



Faire la différence...

De la recherche à la pratique

Monographie de recherche n° 69
Avril 2017

Les bienfaits du codage

Au cœur de la pensée informatique, et des mathématiques, il y a l'abstraction. Lorsque des enfants écrivent du code, ils en viennent à...

1. comprendre de façon concrète les abstractions qui figurent au cœur des mathématiques;
2. modéliser de manière dynamique les concepts et les relations mathématiques;
3. gagner confiance dans leurs propres compétences et leur capacité d'agir en tant qu'apprenants en mathématiques.

Du codage informatique dans le curriculum de mathématiques de la maternelle à la 8^e année?

Par George Gadanidis, Université Western; en collaboration avec Iain Brodie, Leslee Minniti et Bronna Silver, St. Andrews Public School, Toronto District School Board

La tendance à ajouter une forme quelconque de codage informatique aux programmes d'études est un phénomène international. Comment intégrer précisément le codage informatique au curriculum? Devrait-il s'agir d'une matière en soi? Faudrait-il l'intégrer à d'autres matières?

Le codage informatique suscite tout un engouement. Le premier ministre Justin Trudeau a récemment déclaré que nous devons faire un bien meilleur travail à amener les jeunes à comprendre ce qu'est le codage et en quoi c'est important, comment écrire des lignes de code, comment résoudre des problèmes, comment créer l'algorithme le plus élégant possible¹.

La Colombie-Britannique a récemment annoncé que le codage informatique serait ajouté au curriculum à tous les niveaux, de la maternelle à la 12^e année, et la Nouvelle-Écosse a fait une annonce semblable. La tendance à ajouter une forme quelconque du codage informatique aux programmes d'études est un phénomène international; en 2014, l'Angleterre a rendu le programme de codage obligatoire pour tous les élèves de la maternelle à la 12^e année.

Où l'intégrer dans le curriculum?

Actuellement, le codage informatique est conçu davantage comme un domaine du curriculum en lui-même que comme une façon de réfléchir susceptible d'améliorer les matières existantes. Jeannette Wing suggère qu'à la lecture, à l'écriture et à l'arithmétique, nous devrions ajouter la pensée informatique à la capacité d'analyse de tout enfant² (p. 33).

Cet intérêt pour le codage informatique en éducation n'a rien de nouveau. Il s'agissait d'une importante composante des travaux réalisés par Seymour Papert avec Logo, un environnement de programmation qui invite les enfants à écrire du code pour faire déplacer une tortue à l'écran. Papert voyait Logo, à la fois comme un environnement de codage et un environnement d'apprentissage des mathématiques. Il a écrit que Logo est à l'apprentissage des mathématiques ce que le fait de vivre en France est à l'apprentissage du français³ (p. 6).

D'autres chercheurs soulignent le fait que l'intégration du codage à d'autres matières, surtout aux mathématiques, crée des possibilités sur le plan pédagogique. Une récente revue de l'état de la pensée informatique chez les élèves de la maternelle à la 12^e année a toutefois conclu que l'idée de se servir de l'informatique comme moyen d'enseigner d'autres matières est sous étudiée⁴ (p. 42).

Implications pour l'enseignement des mathématiques

Ci-dessous, nous décrivons trois moyens importants par lesquels le recours au codage dans l'enseignement et l'apprentissage des mathématiques peut favoriser le développement conceptuel de l'élève.

1. La concrétisation de l'abstraction

Au cœur de la pensée informatique, et des mathématiques, il y a l'abstraction^{5,6}. Nous nous en servons naturellement dès le plus jeune âge tandis que nous faisons l'acquisition du langage. Par exemple, quand nous en venons à comprendre le mot chat, nous ne tenons pas compte de toutes les différences qui existent entre les chats et nous créons un modèle abstrait des caractéristiques essentielles du chat.

Voyons comment l'abstraction entre en jeu quand nous utilisons le logiciel Scratch, un environnement de codage basé sur des blocs (voir scratch.mit.edu) pour dessiner un carré. Cet outil a été élaboré en 2004 au Massachusetts Institute of Technology (MIT) et sa conception est basée sur Logo et sur les travaux de Papert⁷.

Nous pourrions créer du code pour dessiner un carré en nous servant des instructions de la Figure n° 1. (Le code correspondant aux exemples Scratch que nous avons utilisés est disponible au lien suivant scratch.mit.edu/. Vous pouvez vous en servir, le modifier, l'essayer et le faire connaître à d'autres.) Nous pourrions également accomplir cette tâche en définissant dessiner un carré comme ayant son propre bloc de code et nous en servir au besoin (voir la Figure n° 2). La création d'un nouveau bloc de code pour remplacer un groupe existant est particulièrement utile lorsque nous écrivons un code plus complexe.

