Que disent les sciences de l'éducation à propos de l'apprentissage du code ?

Margarida Romero, Stephanie Noirpoudre, Thierry Viéville

Que disent les sciences de l'éducation à propos de l'apprentissage du code ?

Archive

L'apprentissage de la pensée informatique est une tendance croissante à l'échelle internationale. Dans ce contexte, des projets comme Class'Code ont permis développer des formations pour les éducateurs, des ressources et une communauté d'apprentissage interdisciplinaire. A l'échelle internationale, des nombreuses expérimentations ont eu lieu dans différents contextes, bien que l'analyse de ces différents pilotes n'ait pas été conduit de manière systématique. En vue d'établir un état des connaissances ces apprentissages, nous partageons ici une lecture et une synthèse de différentes études produites dans le domaine des sciences de l'éducation. Cette revue pourra servir à analyser Class'Code en vue d'établir un bilan du dispositif.

Travail effectué sous l'expertise de Margarida Romero du

laboratoire LINE.

Une réponse à quatre niveaux.

Lorsqu'on commence à explorer la littérature en sciences de l'éducation à propos de ce qu'on nomme l'« apprentissage du code », il semble que la réponse soit donnée à quatre niveaux.

– Au niveau de *l'apprentissage de la programmation* à partir d'outils de programmation visuelle. Parmi ces outils, les études sont centrées essentiellement sur Scratch un grand nombre d'études a permis d'établir l'intérêt de cet outil et pratique, on échantillonnera ici quelques éléments.

– Au niveau de *la robotique ludique et éducative* plusieurs études sont aussi disponibles, elles établissent la faisabilité de faire de la robotique avec les enfants et en montrent quelques vertus et limites.

– Quelques études s'intéressent enfin à *l'apprentissage de la pensée informatique* indépendamment de ces deux volets plus technologiques, par exemple de manière débranchée, on en citera.

– Enfin bien que ce ne soit pas à proprement parler le sujet, la question de *l'apprentissage avec le numérique ou des usages du numérique*, et celui des pédagogies innovantes viennent s'inviter dans notre problématique, et les deux ouvrages de synthèse de Tricot et Amadieu fournissent une réponse tout à fait utile à notre niveau, on le résumera ici.

Une cinquantaine de références de la littérature internationale

en sciences de l'éducation est revue ici.

À quoi bon apprendre à programmer avec Scratch ? Plusieurs études visent analyser l'intérêt de l'apprentissage de compétences avec Scratch pour ... ce pourquoi il est fait ! À savoir l'apprentissage de l'algorithmique et de la programmation, de manière plus engageante que les méthodes traditionnelles, pour obtenir des résultats similaires, ceci sans différence de genre. Cet apprentissage est vraiment bien reçu y compris par des enfants en difficulté par ailleurs, tandis qu'il peut parfois bloquer des enfants trop scolaires, qu'il faut aider dépasser ce blocage. En revanche, en tant que tel, il ne semble pas avoir d'effet positif direct sur le développement d'autres compétences, sauf chez des élèves déjà plus matures. L'apprentissage de la programmation est très souvent couplé à des activités débranchées, des dispositifs tangibles ou des approches participatives, et dans ses conditions élargies, il y a alors une vraie différenciation positive.

En quoi la robotique ludique et éducative est elle utile ? Les études disponibles confirment d'abord la faisabilité d'introduire la robotique éducative, y compris auprès de très jeunes enfants. Comme précédemment les activités de robotiques sont attractives mais n'offrent pas systématiquement de levier pour aider à d'autres apprentissages, surtout si on en reste à l'acquisition de savoir faire technologiques. Si ce niveau est dépassé alors l'impact positif de la robotique éducative sur l'apprentissage en science, technologie, ingénierie et mathématiques, liant ces disciplines, est bien établi, et son détournement pour

permettre d'autres apprentissages de manière plus engageante pour les apprenant·e·s est factuellement observé, y compris pour des jeunes scolairement en difficulté.

Comment évaluer l'impact d'une initiation à la pensée informatique ? Quand on dépasse le simple apprentissage de la programmation, par exemple à partir d'activités débranchée ou en orientant le travail vers l'apprentissage de la pensée algorithmique, on arrive alors à vraiment établir un véritable effet positif, au niveau primaire et début de collège, y compris avec des enseignant·e·s nouvellement formé·e·s. Le levier est le passage de l'apprentissage procédural de la programmation à l'intégration pluri-disciplinaire de la programmation créative. Il s'agit de passer du simple apprentissage de la programmation, mais en s'en servant pour une véritable initiation à la pensée informatique. Le point important est que cette notion est bien établie et sa pertinence validée par plusieurs études. On note alors de vrais effets positifs sur par exemple la capacité à résoudre de problèmes, et dans une moindre mesure le raisonnement et la spatialisation. Ces résultats ont pu être établis au niveau de groupes d'étudiant·e·s universitaires (donc des futurs enseignant·e·s) et aussi des élèves du secondaire.

