



Analyse du processus de construction de connaissances dans des activités de programmation à l'école

Margarida Romero¹  · Lucie DeBlois² 

Accepted: 25 May 2022 / Published online: 8 July 2022
© The Author(s) 2022

Abstract The introduction of programming activities in the classroom has given rise to a variety of teaching methods. Some methods focus on learning to code while others take a creative programming approach (Resnick & Rusk, 2020). This study looks at creative programming as an opportunity for students to develop their reasoning skills. Drawing from both mathematics teaching and creative programming practices, we identify the elements of informatics thinking and use them to analyze the behaviour of students during a programming task in educational robotics. We also apply certain conceptual tools from both the field of mathematics teaching and the framework of informatics thinking in the course of our study. Creative programming practices provide six factors for analyzing how the students approached situational problems, framed the problems, learned the code, and developed the programs. From a didactics perspective, we use the components of DeBlois' model (2001, 2003) for interpreting cognitive activities to analyze elements in the knowledge-building process. Our article ends with a comparison of the benefits of each analytical framework, which could have implications for teacher training on programming in the classroom.

Résumé L'introduction d'activités de programmation à l'école a donné lieu à une diversité d'approches d'enseignement. Ces approches sont parfois centrées sur un apprentissage du code et d'autres fois sur une démarche de programmation créative (Resnick & Rusk, 2020). Cette étude analyse une démarche de programmation créative du point de vue des occasions de développement du raisonnement qu'elle offre aux élèves. Nous analysons les activités des élèves devant une tâche de programmation en robotique pédagogique. Pour réaliser l'analyse des activités, nous identifions les composantes de la pensée informatique à partir d'une double articulation: la didactique des mathématiques et les pratiques de program-

✉ Margarida Romero
margarida.romero@univ-cotedazur.fr

Lucie DeBlois
lucie.deblois@fse.ulaval.ca

¹ Laboratoire d'Innovation et Numérique pour l'Éducation (LINE), Université Côte d'Azur, Nice, France

² Centre de Recherche et d'intervention sur la Réussite Scolaire (CRIRES), Université Laval, Québec, Canada

mation créative. Les pratiques de programmation créative ont permis de considérer six composantes pour étudier l'examen d'une situation-problème par les élèves, le cadrage du problème, l'apprentissage du code et la réalisation d'un programme. Du point de vue didactique, les composantes du modèle d'interprétation des activités cognitives de DeBlois (2001, 2003) ont permis d'analyser les composantes du processus de construction de connaissances. Nous terminons cet article en comparant l'apport de chacun de ces cadres d'analyse, ce qui pourrait avoir des retombées pour la formation à l'enseignement de la programmation à l'école.

Mots-clés Apprentissage collaboratif · Programmation · Robotique pédagogique · Créativité · Raisonnement · Résolution de situations-problèmes

Introduction

Depuis les travaux pionniers de Papert (1980), un nombre croissant d'enseignants ont introduit l'apprentissage de la programmation à l'école. Des pays comme la France et le Royaume-Uni ont intégré l'apprentissage de la programmation dans le programme en mathématiques tandis que la discussion sur la place de la programmation dans les programmes se poursuit au Québec (Barma et al., 2015). Certaines initiatives abordent la programmation comme un système formel de codes à partir duquel il est nécessaire de développer un apprentissage algorithmique. D'autres initiatives s'appuient sur les opportunités offertes par la programmation pour mobiliser différentes stratégies de la pensée informatique sous une approche davantage pluridisciplinaire (Lye & Koh, 2014). La pensée informatique correspond à un ensemble de stratégies liées à la modélisation de connaissances et de processus, à l'abstraction, à l'algorithmique et à l'identification, à la décomposition, l'organisation de structures complexes et de suites logiques. Ainsi, la pensée informatique n'est pas uniquement une pensée algorithmique mais une démarche qui engage les humains à réfléchir sur la résolution de situations-problèmes à partir de ce que Wing (2006) appelle des « outils mentaux ».

Ce dernier type d'initiatives ne se limite pas à l'apprentissage d'un système formel. Il engage l'apprenant dans la résolution de situations-problèmes, qui est abordée dans le cadre de la compétence mathématique du programme de formation de l'école québécoise (MELS, 2006). Nous faisons état de la différence entre un problème et une situation-problème. Cette dernière ne peut pas être résolue seulement sur la base de connaissance apprises antérieurement mais nécessite que l'élève s'engage dans une démarche de recherche et raisonnement (Lajoie & Bednarz, 2016). La situation-problème oblige ainsi l'élève « à faire appel à une combinaison nouvelle de concepts et de processus appris antérieurement » (MELS, 2006, p. 240). La résolution d'une situation-problème invite l'apprenant à résoudre des situations nouvelles. Le développement d'un processus original est nécessaire pour trouver une solution plausible. La créativité engagée dans cette démarche de résolution de situations-problèmes doit avoir une valeur dans le cadre d'une communauté de référence (Romero et al., 2012; Runco & Jaeger, 2012). Précisons que la créativité est considérée par Besançon et al. (2011). Elle combine à la fois des phases de pensée divergente ou exploratoire pour la génération de nouvelles idées et des phases de pensée convergente ou intégrative pour l'élaboration des solutions intermédiaires les plus appropriées pour la situation-problème. Cette définition est retenue dans le contexte de notre étude sur la pensée informatique.

Trois approches émergent dans la manière de concevoir la pensée informatique et les liens qu'elle entretient, ou non, avec la résolution de problèmes et la résolution de situations-problèmes. Une première envisage la pensée informatique comme un langage formel de type universel indépendant d'autres disciplines (Abiteboul & Dowek, 2020) sans établir de relations explicites entre la résolution de problème et l'informatique. Certains auteurs préfèrent l'usage de l'expression « pensée algorithmique », qui restreint le périmètre à « un mode de pensée logique et organisé utilisé pour décomposer un objectif

compliqué en une série d'étapes (ordonnées) en utilisant les outils disponibles» (Lockwood et al., 2016, p. 1591). Cette première approche correspond à la notion de problèmes pour laquelle il est possible d'appliquer un ensemble de connaissances apprises sans besoin de s'engager dans une démarche de recherche et de raisonnement. Une deuxième approche considère que la pensée informatique contribue au développement de la compétence à résoudre des situations-problèmes en contexte disciplinaire mais aussi pluridisciplinaire (Riley & Hunt, 2014) puisque l'élève doit s'engager dans une démarche de recherche et de raisonnement. Enfin, une troisième approche considère que l'enseignement de la pensée informatique doit pouvoir se réaliser à partir de la résolution de situations-problèmes (Román-González et al., 2017; Wing 2006, 2011). Nous situons notre étude sous cette troisième approche, considérant que l'enseignement de la pensée informatique doit permettre d'engager les élèves à résoudre des situations-problèmes pour lesquelles il faut mobiliser des connaissances informatiques (Wing, 2006, 2011).

Nous avons voulu étudier la nature des activités d'élèves dans une situation-problème de robotique pédagogique afin de comprendre comment les élèves s'engagent dans la programmation créative et comment ils développent une solution (Jonassen, 2003; Jonassen et al., 2005). L'engagement des élèves dans un processus de programmation nécessite qu'ils modélisent la situation-problème. La modélisation est une démarche qui nécessite à la fois la capacité d'interpréter, de réduire et de représenter les informations pertinentes de la situation-problème, selon différents niveaux d'abstraction et de codification. La démarche de modélisation n'est cependant pas toujours observée dans les activités de programmation créative (Chevalier et al., 2020).

