

---

# QUELS APPRENTISSAGES FONDAMENTAUX POUR LE CODAGE INFORMATIQUE ? RÉFLEXIONS AUTOUR D'UNE PRATIQUE EN MATERNELLE.

---

**Thomas DE VITTORI**<sup>1</sup>

Univ. Artois, UR 2462, Laboratoire de Mathématiques de Lens (LML), F-62300, France

**Perrine PRIEM**<sup>2</sup>

Professeur des écoles, académie de Lille

**Résumé.** Dans cet article, nous proposons un exemple d'utilisation de robots éducatifs pour l'initiation au codage informatique dès le plus jeune âge. Dans un premier temps, nous présentons une séquence de classe complète avec le Blue-Bot réalisée à l'école maternelle. Nous y détaillons les différents supports et les procédures des élèves qui donnent à voir certains apprentissages fondamentaux de l'informatique. Ces premiers apprentissages comme la question de la syntaxe, des essais-erreurs, sont ensuite mis en parallèle de quelques pratiques professionnelles du codage. Pour cela, nous nous appuyons sur une petite enquête diffusée à des informaticiens en entreprise.

**Mots-clés.** Blue-Bot, maternelle, cycle 1, pratiques de codage

**Abstract.** In this paper, we offer an example of use of educational robots for the initiation to computer coding from a very young age. First, we present a complete teaching sequence with the Blue-Bot carried out in kindergarten. We detail the different materials and the procedures of the pupils which show some fundamental learning of computer science. These first learnings, such as syntax, trial and error, are then questioned in relation to some professional coding practices. For this, we rely on a small survey spread to computer scientists in companies.

**Keywords.** Blue-Bot, kindergarten, coding

**Resumen.** En este artículo ofrecemos un ejemplo del uso de robots educativos para la iniciación en la codificación informática desde edades tempranas. En primer lugar, presentamos una secuencia de clase completa con el Blue-Bot realizada en la preescolar. Detallamos los diferentes soportes y los procedimientos de los alumnos que muestran ciertos aprendizajes fundamentales de la informática. Estos primeros aprendizajes, como la cuestión de la sintaxis, la prueba y el error, se ponen en paralelo con algunas prácticas profesionales de codificación. Para ello, nos basamos en una pequeña encuesta distribuida a los informáticos de las empresas.

**Palabras clave.** Blue-Bot, preescolar, código informático

---

<sup>1</sup> [thomas.devittori@univ-lille.fr](mailto:thomas.devittori@univ-lille.fr)

<sup>2</sup> [perrine.priem@ac-lille.fr](mailto:perrine.priem@ac-lille.fr)

## Introduction

Fruits d'une volonté de mise en adéquation des apprentissages scolaires avec les enjeux scientifiques contemporains (Baron & Drot-Delange, 2016), les récentes modifications des programmes ont vu l'introduction du codage informatique à l'école. Principalement focalisée sur les cycles 2 et 3, cette initiation à la programmation repose, surtout en début d'apprentissage, sur l'utilisation de robots éducatifs comme le Bee-Bot, le Blue-Bot ou Thymio, par exemple. La progression suggérée dans le document d'accompagnement disponible sur Eduscol<sup>1</sup> invite, par ailleurs, à préparer le codage de déplacement et cette programmation de robots dès le cycle 1 par des activités de repérage (dessus, dessous, devant, derrière, etc.). Pourtant, les robots éducatifs comme le Bee-Bot ou le Blue-Bot sont parfaitement adaptés à des élèves jeunes (Bellegarde et al., 2021) et on peut donc s'intéresser à la manière dont ils peuvent être exploités dès le cycle 1 (Komis & Misirli, 2015 ; Béziat, 2018 ; Grugier, 2018). C'est cette exploration des compétences potentielles en codage des élèves de maternelles que nous allons présenter dans cet article. Dans une première partie, nous rendons compte d'une séquence avec le Blue-Bot menée en grande section. Nous tentons d'y montrer que les élèves, bien que très jeunes, sont en mesure de mettre en œuvre des pratiques de codage informatique fondamentales comme la notion de débogage ou de sous-programme dans le cadre de situations adaptées à leur niveau. Ces pratiques peuvent faire échos aux usages chez les codeurs de métier et ainsi, dans une deuxième partie, à l'appui d'une enquête réalisée dans une PME en informatique, nous proposons de rendre compte de quelques enjeux d'apprentissage de la programmation qui paraissent accessibles dès le cycle 1 (Leonard et al., 2021), en mettant en parallèle les pratiques professionnelles et celles de l'informatique scolaire.