Et au delà que dire du numérique en éducation et de pédagogies innovantes ? L'utilisation des ressources numériques citées ici augmente la motivation, mais cela peut être sans lien avec les performances d'apprentissage, tandis qu'il est essentiel, comme c'est le cas dans le manuel «1,2,3 codez», que les activités non-ponctuelles soient scénarisées et implique la production d'hypothèses et d'inférences en lien

avec les compétences à partager. L'apprentissage du code peut favoriser l'autonomie et la mise en place d'une démarche de recherche, mais ce n'est pas automatique, cela nécessite un accompagnement sur ces deux volets. Expliquer une solution à l'élève, pour lui permettre de bâtir la suivante, par exemple une variante, lui permettre de s'inspirer de solutions existantes à adapter sont des moyens de réduire le gap trop grand qu'impose un apprentissage autonome complet.
C'est parce que l'apprentissage du code relève en partie d'un savoir-faire que faire manipuler est pertinent et efficace, mais l'apprentissage de la pensée informatique relève aussi de connaissances notionnelles, et là c'est le fait d'être actif cognitivement qui est important.
L'apprentissage en groupe et par projet permet d'aborder de multiples compétences, mais peut aussi très bien échouer. On peut favoriser cet apprentissage en aidant les élèves à s'organiser en leur donnant des rôles et définissant des étapes progressives. Une telle démarche permet d'engager les élèves des activités, et leur perception a posteriori est très positive.

Conclusion. L'évolution de notre environnement, a fait émerger un nouveau but pour l'école (enseigner l'informatique et la pensée informatique) et de nouvelles tâches scolaires : programmer, mais aussi comprendre le codage des objets numériques se faire une représentation suffisante des grands systèmes numériques pour en avoir un usage éclairé. Il s'agit là d'un évènement assez rare, la plupart des tâches scolaires datant de plusieurs siècles. C'est finalement en puisant dans des démarches pédagogiques très anciennes et bien établies (la démarche par projet est une idée plusieurs fois centenaire) que ces quelques résultats en science de l'éducation semblent

confirmer les éléments très positifs ressentis par les acteurs et actrice de Class´Code. Il n'y pas lieu d'en être surpris ni d'en tirer gloire : c'est simplement que dès la conception et la mise en place de ce projet les autrices et auteurs se sont appuyés sur des travaux existants sur ces sujets et n'ont pas hésité à reprendre ce qui était su comme étant les bonnes façons de procéder.

Annexe: Étude bibliographique à l'appui des éléments précédents.

- Introduction. L'apprentissage de la pensée informatique est une tendance croissante à l'échelle internationale [Heintz-Mannila-Färnqvist-2016], et les sciences de l'éducation peuvent dont désormais travailler sur cet objet d'étude. La plupart des références sont très récentes (postérieures à la mise en place du projet Class´Code) et il semble que l'étude de ces aspects soit en plein développement, comme par exemple par [Chiprianov-Gallon-2016] . La présente revue ne peut prétendre à l'exhaustivité mais vise à être représentative des connaissances disponibles. On observe aussi un biais "en faveur" de ces apprentissages, une attention toute particulière a donc été portée aux quelques études "critiques" qui montrent aussi les limites et désavantages de ces approches. Seules les études qui apportent de vrais résultats factuels ou les méta-analyses ou reviews de telles études sont prises en compte ici, en non des « témoignages » comme l'expliquent par exemple [Hickmott-Prieto_Rodriguez- Holmes-2017]. Il faut noter que toutes les études font preuves de prudence, ceci dès

les premières études en science de l'éducation sur le langage Logo proposé par Papert [Clements-meredith-1992].

