite, car ils leur procurent une certaine autonomie. Ces fauteuils motorisés ne pourraient pas exister sans la technologie de la fabrication et la programmation. Il y a plusieurs entrées, par exemple, le mouvement d'une manette, et plusieurs sorties, comme le déplacement dans le sens désiré, qui doivent être programmées. Les systèmes sont testés de manière rigoureuse afin d'assurer la sécurité de l'utilisatrice ou de l'utilisateur.



En salle de classe, les élèves peuvent explorer un problème d'accessibilité à l'aide de l'automatisation et de la robotique.

I – Ingénierie

L'ingénierie est un domaine qui comprend une multitude d'éléments. Elle consiste en l'« étude d'un projet industriel sous tous ses aspects (techniques, économiques, financiers, monétaires et sociaux), qui nécessite un travail de synthèse coordonnant les travaux de plusieurs équipes de spécialistes¹ ». Plusieurs aspects du travail d'ingénierie peuvent être facilités à l'aide du codage.

Le codage par blocs ou par texte est aussi une forme d'ingénierie. La personne qui code crée des structures qui interagissent de manière efficace et harmonieuse. Elle doit tester le code dans plusieurs scénarios afin d'en assurer le bon fonctionnement avant de rendre le produit final accessible au public cible. Ce processus ressemble énormément au processus d'ingénierie associé à la conception d'une structure, comme un bâtiment ou un pont, ou à celle d'une machine. Il existe un domaine de l'ingénierie électrique qui s'appelle l'ingénierie informatique, dans lequel les ingénieures et les ingénieurs développent des appareils informatiques ainsi que les codes qui les font fonctionner.

¹ Larousse.fr. (s. d.). Dictionnaire de français. [En ligne]

Les ingénieures et les ingénieurs peuvent utiliser le codage afin de simplifier leur travail et d'être plus efficaces. Des processus mécaniques répétitifs peuvent être automatisés, à l'aide de boucles, et des logiciels de design peuvent simuler la résistance d'une structure en appliquant des instructions conditionnelles. Ce ne sont que deux exemples simplifiés de l'utilisation du codage dans un contexte d'ingénierie.

En salle de classe, les élèves peuvent utiliser le codage dans un contexte de robotique afin d'automatiser un processus; par exemple, un bras mécanique qui dépose un objet sur une chaîne d'assemblage. Les élèves peuvent aussi créer des codes afin d'automatiser les calculs qui détermineront les coûts d'un projet d'ingénierie en se servant d'un tableur et de formules mathématiques.



M – Mathématiques



Le codage ne peut exister sans les mathématiques. Que ce soit du codage en blocs ou textuel, la création du code dépend d'une multitude de calculs et de problèmes de logique. Les boucles, par exemple, ne sont que des additions répétées ou des multiplications, et les instructions conditionnelles dépendent d'égalités et d'inégalités afin de connaître la façon de procéder.

Le codage, en mathématiques, est souvent employé dans le domaine des finances, par exemple au cours de l'utilisation de logiciels qui utilisent des fonctions et des conditions afin de comptabiliser des dépenses et des revenus et équilibrer des budgets. On utilise des instructions conditionnelles afin de visualiser de façon rapide l'équilibre du budget ou des calculs (par exemple, en mettant en surbrillance des données qui correspondent aux conditions programmées).

En salle de classe, les élèves peuvent aussi créer un budget et s'assurer qu'il est équilibré de façon rapide à l'aide de fonctions et d'instructions conditionnelles.

Le codage est aussi utilisé dans des contextes de raisonnement spatial. Étant donné que l'automatisation prend de l'importance dans notre monde, il est important d'avoir des gens habiles en codage qui créeront des programmes permettant à des robots de faire des livraisons ou à des voitures de conduire de manière autonome. Ces personnes doivent tenir compte des distances, des composantes géographiques, comme le relief du terrain, des conditions météorologiques et des composantes physiques, comme les routes et les feux de circulation, pour programmer le trajet le plus efficace et le plus sécuritaire possible afin que les robots puissent effectuer les déplacements souhaités.



En salle de classe, les élèves peuvent programmer un robot en vue de lui faire suivre un trajet particulier ou de trouver le trajet le plus efficace possible pour que le robot se rende à un point donné, et ce, en utilisant des cartes électroniques et en faisant déplacer le robot sur le parcours planifié. Des robots programmables tels que le LEGO Mindstorm, le LEGO WeDo, le Bee-bot, le Blu-bot, les Ozobots ou les Dash et Dot peuvent tous être utiles. Certaines des plateformes ont aussi des simulateurs en ligne.