## 1. Apprendre à coder en maternelle

La pratique de classe qui va être détaillée ci-après s'est déroulée avec un groupe de six élèves de grande section durant la période 4 de l'année scolaire 2021/2022. Il s'agit d'une séquence composée de cinq séances qui ont eu lieu lors de temps d'activités pédagogiques complémentaires (APC). L'objectif principal de cette séquence était d'apprendre à se déplacer sur un quadrillage en codant un itinéraire. Le codage d'un itinéraire est alors le support pour l'approche des notions d'instruction, de suite d'instructions, de syntaxe et d'organisation du code. Ce faisant, les élèves ont pu également travailler certaines compétences propres au cycle 1 comme « Relater une succession de lieux pour décrire un itinéraire, un déplacement », « Se repérer dans l'espace de la classe », ou encore « Distinguer sa droite et sa gauche ». On notera que certains prérequis sont nécessaires pour la réussite des élèves, notamment la bonne compréhension des notions de quadrillage et de cases qui doivent donc être travaillées en amont.

Le matériel sélectionné pour réaliser cette séquence est le robot Blue-Bot. Nous avons emprunté à la bibliothèque universitaire de l'INSPE d'Arras trois coffrets de robots Blue-Bot ainsi que des tapis de jeu adaptés. Les coffrets sont composés d'un robot Blue-Bot, de son chargeur, d'une barre de programmation et de cartes d'instructions. Du matériel supplémentaire est également prévu : un coffre aux trésors, des images séquentielles et des quadrillages (avec point de départ, point d'arrivée, et sur certains des obstacles).

---

<sup>1</sup> Document d'accompagnement *Initiation à la programmation aux cycles 2 et 3*  
<https://eduscol.education.fr/document/15409/download>

## 1.1. Analyse de la première séance

La première séance est une introduction à la notion d'algorithmes qui doit permettre aux élèves d'aborder le concept de déplacement sur un quadrillage. Cette séance est programmée sur un temps de 45 minutes et est divisée en trois phases. Une première phase vise à mobiliser la logique à partir d'images séquentielles, la seconde est une activité en salle de motricité, et la dernière est une phase de bilan avec les élèves.

Lors de la première phase, les élèves sont amenés à remettre en ordre des images séquentielles (Figure 1) et expliquer leurs choix. Cette activité permet de travailler sur la logique des événements et l'importance de l'ordre des étapes. Le contrôle de l'ordre proposé est de nature sémantique ; c'est la cohérence de l'histoire potentiellement racontée qui valide ou non la proposition. Dans le groupe, la majorité des élèves a réussi à remettre les images dans l'ordre et expliquer les choix. Seul un élève sur les six a mis les étiquettes dans le bon ordre mais de droite à gauche, ce qui ne constitue pas une erreur. Une explication sur le sens conventionnel de lecture a été rappelée pour le bon déroulement de la suite de l'activité.