- Apprentissage de la programmation. Un exemple type d'étude sur l'apprentissage avec Scratch est [Wilson-Moffat-2010] avec une étude de cas sur les compétences de programmation sur vingt enfants de huit ans en zone défavorisée de Glåsgow avec évaluation. C'est une expérience positive permettant de dépasser les frustrations et angoisses de l'apprentissage, des enfants qui apprennent bien même s'ils ont des difficultés et quelques enfants révélant de hautes performances, tandis que certains enfants performants au niveau scolaire vont mal s'approprier l'apprentissage. De plus la progression cognitive plus générale reste réduite, ce qu'on retrouve chez [Kalelioglu-Gülbahar-2014] sur une cinquantaine de jeune de l'école primaire pour qui l'apprentissage de Scratch permet d'augmenter les compétences en programmation, évidemment, et est vu comme un expérience positive, mais sans augmentation de compétence en matière de résolution de problème en général, à moins comme l'analyse bibliographique des auteurs le montre de concevoir des activités spécifiques de résolution de problèmes à partir de Scratch. Plus relatif encore, l'usage de Scratch par rapport à des méthodes traditionnelles d'apprentissage sur papier n'est pas supérieur au niveau des résultats [Tekerek-Altan-2014] sur un groupe de début de collège, mais bien au niveau du plaisir d'apprendre et de l'engagement des élèves. Cette étude montre aussi l'absence de différence de

performances entre filles et garçons. Ce que l'on mesure de positif reste l'engagement des élèves comme le montre [Colón-Romo-2016] sur une classe de niveau collège avec des résultats d'évaluation trop dispersés pour conclure. Ce qui semble bien établi c'est la vertu didactique de Scratch en matière d'initiation à l'informatique comme reporté par [Korkmaz-2016] sur une cinquantaine d'étudiants d'entrée d'université donc en mesure d'apprendre un langage de programmation professionnel, cet apprentissage étant significativement amélioré par une utilisation préalable de Scratch. On a pu observer par ailleurs [Fesakis-Serafeim-2009] que l'utilisation de Scratch sur des étudiant·e·s post-bac, de par l'augmentation de confiance en soi face aux outils numériques, avait une influence positive sur leurs usages du numérique dans l'éducation. Pour mesurer un effet positif sur les performances scolaires des élèves, il semble qu'il faille attendre une certaine maturité comme le montre [Moreno_León-Robles-Román_González-2016] avec une étude portant sur plus de 150 apprenants avec des résultats cohérents avec les études précédentes, sans effet notoire de l'apprentissage du code sur les plus jeunes, mais significatives à partir du début du collège, où là l'effet est double par rapport à ce que peut apporter les mathématiques sur le développement pluridisciplinaire. À l'interface entre apprentissage de la programmation et informatique "tangible" [Horn-Solovey-etal-2009] montrent qu'apprendre à programmer en manipulant les blocs de code à travers des objets tangibles (comme un jeu de construction) augmentent les performances en matière de première

initiation à la programmation dans le cas d'ateliers familiaux dans un musée. Ce résultat est à rapprocher de ce qu'obtient [Boissel-2017] avec un système tangible pour l'initiation scolaire à la programmation pour des enfants voyants et non-voyants. Un autre étude concerne l'apprentissage de la programmation avec des objets tangibles et [Correll-Wailes-Slab-2014] montrent à travers des questionnaires l'apport positif sur l'engagement à étudier les sciences. Notons aussi que Scratch peut se prêter à des démarches pédagogiques vraiment engageantes pour l'enfant comme le système Scratch Community Blocks qui permet aux enfants eux-mêmes d'analyser leur données d'apprentissage [Dasgupta-Hill-2017].

- Robotique éducative. On ne discute ici que de l'apprentissage de la robotique et non de l'utilisation de robots en éducation comme par exemple [Gordon-breazeal-engel-2015], qui étudient dans quelle mesure l'utilisation d'un robot pour aider des enfants à apprendre quelque chose favorise/encourage leur curiosité. La revue systématique de la littérature en robotique éducative de [Benetti-2012] simplifie notre propre analyse, elle montre que de vraies améliorations en matière d'apprentissage sont observées, mais ce n'est pas toujours le cas, le facteur différentiateur étant à la fois méthodologique (attitude de l'enseignant, espace de travail adapté), et curriculaire (quelles compétences sont visées). Le danger semble être de faire de la robotique pour la robotique, de rester au niveau d'un travail de "recopie technologique". Peu d'études regardent, au préalable, si la prise en main du