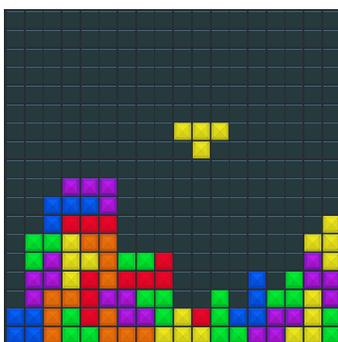
Les liens entre le codage et la vie quotidienne

L'établissement d'un lien entre les mathématiques et la vie de l'élève, y compris ses champs d'intérêt et ses expériences, peut non seulement aider à rendre concret des concepts qui sont parfois très abstraits, mais peut aussi nourrir, chez l'élève, un désir d'apprendre. Le lien entre le codage et la réalité existe même avant que l'élève réalise à cause de l'omniprésence de l'informatique et de la vitesse avec laquelle se développe la technologie.

Le jeu vidéo est une bonne façon de relier la vie de l'élève au codage. Le code qui fait fonctionner un jeu vidéo est très mathématique.

Voici quelques exemples de liens entre le jeu vidéo, le codage et le mathématiques :

- Afin de faire bouger un personnage, que ce soit un avatar, un véhicule ou un bloc, comme la version originale du jeu *Tetris*[®], il faut coder des directives pour faire agir le personnage en fonction des entrées, soit la souris, le clavier ou la manette. Le personnage est dans un plan cartésien, ce qui permet de connaître sa position exacte. Ainsi, il est possible de le faire interagir avec d'autres personnages ou des personnages non joueurs (PNJ).



- Les statistiques associées aux différents avatars sont une forme de modélisation mathématique. La personne qui écrit le code crée un avatar de base comportant des statistiques modifiables. Puis, à l'aide de cet avatar, elle crée tous les autres. Dans un jeu de courses de voitures, par exemple, les voitures ont toutes des vitesses et des capacités différentes. En créant un seul code maître qui comporte la possibilité de modifier ces deux paramètres, cela rend la programmation beaucoup plus facile.



Ce ne sont que deux exemples d'un nombre illimité de possibilités. Il est donc important de poser des questions aux élèves afin de leur faire prendre conscience des éléments mathématiques que comportent les jeux vidéo.

Le jeu vidéo est un fil conducteur efficace entre la vie de l'élève, le codage et les mathématiques, mais il faut se souvenir que ce n'est pas tout le monde qui joue aux jeux vidéo. Il est donc important de leur offrir d'autres exemples liés au codage.

En voici quelques-uns :

- Le code à barres qui détermine le prix d'un objet nécessite du codage. Les distances entre les barres sur le code à barres sont mesurées avec une précision exceptionnelle afin de déterminer l'identité du produit. Le prix du produit est programmé dans le système de caisse du magasin. Cela ne prend que quelques secondes, mais des mathématiques complexes sont en jeu à chaque achat. Les codes à barres 2D (codes QR) fonctionnent d'une façon similaire, mais peuvent contenir beaucoup plus d'informations.



Ce lien pourrait aussi être une porte d'entrée vers la littératie financière, surtout liée à la consommation responsable, comme la création de budgets ainsi que les avantages et les désavantages de divers modes de paiement.

- Les sites Web et les médias en ligne, qu'il s'agisse de médias sociaux ou de services de diffusion de télévision et de films, utilisent des algorithmes complexes afin de faire des recommandations à l'utilisatrice ou à l'utilisateur. Les plateformes proposant des films et des séries, par exemple, se servent du code pour traiter les données liées aux visionnements. Elles créent des listes de films écoutés qui sont ensuite triées. Le code détermine les éléments en commun et génère des suggestions de films. C'est le même fonctionnement pour les sites Web ou les médias sociaux.