Figure 1 : images séquentielles

Pour la deuxième phase, les élèves sont emmenés en salle de motricité, un espace où le sol est carrelé. Les élèves vont y travailler en binôme, un élève jouant le rôle d'un robot et un autre élève jouant le rôle de guide. La consigne rappelle que les instructions données doivent être précises et ordonnées à partir des seules expressions « en avant », « en arrière », « à gauche », « à droite ». Le but de l'exercice est de se déplacer d'un point de départ jusqu'au trésor, tous deux matérialisés sur le sol. Les élèves ont bien compris qu'il s'agit juste de faire un pas à droite, respectivement à gauche, sans se tourner. Du point de vue de l'élève-robot, le déplacement est relatif, car incarné, mais il est pensé en regard des instructions données par l'élève qui guide en fournissant un codage extérieur. Progressivement des contraintes sont ajoutées comme des détours ou les yeux fermés. La majorité des élèves-guides a réussi à bien anticiper les instructions, montrant ainsi leurs compétences en codage absolu du déplacement. Pour les groupes d'élèves n'ayant pas de difficultés, nous avons pu faire des chemins allant jusque dans un quadrillage de 15 par 15.

Le dernier temps de la séance est un bilan réalisé avec les élèves sur ce qu'ils ont appris. Les élèves y ont exprimé l'importance de donner des instructions précises pour que le robot puisse bien comprendre, et qu'il faut les donner dans l'ordre. De plus, ils ont pu repérer que plusieurs chemins différents permettent d'aller à un même endroit.

## 1.2. Analyse de la deuxième séance

Notre deuxième séance porte sur les déplacements sur un quadrillage et la création de lignes d'instructions. Il s'agit donc de passer d'une consigne orale à une consigne codée par un symbole. La rigueur dans la précision et dans l'ordre des différentes étapes s'inscrit pleinement dans la suite des activités de la première séance. Les élèves étant en forte réussite, nous avons

commencé la découverte des Blue-bot dès cette séance. Cette découverte du robot était, au départ, placée en séance 3.

La première partie de la séance consiste en la réalisation d'une suite d'instructions avec des cartes permettant à un pirate de rejoindre le trésor ce qui oblige de passer du meso-espace de la salle de motricité au micro-espace traditionnel de la feuille (Brousseau, 2000). Les élèves sont placés en binômes avec un élève qui réalise la suite d'instructions et un élève qui vérifie si le chemin mène bien au trésor. L'objectif de cette phase est de coder un itinéraire. Le matériel fournit comprend, pour l'élève qui réalise la suite d'instructions, une carte au trésor (Figure 2), un petit pirate et des cartes avec des flèches toutes identiques et orientable dans la direction qu'on souhaite (Figure 3), et pour l'élève qui teste le chemin, un petit pirate et une carte transparente avec uniquement la croix désignant le point de départ.

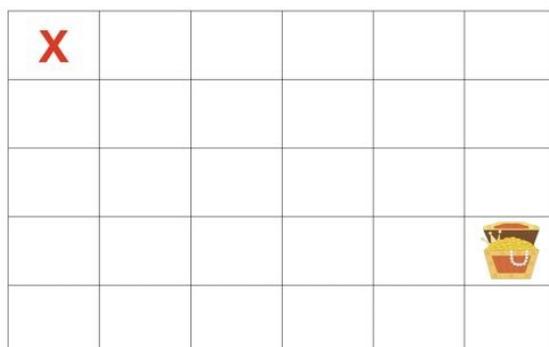


Figure 2 : La carte au trésor



Figure 3 : Flèche orientable et pirate

Si les élèves ont des difficultés, il leur est possible de déposer les cartes d'instructions sur le quadrillage pour matérialiser directement le chemin. Le deuxième élève du binôme (celui qui vérifie) possède un quadrillage transparent avec uniquement le point de départ. Seulement lorsqu'il a indiqué sur ce quadrillage où il pense que le trésor est caché, il peut superposer ce quadrillage avec celui du premier élève afin de vérifier (Figure 4). Plusieurs niveaux de cartes sont proposés aux élèves avec plus ou moins d'obstacles (Figure 5).