Dans cet article, nous nous intéressons à la programmation créative médiatisée par un outil de programmation. Parmi les situations-problèmes de programmation créative qui visent le développement de la pensée informatique, nous avons considéré les défis liés à la programmation de robots pédagogiques. Ce défi a engagé les élèves en équipe et en interdépendance, comme le définit Mondada et al. (2016). Dans ce contexte, la solution à la situation-problème est construite par le biais de l'élaboration d'un programme informatique. En somme, cet article présente une activité de résolution créative de situations-problèmes au cours de laquelle les élèves génèrent de nouvelles idées et posent un jugement sur ces dernières pour trouver une solution nouvelle. Nous étudions comment les élèves combinent des phases d'exploration et des phases d'intégration au cours du processus de création d'un programme informatique.

Un cadre en pensée informatique pour analyser une démarche de programmation créative

Nous situons notre travail dans une approche socioconstructiviste de l'apprentissage (Vygotski, 1985) avec une orientation constructionniste (Kafai & Resnick, 1996; Papert & Harel, 1991) en relation avec la perspective de la pensée *design* (Bjögvinsson et al., 2012; Owen, 2007). Papert et Harel (1991) situent le constructionnisme comme une approche constructiviste dans laquelle l'apprentissage se fait dans un contexte de création. Nous sommes dès lors en mesure de faire une place non seulement aux comportements d'exploration des élèves, mais aussi aux erreurs qui surgissent lors des expérimentations, permettant ainsi les débogages (Papert, 1980). En effet, la résolution des bogues émerge de l'exploration des élèves et fait intervenir non seulement leur système de connaissances et les contraintes de la tâche mais aussi les caractéristiques du système informatique.

À partir de ces caractéristiques, les élèves mettent à l'essai une solution qui peut prendre différentes formes selon la marge créative de l'activité. La marge créative d'une activité se manifeste tant au cours du processus que de la réalisation d'une solution plausible. Lorsque les élèves arrivent à développer des hypothèses à partir de leurs connaissances, ils s'engagent dans des essais contrôlés, visant à mettre en œuvre des hypothèses et à vérifier des connaissances pouvant les aider dans la résolution de la tâche.

La créativité peut ainsi être observée tant sur le processus de diagnostic, d'analyse et de correction d'essais-erreurs que par le biais de la proposition d'une nouvelle solution.

Plus précisément, nous avons identifié six composantes de la pensée informatique (Romero et al., 2017) à partir des travaux de Wing (2006, 2011) même si ses travaux n'évoquent pas la notion de situation-problème. Ainsi, nous utilisons le concept de pensée informatique défini par Wing (2006) pour considérer que la pensée informatique permet à la fois une approche pour résoudre des situations-problèmes, pour concevoir des systèmes et pour comprendre le comportement humain. Quatre des six composantes identifiées sont mises en relation avec les composantes du référentiel pour la résolution collaborative de problèmes de PISA (OECD, 2013). Nous avons choisi de nous référer au référentiel PISA en raison de sa nature internationale et du potentiel d'analyse qu'il offre dans le cadre d'activités de programmation collaboratives engageant de manière simultanée plusieurs équipes dans différents pays. Enfin, le référentiel PISA est un des cadres les plus utilisés dans les recherches récentes sur la résolution collaborative de problèmes en sciences de l'éducation (Care et al., 2016; Herborn et al., 2017). PISA ne fait pas la distinction entre problème et situation-problème, mais nous allons considérer dans cette étude la résolution de situations-problèmes dans un but de développement de la pensée informatique.

Développement du cadre théorique de la pensée informatique

Rappelons que certaines activités de programmation n'engagent pas les élèves dans des démarches de résolution de problèmes car elles consistent en une suite d'étapes prédéterminées, comme certains tutoriels de Code.org. Il s'agit dans ce cas d'activités de programmation guidée et non créative. Dans ce contexte, les élèves sont confrontés à des problèmes auxquels ils peuvent appliquer des connaissances acquises pour trouver une solution. D'autres activités de programmation permettent aux élèves de disposer d'une marge créative pour analyser et développer une solution originale par le biais d'un programme informatique de manière débranchée (*unplugged computing*), sur écran ou sur des artefacts programmables comme des robots pédagogiques. Par exemple, les élèves peuvent être invités à créer un épisode de l'histoire de leur ville pour une certaine période historique avec un outil de programmation visuelle comme Scratch. Ainsi, nous distinguons l'apprentissage de la programmation guidée, qui engage les élèves dans une démarche par étapes dans laquelle l'apprentissage du code est un objectif en soi, de l'apprentissage par des activités de programmation créative qui les engagent à relever un défi comme imaginer, concevoir et créer un artefact informatique (Kafai & Resnick, 1996; Resnick & Siegel, 2015; Romero, 2016). Le type de supports technologiques utilisé influencera la modélisation de la situation-problème, comme l'illustrent les six composantes de cette pensée informatique identifiées dans la Fig. 1.

De façon spécifique, la composante 1 est liée à l'interprétation des constituantes de la situation-problème et aux relations entre elles. Pour des auteurs comme Kramer (2007), cette composante est liée à l'identification des constituantes de la situation-problème, compte tenu de la nécessaire réduction d'informations. La composante 2 correspond à la modélisation des données où la représentation de ces données et de leurs relations est personnelle. La composante 3 correspond à une recherche de sens à l'égard du type de formalisation par le biais du code et par conséquent, par la sélection d'un registre de représentations (Duval, 2006) ou d'une variété de langages qui serviront à communiquer. La composante 4 correspond à la sélection du système technologique (logiciel/matériel/ordinateur/robot) qui donne un support électronique et physique à la conception de la solution informatique. La composante 5 se concentre sur l'écriture du programme informatique conçue (programmation), une formalisation des représentations construites pendant la composante 2. Enfin, la composante 6 correspond au processus itératif et cyclique d'amélioration du programme et de ses différentes versions intermédiaires, amélioration qui gagne à être réalisée dans une approche de conception collaborative comme celle décrite par Voogt et collègues (2015).

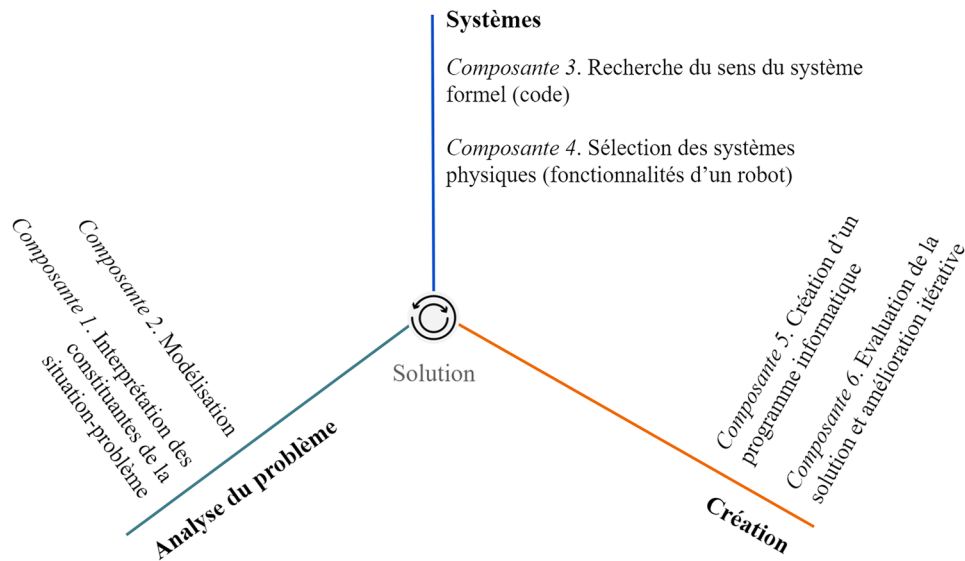


Fig. 1 Les six composantes de la pensée informatique de (Romero et al., 2017)

Rappelons que les six composantes peuvent se dérouler selon une approche cyclique et itérative plutôt que selon une démarche linéaire et unique. Un processus de corégulation des activités entre les élèves émergera durant le processus de réalisation de la solution informatique. Cette démarche de programmation s'inscrit dans les approches de résolution collaborative de situations-problèmes en apprentissage (Kapur et al., 2005). Nous combinerons deux cadres d'analyse. Le premier permet d'analyser les composantes de la pensée informatique pour comprendre le processus temporel entre ces différentes composantes; le deuxième, issu de la didactique des mathématiques, permet d'interpréter les activités cognitives des élèves.