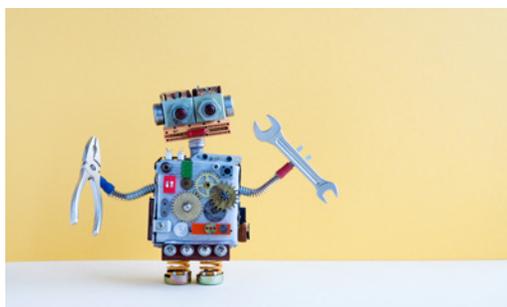


Les élèves qui consultent des sites Web, par exemple, pour en apprendre davantage au sujet de personnalités célèbres, et qui utilisent des médias en ligne, ainsi que les élèves qui aspirent à créer du contenu pour le Web, doivent bien connaître la façon dont ces algorithmes fonctionnent.

- Les appareils électroménagers modernes fonctionnent, eux aussi, à l'aide d'un code. La machine à laver, le sèche-linge et le lave-vaisselle ont tous des codes qui associent les boutons de la machine aux commandes auxquelles ils sont liés. Des capteurs de température et d'humidité, par exemple, collectent des données. Puis, des instructions conditionnelles sont utilisées afin d'assurer que la bonne température est atteinte. C'est la même chose concernant le réfrigérateur, le congélateur, le micro-ondes, etc. De plus, certains électroménagers peuvent se connecter à un réseau Internet sans fil. Cette connexion nécessite un code afin d'envoyer et de recevoir de l'information par le réseau.



Les élèves ne manifestent peut-être pas un intérêt particulier pour les appareils électroménagers, mais ce lien peut être une belle façon d'explorer la robotique. Le monde de la robotique est beaucoup plus que la création de robots. Le but de la robotique est d'utiliser la technologie et le codage afin de résoudre des problèmes et de rendre des processus plus efficaces.



Les liens entre le codage et la vie quotidienne de l'élève sont nombreux, voire illimités. Les exemples présentés n'en sont qu'un aperçu. Afin de permettre aux élèves de profiter pleinement de l'intégration du codage dans le cours de mathématiques, il est important de leur allouer du temps pour explorer et découvrir la manière dont le codage s'intègre dans leur vie.

Les liens entre le codage et les carrières

Il est impossible de prédire le marché du travail de l'avenir. Les carrières d'aujourd'hui peuvent disparaître ou évoluer à un point tel qu'elles ne seront plus reconnaissables. Pour cette raison, il est important d'outiller les élèves à ne pas cibler des carrières spécifiques, mais plutôt à relever des défis. Cette approche permet une plus grande flexibilité sur le plan du cheminement de carrière. Le codage joue un rôle important dans cette approche en développant une identité positive liée aux mathématiques.

Le codage nécessite l'apprentissage d'une « langue », soit le langage informatique. En effet, il faut assembler des morceaux, que ce soit des blocs ou du texte, afin de créer un tout, un peu comme créer une phrase avec des mots ou un paragraphe à l'aide de phrases. De plus, le partage de codes nécessite la rédaction de commentaires expliquant les décisions prises au moment de la programmation. Le codage aide donc à développer des habiletés langagières et de communication, qui sont utilisées avec l'ordinateur, mais aussi avec d'autres élèves. Cette capacité à communiquer dans divers contextes et pour différents auditoires sera un atout, peu importe l'emploi que l'élève occupera sur le marché du travail.

La technologie évolue rapidement. De plus en plus de tâches et de processus sont informatisés. Savoir coder offre plusieurs avantages dans une telle réalité. Coder va plus loin que l'apprentissage d'un nouveau langage, cela donne à l'élève une nouvelle façon de représenter un problème et de le résoudre. Cette flexibilité de la pensée devient un atout lorsque l'élève essaie de relever un défi. Le codage développe chez l'élève la résilience, la pensée critique et la créativité.

Le Système de projection des professions au Canada (SPPC), dans son rapport contenant ses prédictions de carrières pour les années 2019 à 2028, explique que l'automatisation et l'avancement de la technologie créeront un très grand nombre d'emplois, même si ces processus rendront certains emplois désuets. De plus, un rapport publié par Deloitte intitulé *La révolution de l'intelligence* souligne que l'être humain et l'ordinateur ont des habiletés différentes, mais complémentaires. Ces constats montrent l'importance de donner aux élèves l'occasion de se familiariser avec l'ordinateur, comme outil, et aussi avec tous les processus liés au fonctionnement d'un ordinateur lorsqu'il effectue une tâche. Tout ce que l'ordinateur peut accomplir est rendu possible par l'exécution du code. Une compréhension du fonctionnement du codage permet à l'élève de mieux comprendre les appareils utilisés au quotidien.