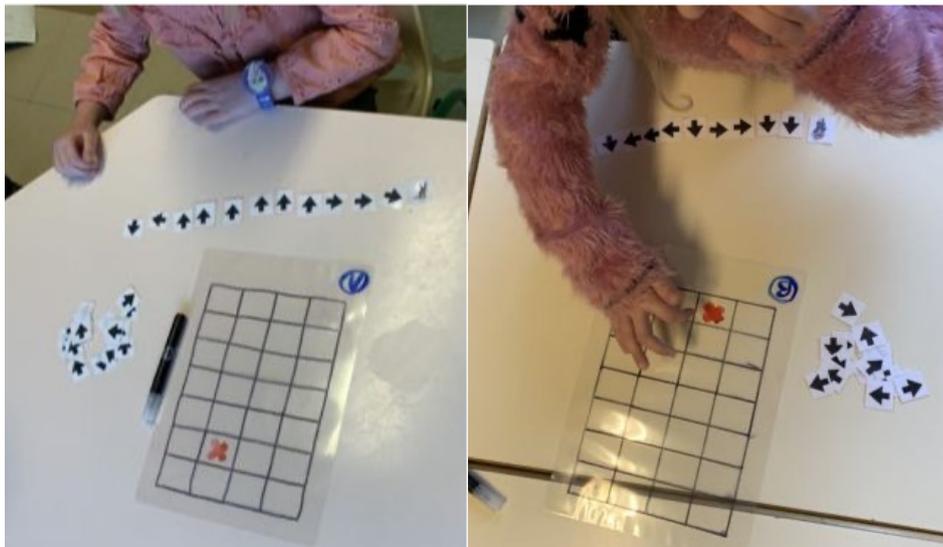


Figure 4 : Photographies d'élèves utilisant le quadrillage transparent

NIVEAU 1	NIVEAU 2	NIVEAU 3	NIVEAU 4

Figure 6 : Les différents niveaux de cartes aux trésors

Le passage de la situation de l'enfant-robot au codage sur papier a mis en évidence plusieurs difficultés quant au nombre d'instruction à donner. En effet, comme on peut le voir sur la figure 4, les élèves vont très souvent coder le nombre de cases du chemin, y compris le point de départ, et non les déplacements requis. Il a été nécessaire de reprendre plusieurs fois avec eux la notion de déplacement sur la grille, une flèche faisant passer d'une case à l'autre et le bord de la feuille donnant facilement à voir l'incohérence du code (le personnage sort de la feuille). Étonnamment, les élèves n'ont pas fait d'erreur sur l'orientation des flèches sans doute car, dans un premier temps, ils plaçaient leurs flèches spatialement en imitant le chemin pour aller aux trésors (Figure 6). A notre demande, les élèves sont passés de cette forme intuitive à des cartes instructions positionnées horizontalement, dans le sens de lecture usuel. Ce passage à une disposition plus compacte s'avérera indispensable pour la suite de l'activité lors du codage de chemins longs présentant des points de passage doubles. On peut remarquer que la contrainte d'un point de passage double aurait pu servir à construire avec les élèves la nécessité d'une mise en ligne du code. Le simple codage d'un aller-retour pouvait alors convenir, mais nous ne l'avons pas prévu dans cette séquence. Nous enchaînons donc avec la présentation du Blue-Bot.





Figure 7 : Blue-Bot et ses cartes de programmation

### 1.3. Analyse des séances 3 et 4

La découverte du Blue-Bot ayant eu lieu dans la séance précédente, la troisième et la quatrième séance sont des séances d'entraînement et de réinvestissement avec le robot. Durant la séance 3, les élèves se sont donnés des missions avec, par exemple, un élève qui plaçait le robot sur le tapis à l'endroit de son choix et un autre élève qui devait réaliser une suite d'instructions sur la barre de programmation afin d'atteindre un certain objectif (Figure 7). La vérification se fait en appuyant sur le bouton « Go » de la barre de programmation qui déclenche l'exécution du code. L'utilisation de la barre oblige ainsi à anticiper la position du robot à chaque étape, contrairement à la séance 2 où chaque déplacement était codé de manière autonome par le déplacement du jeton sur la grille. Dans les séances 3 et 4, le passage au robot oblige une nouvelle manière de penser et coder le déplacement. Cette activité permet aux élèves de se décentrer, de se mettre à la place du robot, affinant ainsi le passage d'un codage absolu du déplacement à un codage relatif au robot.