Un cadre issu de la didactique des mathématiques pour analyser une démarche de programmation

Les études de Lithner (2008) distinguent la résolution créative de situations-problèmes, d'une part, de l'application d'heuristiques pour la résolution de problèmes, d'autre part. Cette dernière concerne davantage la résolution de problèmes de type imitative ou algorithmique. La première, la résolution créative de situations-problèmes, donnerait lieu à de meilleurs apprentissages en mathématique du fait de l'appropriation de la situation-problème et de la mobilisation des connaissances pour l'élaboration d'une stratégie pour le problème. C'est ainsi que le modèle d'interprétation des activités cognitives des élèves élaboré par DeBlois (2003), approfondi dans les travaux de Bélanger, DeBlois et Frieman (2014), permet d'observer la créativité manifestée par les élèves dans une tâche de résolution de situations-problèmes mathématiques. En effet, ce modèle montre que la créativité se manifeste notamment par la variété de relations, appelées coordinations, réalisées par les élèves entre les attentes perçues par ces derniers et les représentations qu'ils se donnent d'une situation-problème. Ces coordinations orientent la sélection des procédures expérimentées par les élèves pour structurer peu à peu une organisation faisant intervenir tant le système de connaissances des élèves que les contraintes de la situation-problème. La Fig. 2 illustre ce modèle d'interprétation.

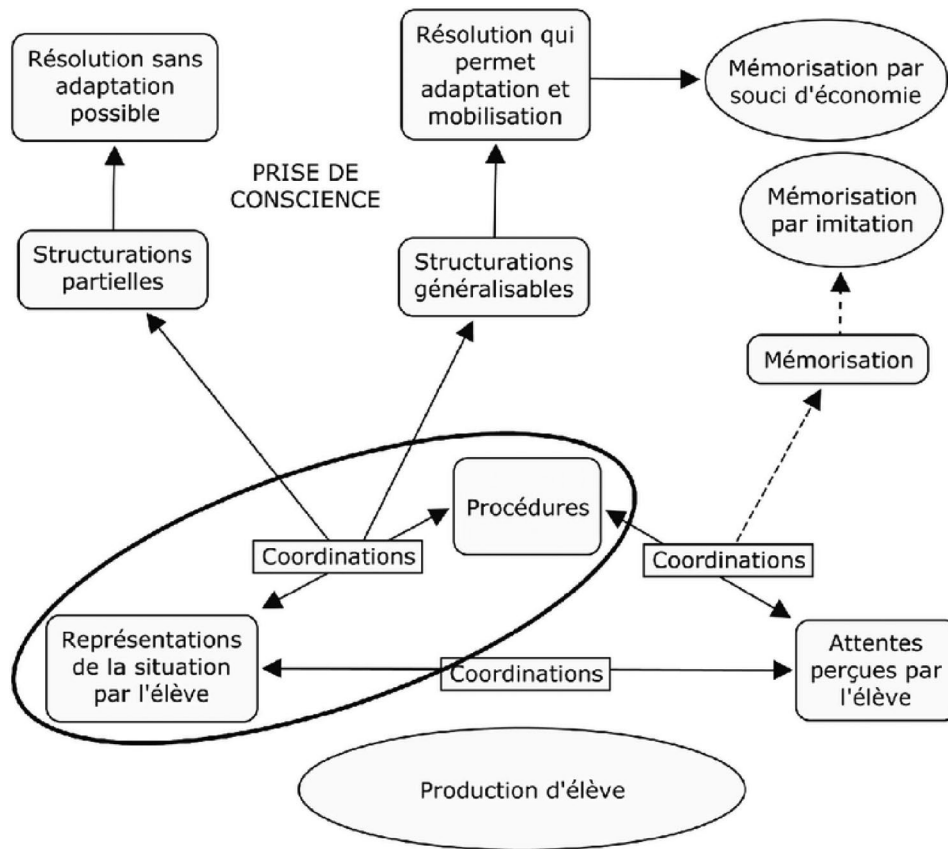


Fig. 2 Modèle d'interprétation des activités cognitives des élèves (Bélangier et al., 2014)

Une activité de programmation, qui engage les élèves à penser à différentes options durant le processus et à élaborer une solution, leur permettrait donc de manifester leur créativité. En effet, la modélisation des éléments du programme informatique qui est engagée dans l'appropriation de la situation-problème conduit les élèves à repérer les caractéristiques de la situation-problème et de l'environnement dans lequel ils résolvent cette situation-problème afin de structurer les informations. Le fait de repérer et de structurer les informations conduit les élèves à évoquer des connaissances, que nous appellerons des représentations mentales, dans le respect du contrat didactique de la classe. C'est ainsi qu'émergent les attentes que les élèves entretiennent à l'égard de la résolution de la situation-problème, par exemple, une attente liée au fait de pouvoir déplacer le robot dans le cadre d'une tâche avec des robots de sol (Komis et al., 2018). Nous considérons également les attentes à l'égard de ce qu'ils croient que les enseignants attendent d'eux. Par exemple, les élèves peuvent envisager que leur enseignant s'attend à qu'ils prennent des notes pendant l'activité. Dans ce contexte, nous appelons coordinations, les relations entre les représentations mentales que les élèves évoquent au contact de la situation-problème et les attentes qu'ils entretiennent dans le contexte de cette situation-problème. Ces coordinations conduiront les élèves à réaliser des expérimentations qui se manifestent par la sélection d'un ensemble de procédures. Les expérimentations des élèves au cours de l'activité sont considérées comme une activité créative. Ainsi, elles correspondent à autant de régulations favorisant des ajustements pour arriver à une solution qu'ils jugeront plausible. Ces expérimentations contribueront à développer de nouvelles représentations du phénomène à l'étude dans le cadre de la situation-problème et permettront l'observation des résultats

obtenus. Le fait que les élèves soient en mesure de repérer les conséquences de leurs décisions permettra de réguler leur avancement vers une solution potentielle.

Notre interprétation prendra donc comme point de départ la situation-problème proposée pour analyser les représentations mentales des élèves mais aussi les attentes qu'ils entretiennent tant à l'égard d'eux-mêmes vis-à-vis de la tâche que de celles qu'ils anticipent de leurs enseignants. Le repérage des coordinations entre leurs représentations mentales et leurs attentes permettra de comprendre le sens des procédures, que nous appelons expérimentations, mises en œuvre en tant que processus créatif pendant la résolution de situations-problèmes. A la suite des expérimentations, les élèves peuvent observer des résultats. Dans le cas où les élèves réalisent des prises de conscience à l'égard des raisons ayant donné lieu à certains résultats, ces raisons favoriseraient un apprentissage qui peut être mobilisé et adapté à d'autres situations.