Les connaissances et les habiletés qu'acquiert les élèves, à l'aide du codage, leur donnent un avantage dans leurs études postsecondaires, dans le contexte d'un apprentissage ou au moment d'intégrer le marché du travail. Le codage devient une compétence précieuse, entre autres, dans les domaines des sciences, des mathématiques, de l'ingénierie et de l'informatique. Les aptitudes liées au codage aident, entre autres, à fortifier la résolution de problèmes, à développer l'esprit collaboratif, à stimuler la curiosité et à favoriser le questionnement. La résolution de problèmes est chose commune et fréquente dans le monde d'aujourd'hui et de demain. Posséder des compétences en codage permet de résoudre des défis dans plusieurs domaines d'emploi et dans de nombreux aspects d'un travail.

La terminologie du codage

TERME FRANÇAIS	TERME ANGLAIS	COURTE DESCRIPTION
ACTIONNEUR	ACTUATOR	Permet de faire des actions dans un système automatisé. Un moteur est un actionneur.
ALÉATOIRE	RANDOM	Une valeur déterminée au hasard.
AFFICHER	PRINT	Affiche des caractères à l'écran.
APPEL DE FONCTION	FUNCTION CALL	Les instructions exécutent le sous-programme.
ARGUMENT	ARGUMENT	La valeur que l'on donne à la variable lorsqu'on appelle une fonction (sous-programme).
ARRONDIR	ROUND	Permet d'arrondir une valeur numérique.
ATTENDRE JUSQU'À...	WAIT UNTIL...	Exécute du code jusqu'à ce qu'une condition devienne vraie. C'est le contraire de la boucle « tant que... ».
AU DÉMARRAGE	ON START	Contient le code qui sera exécuté lorsque le microcontrôleur démarrera.
BALISE	TAG	En HTML, le code qui est limité par les caractères < > (balises d'ouverture et de fermeture).

TERME FRANÇAIS	TERME ANGLAIS	COURTE DESCRIPTION
BOOLÉEN	BOOLEAN	Une valeur qui peut être soit vraie, soit fausse. Dans les langages par blocs, les valeurs booléennes sont souvent représentées par un hexagone.
BOUCLE	LOOP	Permet de répéter des instructions.
BROCHE	PIN	Une petite tige métallique qui permet de brancher une composante électronique à un microcontrôleur. Les broches sont numérotées, ce qui permet d'en choisir une dans le code.
BROCHE ANALOGIQUE	ANALOGUE PIN	Une broche qui peut envoyer ou recevoir un intervalle de valeurs numériques.
BROCHE NUMÉRIQUE	DIGITAL PIN	Une broche qui peut envoyer ou recevoir uniquement deux valeurs, habituellement ON ou OFF (ou HIGH ou LOW).
CAPTEUR	SENSOR	Composante permettant de recueillir des données comme un capteur de température ou de luminosité.
CARACTÈRE	CHARACTER	Signe permettant de former des mots et du texte, ce qui comprend les lettres de l'alphabet en majuscule et en minuscule et les chiffres.
CARTOGRAPHIER	MAP	Permet de convertir une valeur d'une échelle à une autre; par exemple, si un capteur donne des valeurs entre 168 et 875, on peut ramener les valeurs sur une échelle de 0 à 100.
CHAÎNE DE CARACTÈRES	STRING	Plusieurs caractères regroupés; par exemple, des mots, des phrases ou un mot de passe.
CHAQUE...	EVERY...	Une boucle qui se répète selon un certain intervalle; par exemple, toutes les 5 secondes.
COMMENTAIRE	COMMENT	Du code ignoré par la machine, mais qui permet aux êtres humains d'ajouter des annotations.
COMPARAISON	COMPARISON	Permet de comparer deux valeurs ou deux expressions; par exemple, $>$, $<$, $=$, \leq , \geq .
CONCATÉINATION	JOIN	Permet de réunir bout à bout plusieurs chaînes de caractères. Dans les langages par blocs, on parle souvent de « joindre » ou de « regrouper ».