Figure 7 : Élèves en situation avec le matériel

Pour les élèves ayant des facilités, des déplacements comprenant des obstacles leur sont proposés. Par exemple sur le tapis comprenant une rivière, on oblige les élèves à passer par le pont. Cela a permis de réaliser des déplacements différents, notamment au niveau de la longueur du code. Pour certaines missions, les élèves ont ainsi dû utiliser deux barres de programmations afin de mener à bien le déplacement du robot, c'est-à-dire des codes avec plus de dix commandes.

### Remarques sur le codage via les boutons sur le Blue-Bot

En fin de séance 4, la programmation par les boutons sur le robot a été proposée. Après avoir expliqué que le robot enregistre tout ce qu'on lui commande (lorsqu'on clique sur les boutons) et que si on veut refaire un nouveau déplacement il faut lui dire d'oublier à l'aide de la touche « croix bleue », les élèves ont pu s'essayer à quelques codes. Cette tâche est bien plus difficile que les précédentes car, avec la seule utilisation des boutons, les élèves ne visualisent pas ce qu'ils ont codé ; ils perdent vite le fil et ne savent plus où ils en sont. Mettre en mémoire les déplacements codés sans pouvoir les voir oblige à mener une réflexion différente avec une plus grande anticipation du résultat, ce qui peut rapidement devenir inaccessible à un élève de cycle 1.

### 1.4. Dernière séance : de nouveaux apprentissages

La cinquième séance visait à proposer un réinvestissement des connaissances et des compétences relatives au codage de déplacement sur un quadrillage dans une situation complexe. À partir d'un support distribué (Figure 8), le but est de créer un chemin pour ramasser l'ensemble des œufs de la carte tout en évitant les barrières. Le départ se fait à partir de la croix rouge où le Blue-Bot est supposé orienté vers le haut de la grille. D'autres points de départ et d'autres orientations initiales du robot sont évidemment possibles, mais ça n'a pas été proposé aux élèves ici. Chaque enfant réalise sa suite d'instructions sur une bande de papier à partir de petites images imprimées reprenant la liste des commandes du Blue-Bot (même visuel que les cartes de codage). La situation impose volontairement un chemin très long (5 œufs à ramasser) afin d'entrer dans les problématiques liées à lisibilité d'une suite d'instructions longue.

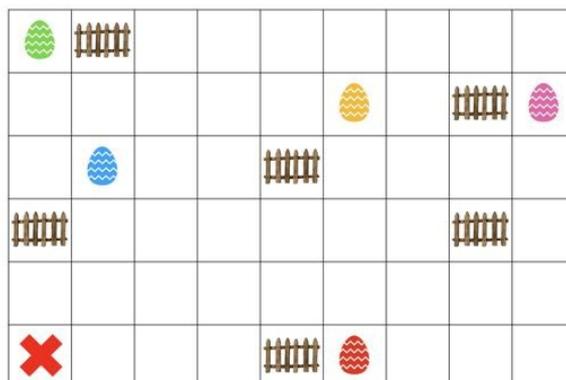


Figure 8 : La chasse aux œufs

Afin d'aider les élèves, des jetons correspondant au nombre d'œufs sur le plan leur étaient fournis pour qu'ils se souviennent des œufs qu'ils ont déjà pris (Figure 9).





d'itinéraires. Une telle programmation « sous contraintes » peut s'avérer très pertinente pour le respect d'une syntaxe (commandes « interdites »), et dans un travail sur la modification-corrrection, voire l'optimisation, d'un code.