Interpréter une démarche de programmation par l'articulation des deux cadres théoriques

Une situation-problème d'une certaine complexité pourrait conduire à trois types de traitement de l'information au cours du déroulement de l'activité. D'un côté, certaines informations seront écartées par un processus de priorisation et de réduction des informations. La réduction d'information est habituellement peu considérée dans l'apprentissage de la programmation de manière explicite. Afin de réduire certaines ambiguïtés de la situation-problème, certains enseignants réalisent un modelage permettant de réduire certaines informations et clarifier la situation-problème. Toutefois, ce modelage¹ diminue les opportunités de créativité des modélisations réalisées puisqu'il oriente la démarche des élèves. Pour que le processus créatif puisse se développer pleinement, les degrés de liberté sur la génération d'idées (pensée divergente-exploratoire) et sur la sélection d'idées (pensée convergente-intégrative) doivent alterner au cours du processus de résolution créative de situations-problèmes. En effet, la réduction émergerait de la sensibilité des élèves à l'égard de certaines caractéristiques de la situation-problème.

L'étude de cette sensibilité nous semble fondamentale pour cerner le processus des élèves qui répond à des critères de pertinence et de créativité. Un niveau de complexité suffisant de la situation-problème est donc nécessaire afin d'étudier l'enjeu de la réduction d'informations. La modélisation des élèves conduit à une explicitation consciente, sensible au contexte et aux personnes. D'un point de vue didactique, la mise en œuvre d'un ou de plusieurs registres de représentation (Bélanger et al., 2014) permet de réduire les informations. Suivant une approche de pensée *design*, les informations traitées au cours de la situation-problème apportent une reconfiguration de la représentation de celle-ci. Ainsi, certaines informations sur le déroulement de l'activité pourront servir à établir des nouvelles procédures de résolution de la situation problème et conduire à des actions opérationnelles concrètes au moyen de supports technologiques, tant de nature matérielle (les robots pédagogiques) que logicielle (programmation).

Nous visons à répondre à la question de recherche suivante: Comment les élèves élaborent-ils leur démarche de résolution créative de situations-problèmes dans un contexte de travail de collaboration entre des équipes d'élèves distribués dans différents pays ? Nous posons l'hypothèse selon laquelle la programmation informatique créative pourrait engager la compétence à résoudre des situations-problèmes complexes (Romero, 2016). Dans ces conditions, l'apprentissage dans le cadre de tâches de

¹ Nous distinguons le modelage de la modélisation. Le modelage, issu de l'enseignement explicite, consiste à montrer comment fait un expert. La modélisation correspond plutôt à une transformation de l'information par le biais d'une lecture relationnelle qui s'exprime par des registres numériques, géométriques ou algébriques.

résolution créative de situations-problèmes avec des robots pédagogiques est défini comme une activité humaine conduisant à explorer les moyens matériels et logiciels à disposition des élèves et ensuite pouvoir réaliser des expérimentations. Si les élèves prennent conscience des raisons ayant donné lieu à un certain résultat lié à l'expérimentation, cela permettrait de prendre des décisions pour avancer dans la démarche visant à résoudre la situation-problème.

La méthode de recherche à travers une démarche de programmation collective

À l'instar de Nizet et Laferrière (2005), nous considérons que lorsque les élèves travaillent sur un projet, ils se situent dans un contexte de conception-collaboration les conduisant à partager des connaissances et des expériences pour s'engager dans une négociation sur la compréhension partagée. DeBlois et Turcotte (2019) ont pu observer l'influence de la présence d'habiletés de type organisationnel et fonctionnel (Goodwin, 2000) au cours du développement de la résolution de problèmes mathématiques. Le processus en jeu conduirait les élèves à réaliser de nouvelles productions basées sur les explications données ou présentées par les pairs. Durant l'expérimentation, les productions originales pourront être exprimées par des artefacts de médias numériques (lors de la validation d'un wiki), audiovisuels (vidéo interactive), multimédia (les histoires numériques) ou un artefact de type programme informatique (projet réalisé avec Scratch). Il devient possible d'observer les raisons les ayant conduits à des prises de décision (Brennan et al., 2014; Resnick & Siegel, 2015).

Toutefois, une construction collaborative de connaissances se produira selon des processus qui ne sont pas toujours linéaires. En effet, dans une démarche de conception collaborative, différentes solutions sont mises à l'essai et vérifiées, de façon cyclique ou itérative, avant que les élèves ne s'entendent sur une solution temporaire permettant ensuite d'avancer vers un artefact qui répond à la situation-problème (Voogt et al., 2015).

Sélection de la tâche de programmation créative

Rappelons que pour susciter le développement d'une modélisation chez les élèves, la situation-problème proposée doit permettre d'engager une résolution créative (et pas uniquement une résolution heuristique de problème). Ainsi, le problème doit présenter une certaine complexité et permettre de créer une solution originale. En effet, si la solution est d'entrée de jeu préétablie, les élèves n'auront qu'à suivre la démarche guidée sans s'engager dans un processus d'analyse, de réduction d'informations et de représentation dans lequel se trouve la modélisation. Dans cette étude, nous analysons une tâche de programmation réalisée en équipe parmi les activités de programmation créative. Le défi de robotique pédagogique #R2T2AmeriCaraibe a engagé des élèves du secondaire situés dans quatre pays différents dans la programmation robotique de robots pédagogiques Thymio. Cette tâche est dénommée R2T2 (Mondada et al., 2016 p. 2).

Le défi #R2T2AmeriCaraibe fait suite au défi #R2T2 réalisé en Afrique du Sud et en Europe (Mondada et al., 2016). Le défi de robotique pédagogique #R2T2AmeriCaraibe a été organisé par l'École polytechnique fédérale de Lausanne (EPFL), le *National Centre of Competence in Research* (NCCR), le Laboratoire d'Innovation et Numérique pour l'Éducation (LINE) à Nice et l'Inspé de Martinique regroupant des équipes d'élèves âgés entre 8 et 14 ans. Pour réaliser ce défi, quatre équipes sont constituées dans quatre pays différents. Chaque équipe est responsable d'un robot pédagogique Thymio (Riedo et al., 2013) qui se trouve situé à l'École polytechnique fédérale de Lausanne (EPFL) en Suisse. Le défi R2T2 engage les élèves à collaborer de manière synchrone

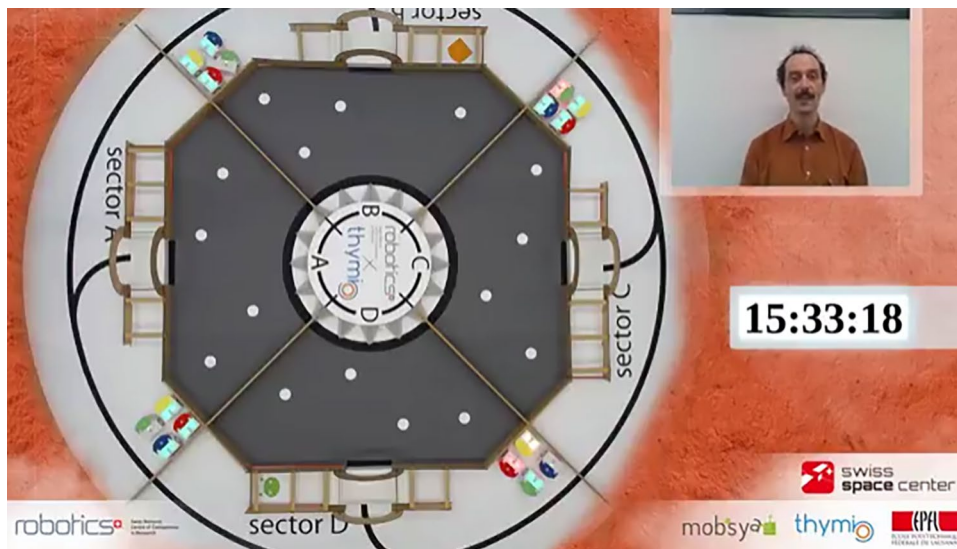


Fig. 3 Visualisation du tapis du défi #R2T2 à l'EPFL

sur quatre sites de pays différents afin de programmer le robot. L'activité se réalise de manière synchrone, engageant chaque équipe à programmer un robot qui est présent à Genève et envoyer à l'équipe enseignante sa programmation par le biais d'un dossier Dropbox partagé. Le défi est inspiré d'une mission de robotique spatiale, avec des idées et des essais préliminaires réalisés avec l'École internationale de Genève. Chacune des équipes dispose d'un tapis de jeu correspondant à l'emplacement de son robot sur le tapis de jeu centralisé à Genève (see Fig. 3).