TERME FRANÇAIS	TERME ANGLAIS	COURTE DESCRIPTION
COORDONNÉES	COORDINATES	Position d'un objet dans un plan exprimé à l'aide d'une valeur en x (horizontal) et d'une valeur en y (vertical) ou dans un espace en 3 dimensions [x , y et z]. Attention, certaines interfaces de codage utiliseront les quatre quadrants du plan cartésien que l'élève connaît, alors que d'autres interfaces utiliseront le coin en haut à gauche comme étant le point (0, 0), les valeurs augmentant en x en allant vers la droite et augmentant en y en allant vers le bas.
DÉBOGAGE	DEBUGGING	Trouver et éliminer les erreurs dans le code.
DÉCLARER	DECLARE	Permet d'introduire une nouvelle variable.
DÉFINIR	DEFINE	Après avoir introduit une variable, permet de lui donner une valeur.
DEL	LED	Diode électroluminescente : une composante électronique qui produit de la lumière.
DISCRET	DISCRETE	Il s'agit de données qui sont distinctes ou discontinues.
ENTIER	INTEGER	Un nombre qui n'est ni une fraction ni un nombre décimal (cela comprend les entiers positifs et négatifs). Une variable peut être déclarée comme contenant uniquement des nombres entiers.
ENTRÉE	INPUT	Introduire des données dans un ordinateur ou un microcontrôleur.
ÉVÉNEMENT	EVENT	Permet de déclencher une partie du code; par exemple, lorsqu'on clique sur le drapeau, sur un bouton, lorsque l'intensité sonore dépasse une certaine valeur.
FONCTION	FUNCTION	Sous-programme qui peut retourner une valeur.
INCRÉMENTER	INCREMENT	Ajouter de façon répétitive une valeur fixe à une variable. Remarque : Il est possible d'ajouter une valeur négative pour diminuer la valeur.
INDEX	INDEX	Une variable permettant de repérer la position d'un élément dans une liste.
INTERVALLE	RANGE	Intervalle entre deux valeurs.
LISTE	LIST	Variable contenant plus d'une valeur souvent appelée <i>tableau à une dimension</i> .
LOGIGRAMME	LOGIC DIAGRAM	Schéma qui montre la structure d'un processus.
LONGUEUR	LEN	Raccourci de « length », permet de déterminer la longueur d'une liste.

TERME FRANÇAIS	TERME ANGLAIS	COURTE DESCRIPTION
MICROCONTRÔLEUR	MICROCONTROLLER	Petit circuit électronique permettant de contrôler un dispositif (par exemple, Micro:bit, Arduino, Hub LEGO).
MINIFIER	MINIFY	Rendre le code plus simple et plus compact.
MODULO ou RESTE	MOD ou REMAINDER	Le reste d'une division (par exemple, 27 MOD 5 donne 2).
OBJET	OBJECT	Élément de base qui permet de coder. Les <i>sprites</i> sont des objets. Les objets ont une identité, des propriétés, des comportements et du code qui leur est rattaché.
OPÉRATEUR	OPERATOR	Symbole qui représente une opération arithmétique (par exemple, +, -, *, /) ou logique (par exemple, ET, OU, >, <, =).
OPÉRANDE	OPERAND	Donnée ou valeur entrant dans une opération arithmétique. Dans les langages par blocs, il y a souvent des trous dans les blocs opérateurs pour y insérer les opérandes.
PARAMÈTRE	PARAMETER	Une variable que l'on transmet à une fonction (sous-programme).
PAUSE	WAIT	Permet d'insérer un temps d'attente dans le code. La pause est souvent exprimée en secondes ou en millisecondes.
PLAFOND	MAX	La valeur la plus grande.
PLANCHER	MIN	La valeur la plus petite.
POINTEUR	POINTER	Symbole à l'écran souvent en forme de flèche. Dans le code, on peut créer des structures conditionnelles « quand le pointeur touche... alors... ».
POUR... DE 0 à...	FOR... FROM 0 to...	Boucle se répétant un certain nombre de fois.
PSEUDOCODE	PSEUDOCODE	Description de l'algorithme dans le langage de tous les jours avant de le convertir en code pouvant être compris par la machine.
RÉCURSION INFINIE	INFINITE RECURSION	Lorsqu'une fonction s'appelle elle-même, une boucle infinie est créée.
SI... ALORS... SINON...	IF... THEN... ELSE...	Utilisé pour créer une structure conditionnelle dans laquelle il y a deux options.
SORTIE	OUTPUT	Émettre des informations vers l'extérieur du microcontrôleur.
STRUCTURE DE CONTRÔLE	CONTROL STRUCTURES	Structure du code qui permet de changer le déroulement du programme (par exemple, les boucles, les instructions conditionnelles).