## 2. Pratiques à l'école maternelle et pratiques professionnelles

Dans notre première partie, nous avons avant tout rendu compte d'une activité conforme à la fois au fonctionnement habituel en ateliers d'une classe de grande section et aux programmes dans une préparation au cycle 2. En ce sens, il s'agit d'une pratique de classe ordinaire dans laquelle les élèves de grande section ont pu s'essayer au codage, retrouvant ainsi certaines potentialités déjà relevées dans la littérature. Ainsi, comme l'indique Komis et Misirli (2011), le « jouet programmable » permet de développer des compétences en mathématiques, notamment la pensée algorithmique et la résolution de problèmes. Cette étude par Komis et Misirli (2011) de la programmation informatique à l'aide de robots montre également que, par les observations et les manipulations, on bouscule certaines représentations initiales des élèves, notamment sur les déplacements présumés du robot. Ces réflexions rejoignent les travaux de Grugier et Villemonteix (2017) qui ajoutent que les robots programmables peuvent aussi bien être pris en charge par des élèves de maternelle que de primaire, ce que nous avons effectivement pu tester dans la lignée des expérimentations de Komis et Misirli (2015) ou de Barrué (2017). Le point clé réside alors dans la construction de compétences pérennes en informatique. Or, comme le souligne Spach (2017, p.293), les activités comme celles réalisées avec le Blue-Bot « attestent de premières constructions d'une forme de la pensée informatique qu'il conviendrait de mettre à l'épreuve dans d'autres situations. » Nos activités n'apportent malheureusement pas d'éclairage sur cet aspect qui interroge aussi la proximité souhaitée entre un apprentissage scolaire de l'informatique et un apprentissage à visée professionnelle, au moins au niveau de certains fondements communs (Vandeveld & Fluckiger, 2022). Toutefois, notre séquence proposée en grande section a incontestablement donné l'occasion aux élèves de construire et de mobiliser les éléments d'une pensée informatique ici principalement séquentielle. Dans cette première approche, le choix de la bonne instruction et le respect de l'ordre sont cruciaux. En cela, il nous semble que ces éléments préparent effectivement le rôle de la syntaxe dans les langages informatiques plus élaborés. Concernant la structure des programmes, on l'a vu, l'explicitation de la notion de sous-programmes peut apparaître très tôt dans les apprentissages. Enfin, au niveau de l'exécution du code, l'exécution pas-à-pas et les allers-retours entre le code et ce qu'il produit réellement une fois exécuté contribuent au développement d'une première vision du débogage, étape indispensable dans le monde de l'informatique industrielle. Il y a incontestablement des parallèles entre la pratique scolaire et la pratique professionnelle du codage, et nous voudrions terminer en rendant compte d'un questionnaire que nous avons soumis à des codeurs professionnels d'une PME (7 codeurs ayant de 1 à plus de 10 ans d'expérience). Lors de la préparation et la mise en œuvre de la séquence avec le Blue-Bot, plusieurs aspects liés à la programmation nous ont interpellés. Le premier est l'approche par essais-erreurs qui est très souvent mise en avant dans le contexte scolaire (Spach, 2018). Comment est-elle perçue par ces professionnels ? Afin d'avoir un premier retour, nous avons proposé une suite de dix adjectifs<sup>1</sup> en demandant d'en choisir trois. Les réponses ont très largement souligné la dimension positive de l'erreur avec des adjectifs comme « intéressant », « normal » ou « constructif » loin devant

---

<sup>1</sup> La question était formulée ainsi : « Lorsqu'une erreur s'affiche, vous trouvez ça... Dans la liste ci-dessous, choisissez 3 adjectifs qui vous paraissent le mieux décrire votre sentiment. Normal, énervant, motivant, insupportable, excitant, intéressant, lassant, constructif, pénible, rageant. »