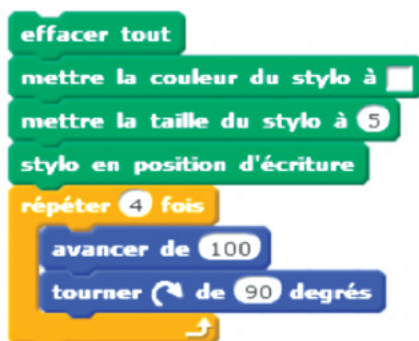


Figure n° 1. Dessiner un carré avec Scratch

Le bloc de code dessiner un carré aide à simplifier le code. Il facilite également l'utilisation de dessiner un carré comme objet d'un autre code. Par exemple, le code de la Figure n° 3 permet de dessiner un carré, de tourner de 36° et de répéter l'opération 10 fois; en changeant au hasard la couleur du stylo avant de dessiner chaque carré, chacun des carrés obtenus devient plus facilement visible.

Que s'est-il passé ici d'un point de vue mathématique?

Premièrement, le concept de carré a été réduit à ses éléments essentiels : avancer de 100 pas, tourner de 90°, répéter 4 fois. Deuxièmement, l'abstraction à laquelle nous avons abouti est conceptuellement riche, car elle représente tous les carrés, de la même façon que le mot chat représentait tous les chats. Nous pouvons modifier le code pour dessiner des carrés de différentes tailles, de différentes orientations et de différentes couleurs. Troisièmement, cette version abstraite du carré est concrète⁸. Elle a été transformée en un bloc de code qui peut être déplacé, manipulé et sur lequel d'autres blocs de code peuvent agir.

N'est-il pas intéressant de voir qu'avec le codage informatique, nous pouvons rendre abstraits des concepts mathématiques et en même temps les concrétiser? Ainsi concrétisées, ces abstractions aident les élèves dans leur compréhension conceptuelle.

Cette concrétisation peut être approfondie lorsque nous utilisons le codage pour enseigner à des robots programmables l'exécution de tâches mathématiques. C'est ainsi que des enfants peuvent écrire un code semblable pour demander à Sphero de s'avancer le long d'un carré. Pour le décrire simplement, disons que Sphero est un robot sur roues emprisonné dans une coquille sphérique. Il se déplace à la façon du hamster sur sa roue (voir sphero.com) et peut être codé à l'aide d'un téléphone intelligent ou d'une application pour tablette électronique.

De telles expériences aident à traiter des attentes du curriculum de l'Ontario tant en géométrie et sens de l'espace (l'élève « décrit, trie, classe, classifie, construit et compare des figures planes et des solides et décrit la position et le déplacement des objets par l'exploration¹⁰. » (p. 134) au programme de la maternelle et du jardin d'enfants; l'élève peut « identifier et construire



Figure n° 2. L'ajout d'un bloc de code Dessiner un carré

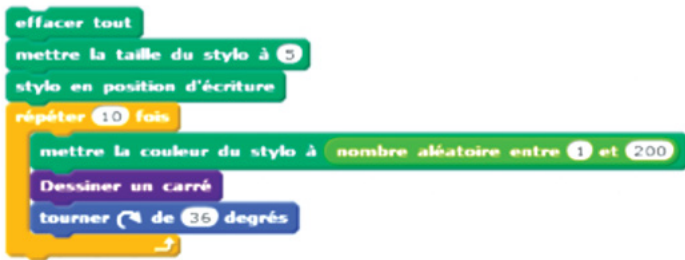


Figure n° 3. Dessiner un motif en faisant pivoter un carré

diverses figures planes et divers solides afin de développer une compréhension de leurs propriétés⁹. » (p. 30) en 1^{re} année; et l'élève peut « comparer et décrire diverses figures planes et divers solides afin de développer une compréhension de leurs propriétés⁹. » (p. 38) en 2^e année qu'en modélisation et algèbre (l'élève « identifie, explore, décrit, compare, crée, prolonge et représente des régularités de différentes façons en cherchant le motif d'une suite et en prédisant les termes suivants de la suite¹⁰. » (p. 134) au programme de la maternelle et du jardin d'enfants; et peut identifier, produire ou effectuer des suites non numériques entre la 1^{re} et la 3^e année^{9, 10}).

2. Automatisation et modélisation dynamique

Jeannette Wing affirme que l'informatique est l'automatisation de nos abstractions⁵ (p. 3718). Par exemple, le bloc de code Scratch dessiner un carré automatise la tâche qui consiste à dessiner un carré. Nous créons le code une fois et nous pouvons l'utiliser pour dessiner autant de carrés que nous le voulons. L'abstraction et l'automatisation offrent aux élèves la possibilité de modéliser les concepts mathématiques de façon dynamique.