TERME FRANÇAIS	TERME ANGLAIS	COURTE DESCRIPTION
TABLEAU	ARRAY	Structure de données à deux ou à plusieurs dimensions.
TANT QUE...	WHILE...	Une boucle qui s'exécute tant qu'une condition est vraie. C'est le contraire d'une boucle « attendre jusqu'à ce que... ».
TOUJOURS	FOREVER	Une boucle qui se répète indéfiniment.
TRONQUER	TRUNCATE	Enlever la partie décimale (sans utiliser les règles pour arrondir).
VALEUR NUMÉRIQUE	NUMERICAL VALUE	Une valeur qui est un nombre. Dans les langages par blocs, les valeurs numériques sont souvent représentées par une forme arrondie.
VARIABLE	VARIABLE	Peut prendre différentes valeurs pendant le déroulement du code.
VARIABLE PUBLIQUE	GLOBAL VARIABLE	Variable qui est accessible dans tout le programme.
VARIABLE PRIVÉE	LOCAL VARIABLE	Variable qui est accessible uniquement dans une fonction ou un <i>sprite</i> .

ANNEXE – MODÉLISATION MATHÉMATIQUE

COMPRENDRE LE PROBLÈME

- Inviter les élèves à formuler des questions liées au problème à résoudre.
- Animer une discussion afin que les élèves échangent sur les questions formulées. S'assurer, au cours des discussions, que les questions proposées ont un lien avec les concepts à l'étude.
- En petits groupes, inviter les élèves à évaluer les questions posées et à prendre des décisions réfléchies qui permettent de les classer par ordre de priorité.
- Demander à chaque équipe de choisir une question essentielle pour laquelle un modèle sera développé afin d'y répondre.
- Poser aux élèves les questions « Quels sont les renseignements que nous avons déjà? » et « De quels renseignements avons-nous besoin pour résoudre le problème de la question ciblée? ».
- Inviter les élèves à déterminer tous les éléments nécessaires à la résolution du problème (par exemple, liste du matériel, prix, temps nécessaire).
- Garder en tête que des groupes ou des élèves peuvent travailler sur différents problèmes liés à la même situation.
- Mentionner aux élèves qu'il y a plusieurs façons de résoudre le problème. Les inviter à déterminer les données manquantes en effectuant des recherches.

Observation	Pistes de questionnement et interventions possibles
L'élève a de la difficulté à se poser des questions liées au contenu visé.	<ul style="list-style-type: none"> Que veux-tu savoir? (PM) Qu'est-ce que tu comprends du problème? (PM) Quelles émotions ressens-tu actuellement? Peux-tu expliquer ou dessiner la raison pour laquelle tu te sens ainsi? (HSÉ) Comment peux-tu utiliser la pensée critique pour établir des liens entre cette image et le concept à l'étude? (HSÉ) <p>Inviter l'élève à revoir l'étape de l'analyse de la situation afin de relever davantage de faits et de suppositions.</p>
Les élèves n'arrivent pas à s'entendre sur la hiérarchisation des questions.	<ul style="list-style-type: none"> Quels mots ou quels indices, dans les questions, sont importants pour déterminer les priorités? (PM) Quelles stratégies pouvez-vous utiliser? (PM) Comment pouvez-vous faire preuve d'inclusion afin de respecter les opinions des membres de l'équipe? (HSÉ) Comment pouvez-vous exprimer vos sentiments et montrer que vous comprenez les sentiments des autres? (HSÉ) <p>Permettre aux élèves d'envisager des solutions telles que les regroupements flexibles ou l'assignation de différentes tâches au sein de l'équipe.</p>
L'élève choisit une question à laquelle on ne peut pas répondre par un modèle mathématique (par exemple, « Quelles fleurs ont une odeur agréable? »).	<ul style="list-style-type: none"> Quelle question complexe liée au concept à l'étude te poses-tu en regardant cette image? (PM) À quel(s) concept(s) mathématique(s) cette question est-elle liée? (PM) Peux-tu nommer un défi lié à la question que vous avez développée? Quelle leçon tires-tu de cette erreur? (HSÉ) Comment cette situation te permet-elle de développer ta persévérance et ta résilience? (HSÉ)