Rôle des élèves et rôles des enseignants

Tant les élèves que les enseignants sont novices par rapport à la tâche. Pour chaque équipe, un enseignant ou un auxiliaire de recherche joue le rôle de facilitateur. Les facilitateurs commencent par une séance de préparation afin de définir le jour et les modalités de communication. Dans chaque pays, une séance d'environ trois heures permet aux facilitateurs et aux élèves de découvrir le robot Thymio et ses principales fonctionnalités. Les élèves sont invités à travailler en collaboration, sans le support du facilitateur. Celui-ci intervient uniquement à la demande des élèves.

Au cours de l'activité synchrone, les équipes doivent résoudre un ensemble d'étapes-défis qui exigent les différents traitements décrits préalablement pour leur propre robot et pour la situation commune des robots sur le tapis de jeu central situé en Suisse. Les équipes travaillent sur l'analyse de la situation et des actions à faire réaliser par leur robot pour ensuite envoyer leur programme informatique par Internet afin de programmer leur robot à l'EPFL. Il s'agit d'une situation-problème présentant de l'interdépendance positive (Johnson & Johnson, 2009) car les différentes équipes doivent se coordonner pour réussir la tâche en faisant arriver leurs robots respectifs, simultanément, à la zone de destination.

Nous avons observé l'équipe situé à Nice. Au cours de la tâche, deux observateurs ont pris en note le déroulement de la résolution. Un échange entre les observateurs, les élèves et les deux enseignants accompagnant la classe a eu lieu à la fin de l'activité dans le but de partager l'expérience vécue par chacun des acteurs et de recueillir les données.

Interprétation des observations

Au moment de réaliser le défi collaboratif, les équipes ont analysé le défi proposé (composante 1) pour ensuite établir une représentation de base de la situation (composante 2). Cette première modélisation de la situation a été réalisée de manière informelle, et selon un degré de profondeur variable, sans réaliser une représentation graphique (graphe ou schéma) de la situation-problème.

Un partage du travail à réaliser, de manière spontanée et non régulée par l'enseignante, a ensuite été décidé au sein des équipes. Ce partage a conduit certains membres à se centrer sur le suivi du robot par le biais de la visualisation de la salle de contrôle à Lausanne et ce, à partir d'une diffusion en temps réel de l'évolution de l'activité sur YouTube. Cette diffusion a conduit d'autres membres de la même équipe à se centrer sur la programmation. Les membres dédiés au suivi des robots et ceux qui ont réalisé la programmation ont rencontré des difficultés dans la répartition du travail à réaliser. Or, cette répartition est essentielle dans le processus de résolution collaborative de situations-problèmes (OECD, 2013), une forme d'habileté sociale de nature fonctionnelle et organisationnelle.

Interprétation selon les composantes de la pensée informatique

Nous avons pu repérer la présence des six composantes de la pensée informatique chez les élèves engagés dans l'activité de programmation créative. Ces composantes ont émergé de manière non-linéaire dans le temps. Le Tableau 1 présente l'analyse des composantes de la pensée informatique dans le cadre de l'activité R2T2. La création des programmes (composante 5) a nécessité de multiples itérations (composante 6) dues non seulement aux erreurs d'interprétation, de réduction d'informations et de représentation de la situation-problème de l'équipe, mais aussi aux interactions entre les robots des différentes équipes dont le travail a été peu négocié. En effet, des robots programmés par différentes équipes sont entrés en collision, forçant à reprogrammer la séquence d'actions pour rétablir les objectifs initiaux du programme. Il semble donc que les composantes 5 et 6 émergent davantage au moment de la validation à la fin de la programmation. C'est à ce moment que le rôle de l'enseignant concernant les boucles itératives permet de corriger la programmation.

Interprétation des observations selon le cadre didactique retenu

La modélisation, considérée comme une démarche nécessitant à la fois la capacité d'interpréter, de sélectionner et de représenter des informations de la situation-problème, conduit les élèves à s'approprié autrement la situation en jeu (Jonassen, 2003; Jonassen et al., 2005). Pour Jonassen et collègues (2005), bien que la modélisation prenne ancrage dans la composante liée à l'interprétation de la situation-problème, elle peut aussi se manifester lors de l'amélioration itérative des solutions informatiques.

Au moment de réaliser le défi collaboratif, l'analyse des démarches des équipes conduit à repérer que les élèves conviennent d'une interprétation du problème, interprétation qui correspond à une représentation mentale du défi. Toutefois, ces interprétations sont réalisées sans représentation graphique ou schématique. Cela pourrait expliquer que les premières expérimentations semblent d'un degré de profondeur variable. En effet, en l'absence d'une représentation numérique, graphique, algébrique ou schématique du problème, l'interprétation demeure personnelle plutôt que partagée. L'entrée dans le processus de réflexion devient multiple. Dans ces conditions, un changement de registre entre une représentation verbale écrite (énoncé) et une représentation verbale orale (échanges entre les élèves réalisés collectivement) semble conduire à une prise de distance entre la situation-problème et l'élève. Cette prise de conscience à l'égard de cette distance conduirait à une représentation partagée de la manière d'aborder le problème.

Nous interprétons la répartition du travail à réaliser entre les élèves comme une manifestation de régulation selon que les élèves se centrent sur le suivi du robot télécommandé ou sur la programmation.

Tableau 1 Analyse de la pensée d'un point de vue informatique dans le cadre de l'activité R2T2

Composantes de la pensée informatique	Activité des élèves au cours de l'activité R2T2	Observations repérées au cours de l'activité R2T2
<i>Composante 1.</i> Interprétation des constituantes de la situation-problème	En équipes, la situation-problème posée par le défi R2T2 est étudiée en vue de comprendre les ressources, les composantes technologiques et les visées de la situation-problème.	Les élèves identifient qu'il faudrait éviter les obstacles. Initialement, l'hôpital est perçu comme un obstacle, mais en relisant les consignes, les élèves comprennent qu'il s'agit d'une des destinations du robot « allez à l'hôpital », donc l'hôpital n'est pas un obstacle.
<i>Composante 2.</i> Modélisation	Interprétation, réduction et représentation des activités à réaliser au sein de l'équipe pour atteindre les objectifs.	La situation distribuée avec d'autres élèves situés dans d'autres pays conduit les équipes à négocier: « il y aura besoin d'une équipe de communication ».
<i>Composante 3.</i> Recherche du sens du système formel (code)	Appropriation de l'environnement de programmation VLP du studio Aseba.	Les élèves ont installé préalablement VLP. Ils testent les différentes fonctionnalités
<i>Composante 4.</i> Sélection des systèmes physiques (fonctionnalités d'un robot)	Se familiariser avec les Thymio et explorer les fonctionnalités de programmation	Les robots Thymio sont programmés par le biais de VLP. L'envoi de la programmation à Lausanne par le biais de l'adresse IP pose problème pendant une partie de l'activité, menant les élèves et facilitateurs à tester la bonne connectivité des robots et les adresses IP.
<i>Composante 5.</i> Création d'un programme informatique	Création des programmes pour chacune des étapes du défi R2T2	Les élèves choisissent de suivre la ligne noire présente sur une partie de la carte. Pour cela ils programment le robot pour réaliser le suivi de la ligne.
<i>Composante 6.</i> Évaluation de la solution et amélioration itérative	Mise à l'essai de la programmation avec les robots, coordination avec les autres équipes et analyse des remédiations à considérer	Au moment de la mise à l'essai, les élèves constatent les limites de leur programmation et décident d'ajuster leur programme: « quand il n'y a rien, tournez. N'utilisez qu'une seule flèche ».