La modélisation dynamique permet aux élèves d'explorer les relations, de poser des questions hypothétiques et de communiquer leurs apprentissages et leur savoir à leurs camarades, à leur famille et à leurs amis.

3. Point d'entrée accessible, niveau de complexité élevé³, et responsabilité de l'élève à l'égard de son apprentissage

Les environnements de codage disponibles aujourd'hui, en particulier ceux basés sur des blocs, offrent un point d'entrée accessible, permettant une participation avec des connaissances préalables minimales, et l'atteinte d'un niveau de complexité élevé, offrant des possibilités d'étendre l'apprentissage à des concepts plus complexes et des représentations plus variées, comme c'est le cas avec Logo de Papert.

Un environnement dans lequel le point d'entrée est accessible pour aller jusqu'à un niveau de complexité élevé favorise l'apprentissage différencié. Les élèves peuvent s'engager selon leur niveau d'aptitude et d'aisance et ainsi élargir leur apprentissage à mesure qu'ils acquièrent une compréhension conceptuelle et gagnent confiance en eux.



Le codage, dans un milieu où il existe un point d'entrée accessible et un niveau de complexité élevée, encourage également l'élève à faire preuve de responsabilité et lui permet de s'approprier son apprentissage. Les élèves qui écrivent un code pour modéliser un motif ou une relation ont une

certaine maîtrise. Il y a de nombreuses façons de résoudre un problème à l'aide du codage et les élèves peuvent utiliser les méthodes qui pour eux ont un sens. Ils peuvent aussi chercher à résoudre des problèmes connexes.

Commentaires d'enseignants

Il y a quelques années, les trois coauteurs de cette monographie (Iain Brodie, Leslee Minniti et Bronna Silver) ont essayé pour la première fois d'utiliser le codage dans leur enseignement des mathématiques. Leurs commentaires révèlent la façon dont ils s'y sont pris et ce qu'ils ont vécu :

- « Les élèves de 7^e et 8^e année ont appris à dessiner un carré dans Scratch, de façon à pouvoir ensuite l'enseigner aux élèves de la 1^{re} à la 3^e année. Ils se sont également mis au défi de dessiner d'autres formes. Le mot qui a été le plus important pour les élèves est sans doute *curiosité*. »
- « Les élèves de la 1^{re} à la 3^e année ont été mis au défi d'enseigner à leur enseignant sur la façon de "se déplacer le long d'un carré" (l'enseignant faisant délibérément des erreurs lorsque les instructions n'étaient pas suffisamment claires). »
- « Dans la salle de classe de 1^{re} année, c'était un plaisir de voir les élèves plus âgés avec les plus jeunes, ceux-ci plaçant leurs petites mains juste au-dessus du clavier, et les abaissant parfois, et de voir comment les plus grands étaient si beaux à leur dire : "Non, tu utilises tes mains, tu vas y arriver". »
- « C'était formidable. Je voyais des enfants qui disaient : "Qu'est-ce qui se passe si...?" Il arrivait bien souvent que ce soit mes élèves de 2^e et 3^e année qui enseignaient des choses aux plus âgés. »
- « Je n'ai jamais entendu l'un de mes élèves dire "comment tu épelles cela?" Il n'y avait pas cette inquiétude. Ce programme ne les contraint pas à le faire. C'était vraiment une expérience très réjouissante. »
- « Ce que j'ai vraiment aimé, c'était le lien que nous avons établi avec les familles à la maison. Nous avons dit aux élèves : "Ce projet est maintenant le vôtre et vous devez partager ce que vous avez appris". Ils étaient sceptiques à l'idée que leurs parents puissent le faire, mais ils sont revenus pleins d'enthousiasme raconter ce qu'ils avaient fait avec eux. »

En résumé : Allez-y! Essayez!

Il existe plusieurs ressources disponibles pour intégrer le codage informatique à la salle de classe. Ces dernières années, nous avons élaboré du matériel pédagogique pour apporter un soutien aux enseignants qui utilisent la pensée informatique

dans l'enseignement et l'apprentissage des mathématiques. En voici une liste annotée.