système robotique est aisée, à contrario de [Desprez-Noirpoudre-2017] prennent la peine de valider l'utilisabilité de leur dispositif. Cette étude montre aussi que sur 70 publications analysées, seule une 10zaine comporte une véritable étude quantifiée au delà de déclarations à priori, ou de témoignages ponctuels, comme [Robinson-2005], qui ne sont pas dénués d'intérêts, mais dont la validité générale est discutable. Plus récemment [Kim-Kim-etal-2015] montrent l'intérêt de l'expérimentation robotique pour l'apprentissage en science, technologie, ingénierie et mathématiques, liant ces disciplines, les résultats positifs ou non dépendant beaucoup de l'engagement motivationnel de l'enseignant. Le lien entre robotique éducative et pensée informatique (voir section suivante) est fait par [Atmatzidou-Demetriadis-2016] qui notent le besoin de temps long pour ses apprentissages, avec une homogénéité sur qualité des performances quelque soit l'âge ou le genre, avec le fait que les filles travaillent sur un temps plus long (je note une sur-interprétation des auteurs qui disent "elles ont besoin de plus de temps pour arriver au même résultat" alors que les faits montrent "qu'elles y consacrent plus de temps", si le temps avait été contraint rien ne prouve que les résultats eussent été moins bon). Au niveau de plus petits [Bers-Flannery-etal-2014] établissent sur une cinquantaine d'enfants de début de maternelle leur capacité à réellement s'approprier les premières notions de robotique adaptées à leur âge. Une observation qualitative [Sullivan-Kazakoff-etal-2013] confirme la faisabilité d'initier les plus jeunes (ici niveau maternelle) à la robotique éducative. De

manière plus précise, en se basant sur des interviews, [Highfield-Mulligan-Hedberg-2008] montrent les apprentissages en matière de spatialisation et géométrie à partir de la manipulation d'un objet programmable et d'activité impliquant le corps. Plus récemment [Spalaôr-Benetti-2017] font une revue de littérature systématique de l'utilisation de la robotique éducative pour d'autres apprentissages, et concluent sur le potentiel réel, établi dans plusieurs cas, surtout quand l'approche est combinée avec une approche pédagogique constructiviste. Dans ces travaux sur les ressources d'IniRobot avec des Thymio-II [Roy-2015] montre une réelle augmentation de compétences sur un groupe de 24 enfants de primaire avec un taux final de 93% au post-test établissant que l'apprentissage est vraiment accessible à toutes et tous. Le travail pédagogique avec le Thymio-II démontrant sa capacité à engager les enfants du primaire et collège a été établi par les créateurs de l'objet [Riedo-Chevalier-Magnenat-Mondada-2013], y compris la capacité à s'approprier des notions de bases de robotique [Magnenat-Riedo-Bonani-Mondada-2012].

- Initiation à la pensée informatique. Quand on dépasse le simple apprentissage de la programmation, par exemple à partir d'activités débranchées comme étudié sur deux classes avec groupe témoin [Brackmann-Román_González-etal-2017] on observe alors une amélioration statistiquement significative des performances des enfants en matière de manipulation de la pensée informatique en éducation au sens défini par [Tchounikine-2017] ou [Romero-2016], et formalisé dans le référentiel proposé

par [ISTE-2015] (voir page 16 pour avoir une vue du référentiel). La validité de ce référentiel a été étudiée et validée, au moins dans le contexte d'étudiant·e·s universitaires [Korkmaza-Çakirb-Özdenc-2017]. En citant le premier avec [Tricot-2017] « les compétences sous-jacentes à ce qu'on appelle la » pensée informatique « , [...] c'est-à-dire : savoir décomposer un problème en sous-problèmes plus simples, savoir réfléchir aux tâches à accomplir pour résoudre un problème en termes d'étapes et d'actions (algorithme), savoir décrire les problèmes et les solutions à différents niveau d'abstraction, ce qui permet d'identifier des similitudes entre problèmes et par suite de pouvoir réutiliser des éléments de solution». Une telle amélioration de ces compétences est confirmée dans [Moreno_León-Robles-2015] sur une soixantaine d'élèves de 9 à 11 ans sur deux niveaux de classe, après un travail incluant une initiation à la programmation tournée vers l'apprentissage de la pensée algorithmique, avec un élément clé: les enseignants venaient juste de se former à ce nouvel apprentissage, tandis que l'appétence pour l'anglais est significativement augmentée cet apprentissage se faisant en anglais. Afin d'établir ces compétences de computational thinking, [Moreno_León-Robles-Román_González-2015] relient chaque compétences à l'usage de constructions du langage Scratch, comme résumé ici (reproduit du papier précédent) :