En somme, ce cadre théorique illustre comment la démarche de ce défi collaboratif de programmation montre une approche itérative centrée sur de nombreux efforts de préparation sur le plan technologique (composantes 3 et 4). Cette démarche a toutefois négligé la modélisation (composante 2) lors de la réalisation et cela, dans l'ensemble des équipes des différents pays. Il semble que les composantes 1 et 2 émergent davantage durant la réalisation du défi et dans le contexte de nécessaires habiletés sociales. C'est donc à ce moment que le rôle de l'enseignant semble fondamental pour accompagner l'exploration de manière à susciter l'interprétation, la réduction d'informations et la représentation. Enfin, la validation a permis aux élèves de considérer l'erreur comme une occasion d'amélioration

Les multiples itérations dans la réalisation du programme informatique sont autant de manifestations d'un autre type de régulation: des expérimentations ajustées en fonction des collisions entre les robots des différentes équipes d'élèves. Ces interactions contribuent à des prises de conscience chez les élèves comme celle relative à la pertinence ou non de la séquence de la planification de la programmation. La validation, située à la fin de la programmation, contribue aussi à des prises de conscience à l'égard des raisons de la pertinence des solutions chez les élèves, autant de manifestations de la construction des connaissances. Dans ces conditions, la correction des boucles itératives par l'enseignant pourrait jouer sur le sens que les élèves attribuent à l'erreur comme composante de l'apprentissage (see Tableau 2).

L'approche itérative observée montre la dominance de la régulation dans la démarche de programmation. Le cadre didactique contribue à approfondir notre analyse de la réalisation. En outre, il permet de documenter la nature du rôle de l'enseignant qui accompagne le processus d'interprétation, de réduction d'informations et de représentation. Ainsi, l'éclairage de la didactique des mathématiques nous permet d'observer des différences dans des connaissances disciplinaires qui ont des sens différents en informatique et en mathématiques. Par exemple, la notion de variable en mathématiques marque un rôle dans une formule alors qu'en informatique ce concept réfère à une mémoire ou paramètre. La relation 'si... alors' en mathématiques, correspond à une implication logique alors qu'en informatique, cette relation donne lieu à l'exécution d'une procédure ou d'une action si la condition est vraie.

Discussion

Nous avons posé une hypothèse selon laquelle la démarche de résolution créative de situations-problèmes pour l'apprentissage de la programmation informatique pourrait contribuer au développement de la compétence à résoudre des problèmes complexes (Romero, 2016). Comparer les expériences des élèves en jeu permet de considérer la place de l'erreur, la complexité en jeu, l'organisation sociale en cours de résolution de situations-problèmes et la pensée créatrice collective émergeant de la négociation entre les productions des membres des équipes.

- a. Les composantes à considérer pour la construction de connaissances dans l'apprentissage de la programmation

Les contraintes informatiques sont introduites par les langages et les systèmes technologiques. En effet, ces environnements transforment l'environnement dans lequel les élèves évoluent, notamment parce que le milieu auquel ils sont sensibles dévoile un processus influencé par les rétroactions offertes par l'outil technologique (de Champlain et al., 2018). Nos analyses montrent que ces rétroactions semblent contribuer à des prises de conscience pouvant initier l'interprétation, la réduction d'informations et la représentation. Enfin, la prise de conscience de la valeur et du sens des savoirs offre un cadre nouveau en informatique pour sortir des sentiers familiers et reconnaître, par exemple, l'influence de l'ajout de contraintes devant une erreur, plutôt que la diminution de ces contraintes. Ainsi, du point de vue de l'apprentissage, nous nous attendions à ce que la rétroaction devant une erreur ou un bogue durant la démarche de programmation créative soit apportée par l'outil informatique et pas uniquement par un enseignant. Toutefois, nos analyses montrent que la rétroaction de l'outil informatique est interprétée de différentes façons par les membres des équipes au moment de mettre à l'essai leur programmation (composante 6). Les échanges sont influencés par ces différentes interprétations. Ainsi, nous avons pu constater que quand les élèves créent un programme (composante 5) dans le but de faire pivoter le robot, une mise à l'essai infructueuse du programme où le robot ne bouge pas (composante 6) donne lieu à des interprétations ou des hypothèses de différents types: le programme n'est pas correct, il n'a pas été bien transmis, ou encore, le robot présente un problème. Par l'exploration des différentes interprétations des erreurs ou des bogues rencontrés au cours de la

Tableau 2 Analyse de la pensée d'un point de vue didactique dans le cadre de l'activité R2T2

Composantes de la démarche	Activité des élèves au cours de l'activité R2T2	Observations repérées au cours de l'activité R2T2
Représentations mentales des élèves (interprétation des élèves)	En équipes, les élèves interprètent la situation-problème posée par le défi R2T2 en vue de comprendre les composantes technologiques et les objectifs de la situation-problème.	Passage d'un registre verbal écrit (consignes) à un registre verbal oral (représentation partagée du problème).
Les attentes des élèves	Anticipation des élèves à l'égard des activités à réaliser au sein de l'équipe pour atteindre les objectifs.	Engagement dans la programmation sans passer par une interprétation, une réduction et une ou plusieurs représentations pour atteindre le but le plus rapidement possible. Pour les élèves, la résolution de situations-problèmes avec l'informatique est anticipée comme étant un processus rapide.
Les procédures des élèves	Exploration des fonctionnalités de programmation avec les Thymio.	Expérimentation de déplacements du robot.
Les coordinations réalisées	Appropriation de l'environnement de programmation VLP du studio Aseba.	À la suite des collisions entre les différents robots des différents groupes d'élèves, la coordination entre représentations mentales et expérimentations se manifeste par des itérations et des expérimentations ajustées.
Les prises de conscience des élèves	Création des programmes pour chacune des étapes du défi R2T2.	Évaluation par les élèves des raisons de la pertinence ou non de la séquence de programmation en relation avec le but.

mise à l'essai, les élèves prennent conscience de la diversité des représentations mentales possibles, soit autant d'interprétations.

Du point de vue de l'enseignement, une telle approche exige de maintenir une médiation non seulement entre l'enseignant et les élèves mais aussi entre les élèves entre eux puisqu'ils doivent surmonter des obstacles et valider leur solution. Pour cela, des compétences organisationnelles sont nécessaires pour leur permettre de prendre en considération les idées des coéquipiers et de reconstruire de nouvelles interprétations du travail à réaliser. À l'instar des travaux d'Archer-Kath et al. (1994), de Johnson et Johnson (2009), de DeBlois et Turcotte (2019), nous constatons que la planification de l'enseignement doit faire intervenir le développement d'habiletés sociales chez les élèves afin de favoriser la diversité de points de vue et l'appréciation de l'apport de cette diversité pour enrichir les idées et identifier les meilleures solutions.