- **Heure de CODE** (<https://code.org/>) offre des activités de codage d'une heure qui donnent lieu à des événements communautaires sur l'apprentissage plus poussé de la pensée informatique. Sur la page Web, on y trouve notamment des renseignements détaillés sur les activités qui peuvent intéresser les élèves (y compris des activités sans exigences technologiques), des suggestions au personnel enseignant souhaitant être impliqué et des certificats imprimables.
- **La communauté de pratique pour la pensée informatique du Réseau de connaissances en mathématiques de l'Ontario** (mathnetwork.ca/ct) offre des documentaires sur des leçons modèles filmées dans des salles de classe ontariennes ainsi que d'autres ressources. Financé par le ministère de l'Éducation de l'Ontario (un projet du Réseau d'échange des connaissances pour la recherche appliquée en éducation (RECRAE) présenté par le Fields Institute for Research in Mathematical Sciences).
- **Math + Code » Zine** (researchideas.ca/mc) est une publication trimestrielle qui offre un soutien au codage dans divers contextes d'enseignement des mathématiques. Ce magazine

bénéficie de l'appui du centre de soutien pédagogique de l'Université Western et du Fields Institute (en anglais seulement).

- **La pensée informatique dans l'enseignement des mathématiques** (ctmath.ca), est un projet de recherche mené en partenariat par sept universités et le Fields Centre for Mathematics Education. Il vise à explorer l'utilisation de la pensée informatique dans l'enseignement des mathématiques de la maternelle au premier cycle d'études universitaires. Des projets, des rapports, des documentaires et des plans de cours seront affichés sur le site Web du projet à mesure de leur réalisation.

N'hésitez pas à intégrer le codage informatique à votre enseignement. Vous et vos élèves en raffolerez! Comme l'a fait remarquer un des coauteurs durant le cours : « J'ai été très enthousiasmé de voir que toutes les personnes présentes dans la salle de classe de 1^{re} année réussissaient, s'estimaient capables et compétentes, et étaient d'enthousiastes mathématiciennes et mathématiciens. Pas seulement les élèves, mais aussi les enseignants. »

Bibliographie

1. SNELL, Laurie. "Google opens its doors on Breithaupt; PM Trudeau takes part", *KITCHENER POST*, 20 janvier 2016. En ligne : <http://www.kitchenerpost.ca/news-story/6240341-google-opens-its-doors-on-breithaupt-pmtrudeau-takes-part/>
2. WING, J. M. (2006). "Computational thinking", *Communications of the ACM*, vol. 49, n° 3, p. 33–35.
3. PAPERT, S. (1980). *Mindstorms: Children, computers, and powerful ideas*, New York: Basic Books.
4. GROVER, S. et R. PEA (2013). "Computational thinking in K–12: A review of the state of the fields", *Educational Researcher*, vol. 42, n° 1, p. 38–43.
5. WING, J. M. (2008). "Computational thinking and thinking about computing", *Philosophical Transactions of the Royal Society A*, vol. 366, n° 1881, p. 3717–3725.
6. AHO, A.V. (2012) "Computation and computational thinking", *Computer Journal*, vol. 55, p. 832–835.
7. LOGO HISTORY. [En ligne], 2017. [http://el.media.mit.edu/logo-foundation/what_is_logo/history.html]
8. GADANIDIS, G. "Five affordances of computational thinking to support elementary mathematics education", *Journal of Computers in Mathematics and Science Teaching*, vol. 36, n° 2, p. 143-151. [À paraître en 2017].
9. ONTARIO. Ministère de l'Éducation (2005). *Le curriculum de l'Ontario de la 1^{re} à la 8^e année, Révisé*, Imprimeur de la Reine pour l'Ontario, Toronto.
10. ONTARIO. Ministère de l'Éducation (2016). *Programme de la maternelle et du jardin d'enfants*, Imprimeur de la Reine pour l'Ontario, Toronto.

La série de monographies *Faire la différence... De la recherche à la pratique* est produite en collaboration par l'Ontario Association of Deans of Education et la Division du rendement des élèves du ministère de l'Éducation de l'Ontario.

Pour en savoir plus sur la façon de rédiger une monographie, **cliquez ici** : [Mobiliser les résultats de la recherche pour les appliquer de façon significative](#), par Michelann Parr, Ph. D., et Terry Campbell, Ph. D., co-rédactrices.

La série *Faire la différence* est mise à jour et publiée à www.edu.gov.on.ca/fr/literacynumeracy/inspire/research/WhatWorks.html.

Les opinions et les conclusions exprimées dans ces monographies sont celles des auteurs; elles ne reflètent pas nécessairement les politiques, les opinions et les orientations du ministère de l'Éducation de l'Ontario ou de la Division du rendement des élèves.

ISSN 1913-1097 *Faire la différence... De la recherche à la pratique* (imprimé)

ISSN 1913-1100 *Faire la différence... De la recherche à la pratique* (en ligne)