ANALYSER LA SITUATION

- Montrer aux élèves l'image représentant la situation d'apprentissage, puis leur poser les questions « Que remarques-tu? » et « Quelles suppositions fais-tu? ».
- Utiliser diverses stratégies de communication telle que la stratégie Pense-Parle-Partage pour répondre aux questions.
- Inviter les élèves à noter leurs observations de façon individuelle. Par la suite, grouper les élèves en équipes et leur permettre de réagir aux questions en échangeant entre elles et eux.
- Revenir en grand groupe pour discuter au sujet des constats, des réflexions et des idées échangées lors de la conversation en équipes. Énumérer les faits pertinents et les suppositions émises quant à la mise en situation.
- Demander aux élèves de revoir les faits pertinents et les suppositions raisonnables afin de déterminer le ou les problème(s) à résoudre.
- À la suite des observations, intervenir auprès des élèves en employant une stratégie liée au besoin démontré (HSÉ).

Observation	Pistes de questionnement et interventions possibles
L'élève semble rencontrer des difficultés dans l'établissement de liens avec son expérience personnelle et participe peu à l'échange.	<ul style="list-style-type: none"> Où as-tu déjà vu cela? (PM) As-tu déjà parlé de cela dans un cours de sciences? de santé? d'études sociales? (PM) Peux-tu nommer la ou les raisons pour lesquelles tu te sens ainsi? (HSÉ) As-tu expliqué tes idées à tes partenaires, même si elles étaient différentes? (HSÉ)
L'élève établit des liens avec son expérience personnelle, mais ceux-ci ne concernent pas des concepts mathématiques.	<ul style="list-style-type: none"> À quoi ce problème te fait-il penser? (PM) Quelles sont les ressemblances ou les différences entre les liens que tu fais et les mathématiques? (PM) Où situerais-tu ton émotion sur un thermomètre d'intensité? (HSÉ) Il est important de se rappeler que l'on ne doit pas essayer de tout comprendre du premier coup. Quelle pourrait être la prochaine étape? (HSÉ)
L'élève ne parvient pas à relever des faits pertinents ou à faire des suppositions raisonnables quant au concept à l'étude.	<ul style="list-style-type: none"> Visualise le contexte à partir de l'image. Quelles informations importantes faut-il considérer avant même de déterminer un problème? (PM) Que veux-tu savoir en lien avec les mathématiques dans cette image? (PM) Es-tu en mesure d'utiliser une stratégie efficace pour gérer tes émotions tandis que tu réfléchis à cette situation? (HSÉ) Comment peux-tu utiliser la pensée créative pour établir des liens entre cette image et les mathématiques? (HSÉ)
L'élève cerne facilement un problème et démontre une impatience de passer à la prochaine étape.	<ul style="list-style-type: none"> Quelles ont été tes observations par rapport à l'image de départ? (PM) Comment te sens-tu par rapport au temps pris pour analyser la situation? (HSÉ) <p>Dans le cas où l'élève montre qu'une analyse approfondie a été effectuée et que le problème a un lien avec celle-ci, lui permettre de passer à l'étape de la compréhension du problème.</p>

CRÉER UN MODÈLE MATHÉMATIQUE

- Rappeler aux élèves de tenir compte de tous les éléments nécessaires à la résolution du problème, relevés au moment de sa compréhension, tout en sachant que certains éléments pourraient s'ajouter au cours du développement du modèle.
- Allouer aux élèves le temps requis pour travailler, réfléchir et déterminer la façon de résoudre le problème en faisant plusieurs tentatives et en utilisant différentes données, divers concepts mathématiques et diverses habiletés afin de créer un modèle.
- S'assurer que les élèves assument les rôles assignés dans l'équipe de travail afin de leur donner un sentiment d'appartenance à l'égard de la tâche.
- Observer les équipes pendant qu'elles travaillent et repérer celles qui ont des difficultés.
- Prévoir certains défis que les élèves pourraient rencontrer.