Nos observations montrent la pertinence, pour l'enseignant, d'engager les élèves dans une situation-problème qui permet d'utiliser le potentiel de l'informatique pour trouver des solutions que n'ont ni les élèves ni les enseignants. D'une part, des situations-problèmes trop simples ne contribuent pas à rendre la collaboration pertinente (Kirschner et al., 2009) puisque les efforts de négociation sont trop importants par rapport au partage d'idées. D'autre part, dans des tâches d'une certaine complexité, les échanges entre les élèves semblent contribuer à surmonter les difficultés d'interprétation qui ont été rencontrées et à éviter l'abandon de la tâche. En effet, l'échec n'est plus personnalisé mais devient un défi d'équipe.

Notre étude montre que des échanges entre les élèves ont lieu lorsque le travail porte sur les composantes liées à la production et à l'amélioration itérative du programme. Toutefois, ces échanges n'ont pas été suffisants pour négocier une répartition du travail qui aurait pu conduire à une solution satisfaisante. Ces observations nous conduisent à réfléchir au rôle de l'enseignant. Celui-ci doit s'assurer que les élèves échangent entre eux de manière à transformer leurs premières interprétations de la situation-problème.

Enfin, les expérimentations des élèves peuvent être considérées comme une manifestation de leur engagement et de leur créativité. Une pensée créatrice collective ne semble pas émerger s'il n'y a pas de valorisation de la négociation entre les productions des membres des équipes. Ainsi, dans notre étude, les composantes 1 et 2 auraient pu être à l'origine de la négociation. Elles sont escamotées au profit des expérimentations successives des solutions. Les élèves prennent conscience du besoin de reconsidérer leurs solutions seulement au moment des expérimentations (composantes 6).

b. Les activités de programmation au service du raisonnement des élèves

La manière d'étudier les problèmes à résoudre est documentée par les recherches sur les analyses a priori (Artigue, 2002). En outre, la résolution de problèmes mathématiques offre une complexité souvent « évaluée » au préalable par l'enseignant qui détient déjà une solution, provoquant une asymétrie dans la relation enseignant-élève. L'univers des contraintes d'un problème en sciences numériques, quant à lui, est ouvert parce que des erreurs générées sont imprévues, notamment dans des environnements de programmations non simplifiées. Le nombre de solutions et de démarches possibles est souvent illimité.

Rappelons que les sciences numériques se sont d'abord développées dans un contexte professionnel marqué par l'autoformation. D'autre part, les mathématiques sont travaillées à partir de concepts établis alors que les sciences numériques ont connu des évolutions tellement rapides depuis les années 1970 qu'il a fallu construire les concepts et procédures en concomitance avec le développement du domaine. Enfin, les sciences numériques font intervenir un support technologique qui influence la manière dont la programmation est affichée. La question qui se pose alors est celle d'une didactique de l'informatique.

Dans cette étude, la programmation du défi R2T2 donne lieu à un déplacement des robots Thymio. Ce support technologique vient influencer le raisonnement. En effet, le feedback immédiat du matériel informatique, par les collisions entre les robots, a une influence sur la manière dont les élèves entrent

dans les ajustements. Ces derniers prennent leur origine dans les coordinations entre les représentations mentales, les attentes entretenues et les expérimentations. Dans ces conditions, l'informatique pourrait conduire les élèves à entrer dans la transformation des interprétations par le feedback généré par l'outil. Un cadre théorique didactique pourrait contribuer à analyser les processus mis en jeu par les élèves durant l'activité.

Comment les élèves élaborent-ils leur démarche de résolution créative de situations-problèmes dans un contexte de travail de collaboration qui se réalise à distance avec des référents culturels différents ? Nous observons, dans l'activité de résolution de situations-problèmes étudiée en robotique, que les élèves s'engagent rapidement dans la programmation (composante 5) sans passer par une interprétation, une réduction d'informations et une représentation de ces dernières (composantes 1 et 2). Des observations semblables ont également été repérées dans des activités de résolution de problèmes mathématiques où les élèves se sont engagés rapidement sur une solution sans considérer les relations entre les données, ce qui a conduit à la recherche de repères ou de méthode de travail (DeBlois & Bélanger, 2016).

Conclusion

Cette étude consistait à analyser d'une démarche de programmation créative collective et à la considérer comme une occasion de développement du raisonnement chez les élèves. La résolution de la situation-problème proposée, par le biais de la programmation créative, a conduit à observer certaines composantes spécifiques à la pensée informatique, comme les composantes liées à l'alphabétisation du code et à la connaissance des systèmes technologiques, mais aussi à des composantes liées aux habiletés sociales et à la conception de l'erreur comme occasion d'amélioration. Nous pouvons reconnaître qu'une démarche de programmation créative collective pourrait contribuer au développement du raisonnement par l'expérience de recherche de sens au cours duquel plusieurs registres de représentation sont invoqués simultanément.

Notre analyse montre aussi l'importance de développer des études sur l'apprentissage de la programmation (Drot-Delange, 2018) qui puisse considérer les caractéristiques du numérique. Ces études pourraient contribuer à développer une rigueur pour interpréter les prises de conscience réalisées par les élèves lors d'un apprentissage de la programmation. En outre, exiger des enseignants d'interpréter les connaissances mises en jeu durant le processus de résolution de situation-problème exige de les former à ce type d'interprétation à partir d'un cadre théorique qui guide leur réflexion, notamment lors des démarches de résolution de situations-problèmes. Cette formation pourrait offrir des occasions de concevoir des situations-problèmes qui font intervenir des enjeux collectifs.

Declarations

Approbation éthique L'ensemble de participants ont donné leur consentement éclairé à la participation à l'étude. Le projet ANR CreaMaker a l'autorisation du CERNI (2019–6) de l'Université Côte d'Azur.

Conflits d'intérêt Les auteurs de ce chapitre déclarent n'avoir aucun conflit d'intérêt.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Références