« pénible » ou « rageant ». Néanmoins, au travers d'autres questions, ces professionnels mettent aussi en avant l'importance de la vérification du code avant son exécution, c'est-à-dire qu'ils ne chercheront pas à attendre la rétroaction du bug pour agir, contrairement à une pratique scolaire qui nous paraît utiliser beaucoup plus cette modalité. Pour le monde de l'entreprise, même si elles sont intellectuellement intéressantes, les erreurs peuvent faire perdre beaucoup de temps et donc de la productivité, ce qui a un coût. La question de la productivité nous amène à une deuxième remarque que nous aimerions partager ici. Elle concerne la forme du code. Comme nous avons pu le voir dans les activités avec les élèves de maternelle, la forme du code devient très vite importante dès lors que ce dernier gagne en longueur. Ceci fait référence à la structuration d'un programme en sous-programmes, procédures, voire fonctions, mais pas seulement. Tous les programmeurs que nous avons interrogés accordent une place très importante à la mise en forme de code<sup>2</sup>. Ils utilisent, par ailleurs, des interfaces de programmation qui sont chargées de nombreux assistants. Dans les environnements de codage majoritairement utilisés<sup>3</sup> comme Visual Studio ou Notepad++, la coloration syntaxique est omniprésente tout comme l'indentation automatique du code ou la vérification des ouvertures/fermetures de parenthèses ou crochets. Ces aides visuelles font partie intégrante de la manière dont est présenté le code. Il s'agit là d'apports dont on peut expliciter l'importance avec des élèves dans une continuité des apprentissages (Tchounikine, 2016) en mathématiques (argumentation/démonstration, organisation d'une série de calculs, ...) mais aussi en production d'écrit en français dans lesquels une bonne présentation contribue à la lisibilité et donc à l'intelligibilité du message. Le travail sur la mise en forme du code nous paraît sous-estimé dans un contexte scolaire, en particulier lors des premiers apprentissages. Même si le passage à des outils numériques plus développés comme Scratch ou Python compense cette lacune, quelques activités spécifiques relatives à la structuration et l'organisation spatiale du code pourraient, sans doute, être développées.

## Conclusion

La codage en maternelle ouvre de nombreuses perspectives quant aux compétences que peuvent développer des élèves de cycle 1 relativement à l'outil informatique. Dans la séquence que nous avons présentée, les élèves ont très rapidement montré une capacité à prendre en main le Blue-Bot. Ce dernier ayant été conçu à cette fin, ce n'est pas forcément surprenant, mais au-delà du jeu avec le robot, les activités ont permis de travailler avec les élèves les notions d'instructions et de suites d'instructions, de logique, d'étapes intermédiaires dans une tâche complexe, etc. On peut ajouter que le codage offre aussi aux élèves un espace de liberté créative. Comme en mathématiques ou dans d'autres disciplines, mais de manière peut-être plus perceptible directement par l'élève, la réalisation d'un programme peut se faire de nombreuses bonnes manières. Dans la séquence avec le Blue-Bot, il n'y avait pas de chemin unique, et un même chemin pouvait être codé de plusieurs manières (avec ou sans la commande « Recule » par exemple). Le code a cette dimension profondément personnelle ; chacun ayant sa manière de faire. Cette richesse est un élément qui a une influence sur les pratiques de codage professionnelles. Ce n'était pas le propos ici, mais l'ajout de commentaires, par exemple, dans un programme est une pratique presque indispensable tant pour des questions de lisibilité que pour la maintenance ultérieure. À l'interface entre l'écriture en langage naturel et l'algorithme en

<sup>2</sup> À la question « Quelle importance accordez-vous à la mise en forme de votre code (saut de ligne, indentation, commentaires, etc.) ? », un score moyen de 4.6 a été obtenu sur une échelle de Likert de 1 (« Pas importante du tout ») à 5 (« Très importante »).

<sup>3</sup> Voir par exemple l'enquête auprès de plus 90000 codeurs réalisée par le site Stackoverflow <https://insights.stackoverflow.com/survey/2019>