Observation	Pistes de questionnement et interventions possibles
L'élève éprouve de la difficulté à s'organiser pour créer son modèle.	<ul style="list-style-type: none"> Comment peux-tu représenter les données pour t'aider à mieux comprendre le problème? (PM) Quelles seraient les étapes à suivre afin de créer un modèle? (PM) Quelle serait une bonne stratégie pour résoudre ce problème? (PM) Comment peux-tu utiliser la pensée critique pour morceler la tâche? (HSÉ)
L'élève oublie d'inclure certains éléments importants dans son modèle.	<p>Encourager l'élève à revoir l'étape de la compréhension du problème afin de faire ressortir les éléments nécessaires à la résolution du problème. (PM)</p> <ul style="list-style-type: none"> Que dois-tu déterminer? (PM) Quelles stratégies peux-tu utiliser pour continuer l'élaboration de ton modèle? (HSÉ)
L'élève a de la difficulté à élaborer un modèle mathématique lié à sa question.	<ul style="list-style-type: none"> Il est important de se rappeler que l'on ne doit pas essayer de tout comprendre du premier coup. Quelle pourrait être la prochaine étape? (HSÉ) Quelles questions pourrais-tu poser aux membres de ton équipe afin de mieux cerner le problème à résoudre? (HSÉ et PM) <p>Fournir des exemples de modèles concrets, semi-concrets et symboliques.</p>
L'élève crée un modèle qui répond au problème en peu de temps.	<ul style="list-style-type: none"> Peux-tu passer à la prochaine étape du problème? (PM) Peux-tu utiliser un autre modèle pour répondre au problème? (PM) <p>Inviter l'élève à trouver des stratégies afin de s'adapter au rythme de travail de ses pairs. (HSÉ)</p> <p>Inviter l'élève à fournir, au besoin, un appui aux autres équipes. (HSÉ)</p>

ANALYSER ET ÉVALUER LE MODÈLE

- Amener l'élève à analyser tous les éléments de son modèle afin de s'assurer de bien répondre aux différentes parties du problème.
- Amener l'élève à évaluer son modèle.
- Recueillir tous les modèles mathématiques pour les afficher en salle de classe avec la question ciblée.
- Demander aux élèves de circuler dans la salle de classe et d'observer tous les modèles affichés.
- Demander aux élèves de formuler des commentaires et de poser des questions quant aux différents modèles, tout en s'assurant que les questions et les commentaires sont constructifs et liés à l'intention pédagogique de la situation d'apprentissage. Faire réfléchir les élèves en leur posant des questions, telles que :
 - Comparez votre travail avec celui d'autres équipes. Êtes-vous convaincues et convaincus de votre solution? Si oui, expliquez-en la raison. Si non, modifiez votre solution.
 - Quelles sont les ressemblances et les différences entre les différents modèles utilisés?
 - Y a-t-il un lien entre les différents modèles? Si oui, lesquels?

Observation	Pistes de questionnement et interventions possibles
L'élève ne parvient pas à relever les ressemblances et les différences entre les divers modèles.	<ul style="list-style-type: none"> Qu'as-tu remarqué? (PM) Quels éléments ressortent dans les modèles mathématiques? (PM) Que penses-tu de la façon dont une autre équipe a procédé? (PM) Comment peux-tu faire preuve d'inclusion afin de respecter les opinions des autres? (HSÉ)
L'élève peut comparer son modèle avec les autres et cibler des lacunes.	<ul style="list-style-type: none"> As-tu des décisions à prendre? Quelles seront les prochaines étapes? (PM) Si tu devais répondre à la question de nouveau, utiliserais-tu le même modèle? (PM) Pourquoi penses-tu que le modèle qu'a utilisé ton amie ou ami était plus efficace? (PM) Quelle leçon tires-tu de cette erreur? (HSÉ)
L'élève ne comprend pas ses erreurs ou ne réfléchit pas afin de les comprendre.	<ul style="list-style-type: none"> Est-ce que ta solution est logique? Pourquoi? (PM) Est-ce que tu peux représenter l'idée autrement? (PM) Quelles émotions ressens-tu actuellement? (HSÉ)
Le modèle de l'élève ne pourrait pas s'appliquer à la vie quotidienne.	<ul style="list-style-type: none"> Comment peux-tu utiliser la pensée critique pour déterminer la vraisemblance de ton modèle? (HSÉ) <p>Inviter l'élève à montrer son modèle à une autre personne hors de la salle de classe (par exemple, un membre du personnel, une ou un autre élève de l'école, un membre de sa famille ou de sa communauté).</p>
Le modèle ne fournit pas une solution adéquate au problème.	<p>Encourager l'élève à revoir une étape précédente et à redéfinir le problème, à faire de nouvelles suppositions ou à apporter des changements au modèle.</p>