- Abiteboul, S., & Dowek, G. (2020). *The Age of Algorithms*. Cambridge University Press.
- Archer-Kath, J., Johnson, D. W., & Johnson, R. T. (1994). Individual versus group feedback in cooperative groups. *The Journal of Social Psychology*, 134(5), 681–694.
- Artigue, M. (2002). Ingénierie didactique : quel rôle dans la recherche didactique aujourd'hui ? *Les Dossiers des Sciences de l'Éducation*, 8, 59–72.
- Barma, S., Voyer, S., Dewailly, C., Vincent, M. C., Duguay, S., & Kouamkam, R. K. (2015). Réaliser une étude de cas multiple qui vise à affiner les connaissances sur l'usage pédagogique ou didactique de la programmation dans les écoles du Québec: rapport final. Centre de recherche et d'intervention sur la réussite scolaire (CRIRES).
- Bélanger, J.-P., DeBlois, L., & Freiman, V. (2014). Interpréter la créativité du raisonnement dans les productions d'élèves en mathématiques d'une communauté d'apprentissages multidisciplinaires interactifs. *Éducation et Francophonie*, 42(2), 44–63.
- Besançon, M., Barbot, B., & Lubart, T. (2011). Évolution de l'évaluation de la créativité chez l'enfant de Binet à nos jours. *Recherches & éducatives*, (5), 215–226.
- Bjögvinsson, E., Ehn, P., & Hillgren, P.-A. (2012). Design things and design thinking: Contemporary participatory design challenges. *Design Issues*, 28(3), 101–116.
- Brennan, K., Balch, C., & Chung, M. (2014). *Creative computing* (Harvard University Press). Cambridge, MA. Retrieved from <http://scratched.gse.harvard.edu/guide/>
- Care, E., Scoular, C., & Griffin, P. (2016). Assessment of collaborative problem solving in education environments. *Applied Measurement in Education*, 29(4), 250–264.
- Chevalier, M., Giang, C., Piatti, A., & Mondada, F. (2020). Fostering computational thinking through educational robotics: A model for creative computational problem solving. *International Journal of STEM Education*, 7(1), 1–18.
- de Champlain, Y., DeBlois, L., Robichaud, X., Freiman, V. (2018). The Nature of Knowledge and Creativity in a Technological Context in Music and Mathematics: Implications in Combining Vygotsky and Piaget's Models. In *Creativity and Technology in Mathematics Education*. Eds, Janet Tassell & Viktor Freiman, Springer. 479–505.
- DeBlois, L. (2001). Un modèle d'interprétation des activités cognitives pour des élèves qui éprouvent des difficultés d'apprentissage en mathématiques. In *Actes du colloque Constructivisme : Usages et Perspectives en Éducation* (Vol. 2, pp. 565–573).
- DeBlois, L. (2003). Interpréter explicitement les productions des élèves : une piste.... *Éducation et francophonie*, XXXI, 2, 176–199.
- DeBlois, L., Bélanger, J.-P. (2016). La résolution de problèmes vue par les élèves qui manifestent des réactions d'évitement, d'anxiété ou d'agitation. *Vivre le primaire*. 29 (2). 62–66.
- DeBlois, L., & Turcotte, B. (2019). Le développement des habiletés sociales comme support aux apprentissages cognitifs. *Éducation et francophonie*, 47(3), 11–34.
- Drot-Delange, B. (2018). Reconfiguration de l'enseignement de l'informatique à l'école primaire : quelle conscience disciplinaire chez les professeurs des écoles stagiaires?. *Recherches en didactiques*, (1), 27–40.
- Duval, R. (2006). A cognitive analysis of problems of comprehension in a learning of mathematics. *Educational studies in mathematics*, 61(1), 103–131.
- Goodwin, K. A. (2000). *Implicit and explicit memory retrieval of prior information during problem solving*. The Florida State University.
- Herborn, K., Mustafić, M., & Greiff, S. (2017). Mapping an Experiment-Based Assessment of Collaborative Behavior Onto Collaborative Problem Solving in PISA 2015: A Cluster Analysis Approach for Collaborator Profiles. *Journal of Educational Measurement*, 54(1), 103–122.
- Jonassen, D. (2003). Using cognitive tools to represent problems. *Journal of Research on Technology in Education*, 35(3), 362–381.
- Jonassen, D., Strobel, J., & Gottdenker, J. (2005). Model building for conceptual change. *Interactive Learning Environments*, 13(1–2), 15–37.
- Johnson, D. W., & Johnson, R. T. (2009). An educational psychology success story: Social interdependence theory and cooperative learning. *Educational Researcher*, 38(5), 365–379.
- Kafai, Y. B., & Resnick, M. (1996). *Constructionism in practice: Designing, thinking, and learning in a digital world* (Vol. 1). New York, NY : Routledge.
- Kapur, M., Voiklis, J., & Kinzer, C. K. (2005). Problem solving as a complex, evolutionary activity: A methodological framework for analyzing problem-solving processes in a computer-supported collaborative environment. In *Proceedings of the 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years!* (pp. 252–261). International Society of the Learning Sciences.
- Kirschner, F., Paas, F., & Kirschner, P. A. (2009). A cognitive load approach to collaborative learning: United brains for complex tasks. *Educational psychology review*, 21(1), 31–42.

- Komis, V., Karsenti, T., Depover, C., Romero, M., Brugmann, J., Strebelle, A., ... & Misirli, A. (2018, February). Robots de sol et robots humanoïdes à l'école primaire. In *Didapros 7–DidaSTIC. De 0 à 1 ou l'heure de l'informatique à l'école*.
- Kramer, J. (2007). Is abstraction the key to computing? *Communications of the ACM*, 50(4), 36–42.
- Lajoie, C., & Bednarz, N. (2016). La notion de situation-problème en mathématiques au début du XXIe siècle au Québec: rupture ou continuité?. *Canadian journal of science, mathematics and technology education*, 16(1), 1–27.
- Lithner, J. (2008). A research framework for creative and imitative reasoning. *Educational Studies in mathematics*, 67(3), 255–276.
- Lockwood, E., DeJarnette, A. F., Asay, A., & Thomas, M. (2016). *Algorithmic Thinking: An Initial Characterization of Computational Thinking in Mathematics*. Proceedings of the 38th Annual Meeting of the North American Chapter of the International Group for the Psychology of Mathematics Education. Tucson, AZ: The University of Arizona.
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51–61.
- MELS. (2006). Programme de formation de l'école québécoise : éducation préscolaire et primaire. Québec : Ministère de l'éducation, Gouvernement du Québec.
- Mondada, F., Bonnet, E., Davrajh, S., Johal, W., & Stopforth, R. (2016). R2T2 Robotics to integrate educational efforts in South Africa and Europe. *International Journal of Advanced Robotic Systems*, 13(5), 1729881416658165.
- Nizet, I., & Laferrière, T. (2005). Description des modes spontanés de co-construction de connaissances : contributions à un forum électronique axé sur la pratique réflexive. *Recherche et Formation*, 48, 151–166.
- OECD. (2013). *PISA 2015 Collaborative Problem Solving Framework*. OECD Publishing.
- Owen, C. (2007). Design thinking: Notes on its nature and use. *Design Research Quarterly*, 2(1), 16–27.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- Papert, S., & Harel, I. (1991). Situating constructionism. *Constructionism*, 36, 1–11.
- Resnick, M., & Rusk, N. (2020). Coding at a crossroads. *Communications of the ACM*, 63(11), 120–127.
- Resnick, M., & Siegel, D. (2015, November 10). A Different Approach to Coding. Retrieved from <https://medium.com/bright/a-different-approach-to-coding-d679b06d83a#b9jcw2js5>
- Riedo, F., Chevalier, M., Magnenat, S., & Mondada, F. (2013). Thymio II, a robot that grows wiser with children. In 2013 IEEE Workshop on Advanced Robotics and its Social Impacts (pp. 187–193). IEEE.
- Riley, D. D., & Hunt, K. A. (2014). Computational thinking for the modern problem solver. CRC press.
- Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior*, 72, 678–691.
- Romero, M. (2016). De l'apprentissage procédural de la programmation à l'intégration interdisciplinaire de la programmation créative. *Formation et Profession*, 24(1), 87–89. <https://doi.org/10.18162/fp.2016.a92>
- Romero, M., Hyvönen, P., & Barberà, E. (2012). Creativity in Collaborative Learning across the Life Span. *Creative Education*, 3(4), 0–0.
- Romero, M., Lepage, A., & Lille, B. (2017). Computational thinking development through creative programming in higher education. *International Journal of Educational Technology in Higher Education*, 14(1), 1–15.
- Runco, M. A., & Jaeger, G. J. (2012). The standard definition of creativity. *Creativity Research Journal*, 24(1), 92–96.
- Voogt, J., Laferrière, T., Breuleux, A., Itow, R. C., Hickey, D. T., & McKenney, S. (2015). Collaborative design as a form of professional development. *Instructional Science*, 43(2), 259–282.
- Vygotski, L. S. (1985). *Pensée et langage*. Paris: Messidor–Editions Sociales.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33–35.
- Wing, J. M. (2011). Computational thinking. In *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)* (p. 33). IEEE.