CT Concept	Competence Level			
	Null (0)	Basic (1 point)	Developing (2 points)	Proficiency (3 points)
Abstraction and problem decomposition	-	More than one script and more than one sprite	Definition of blocks	Use of clones
Parallelism	-	Two scripts on green flag	Two scripts on key pressed, two scripts on sprite clicked on the	Two scripts on when I receive message, create clone, two scripts when %s is > %s, two

			same sprite	scripts on when backdrop change to
Logical thinking	-	If	If else	Logic operations
Synchronization	-	Wait	Broadcast, when I receive message, stop all, stop program, stop programs sprite	Wait until, when backdrop change to, broadcast and wait
Flow control	-	Sequence of blocks	Repeat, forever	Repeat until
User Interactivity	-	Green flag	Key pressed, sprite clicked, ask and wait, mouse blocks	When %s is >%s, video, audio
Data representation	-	Modifiers of sprites properties	Operations on variables	Operations on lists

Table 1. Competence Level for each CT concept.

et proposent un outil http://www.drscratch.org pour évaluer le niveau de compétence par la présence de l'utilisation de telles constructions, à partir du cadre proposé par [Brennan-Resnick-2012] et repris dans un manuel tel que [Ozcinar-Wong-Ozturk-2018] pour ne citer que le plus récent. Dans ces études, le levier est le passage de l'apprentissage procédural de la programmation à l'intégration pluri-disciplinaire de la programmation créative, comme expliqué dans [Romero-2016]. Un exemple de transfert de compétences entre l'apprentissage de la programmation et d'autres apprentissages est donné par l'amélioration significative de la distinction gauche-droite après une activité robotique [Romero-Dupond-Pazgon-2106], d'autres types de transferts sont assez probables mais restent à établir. Cet effet positif de sur la capacité à résoudre de problèmes, et dans une moindre mesure le raisonnement et la spatialisation est établie sur des groupes d'élèves de fin de primaire et collège [Román_González-Pérez_González-Jiménez_Fernández-2017].

C'est encore avec une activité débranchée que [Greff-1998]

montre sur des enfants de 4 à 6 ans, une augmentation effective de compétences dans les domaines de la motricité, la latéralisation, la sémiologie de l'image, la communication, la rigueur, la résolution de problème, la représentation de parcours, etc. De même, le lien entre la capacité à développer une pensée informatique et d'autres facteurs liés à l'apprentissage a pu être étudié [Duraka-Saritepeci-2018] on y note une corrélation forte avec la façon de penser, et le niveau en science, mais pas d'autres facteur comme le genre, le rejet des mathématiques ou le taux d'usage du numérique. Dans [Israel-Pearson-etal-2015] l'apport à des élèves en difficulté est établi, y compris en enseignant ces compétences au sein d'autres disciplines. Relativement au genre, on observe un biais mineur en faveur des élèves masculins qui augmente avec l'avancée des études, en lien fort avec la pression culturelle [Román_González-Pérez_González-Jiménez_Fernández-2017]. Ces résultats sont en cohérence avec les travaux de [Chen-Shen-etal-2017], qui incluent l'usage de la robotique éducative.

L'évaluation par compétences sur ce sujet a été étudié finement par [Lepage-Romero-2017] et [Romero-Lepage-Lille-2017]. Pour évaluer l'engagement des participant·e·s dans une démarche critique, empathique et créative de résolution de problèmes d'une certaine complexité et authenticité, tout en faisant appel à l'usage de stratégies et de processus des sciences informatiques pour la création d'une ou plusieurs solutions, un outil de grille d'évaluation et validé.

- Perspectives pédagogiques. Le travail de synthèse et de

publication de [Amadieu-Tricot-2014] sur les mythes et réalités à propos d'apprendre avec le numérique et de [Tricot-2017] sur l'innovation pédagogique, dont on garde les résumés ici, (voir aussi la conférence et les slides sur ce sujet) permettent de replacer l'apprentissage de ce qu'on appelle le code, dans un contexte plus général. Ces deux livres font un travail de synthèse de plusieurs dizaines de références. Il semblerait que l'avantage majeur des pédagogies déployées se situent au sens de l'engagement (comme re-défini par [Bouvier-Lavoué-Sehaba-2014]) et en cohérence ce ce que l'on sait au niveau des sciences cognitives et neuroscience concernant l'apprentissage comme présenté par [Dehaene-2012]. Du double travail de synthèse de Tricot et al, nous pouvons en extraire les éléments présentés ci dessus.

Margarida Romero, Stéphanie Noirpoudre et Thierry Viéville.

Modifier

← Nos newsletters

BrainyUP remixe Class´Code →

Suivez notre actualité sur Twitter ou Facebook !