langage informatique, une pratique scolaire pourrait s'en inspirer pour enrichir l'activité des élèves et leur compréhension. Commenter son code, c'est aussi mieux le comprendre. Dans l'univers professionnel, nous pouvons remarquer, par ailleurs, la place de plus en plus grande reprise par le langage naturel dans les pratiques de codage. En effet, parmi les assistants intégrés aux interfaces de travail des codeurs professionnels, les générateurs automatiques de code conçus à partir d'intelligences artificielles (Finnie-Ansley et al., 2022 ; Vaithilingam et al., 2022) sont devenus incontournables. Le moteur le plus connu actuellement est sans doute Copilot<sup>1</sup> qui, même dans la toute petite PME que nous avons contactée, est utilisé quotidiennement par plus de la moitié des programmeurs. À l'échelle mondiale, cet outil est adopté par une très large communauté qui y trouve le moyen d'augmenter sa productivité (Kalliamvakou, 2022 ; Ziegler, 2022) tant par la génération automatique de code à partir de requêtes en langage naturel que par des systèmes très performants de correction syntaxique et de complétion automatique de fonctions. Des évolutions qui auront sans doute un impact, elles-aussi, sur les pratiques scolaires et les enjeux d'apprentissage.

## Références bibliographiques

- Barrué, C. (2017). Les jouets programmables à l'école maternelle : Pratiques pédagogiques de professeurs stagiaires. *Grand N*, 99, 69-86.
- Baron, G.-L., & Drot-Delange, B. (2016). L'informatique comme objet d'enseignement à l'école primaire française ? Mise en perspective historique, *Revue française de pédagogie*, 195, 51-62.
- Bellegarde, K., Boyaval, J. & Alvarez, J. (2021). Initier des élèves de maternelle à la robotique/informatique : quand les supports médiateurs impactent la grammaire de l'agir enseignant. *Sticef*, 28:3, pp.13-38. <http://sticef.univ-lemans.fr/num/vol2021/Sticef-V28-3-final.pdf>
- Béziat, J. (2018). L'informatique dans les premières classes de l'école primaire. Quel enseignement pour quels apprentissages. In *Symposium « Enseignement et apprentissage de l'informatique à l'école primaire » au colloque du CRIFPE, 3-4 mai 2018, Montréal*.
- Brousseau, G. (2000). Les propriétés didactiques de la géométrie élémentaire ; l'étude de l'espace et de la géométrie, *Séminaire de Didactique des Mathématiques, Réthymnon, Grèce*, pp.67-83.
- Finnie-Ansley, J., Denny, P., Becker, B., Luxton-Reilly, A., & Prather, J. (2022). The Robots Are Coming: Exploring the Implications of OpenAI Codex on Introductory Programming. In *Australasian Computing Education Conference (ACE '22)*. Association for Computing Machinery, New York, NY, USA, 10–19. <https://doi.org/10.1145/3511861.3511863>
- Grugier, O. (2018). Des robots programmables à la maternelle. Un monde technologique à comprendre. In *3e édition des Rencontres nationales de la robotique éducative, « Robots, élèves, enseignants, quelle intelligence partagée ? », 2-3 octobre 2018 ; IFÉ-ENS de Lyon*.
- Grugier, O., & Villemonteix, F. (2017). Apprentissage de la programmation à l'école par l'intermédiaire de robots éducatifs. Des environnements technologiques à intégrer. *Apprentissage de la pensée informatique à EIAH 2017*, Strasbourg. Consulté le 5 février 2021, [https://wikis.univ-lille.fr/computational-teaching/media/wiki/actions/2017/aai-eiah/5-olivier-grugier-apimu\\_eiah17.pdf](https://wikis.univ-lille.fr/computational-teaching/media/wiki/actions/2017/aai-eiah/5-olivier-grugier-apimu_eiah17.pdf)

---

<sup>1</sup> <https://github.com/features/copilot>

- Kalliamvakou, E. (2022). Research: quantifying GitHub Copilot's impact on developer productivity and happiness. En ligne : <https://github.blog/2022-09-07-research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/> Consulté le 7 octobre 2022.
- Komis, V., & Misirli, A. (2011). Robotique pédagogique et concepts préliminaires de la programmation à l'école maternelle : Une étude de cas basée sur le jouet programmable Bee-Bot. *Actes du Colloque International DIDAPRO 4-Dida&Stic, 24-26 octobre 2011*, pp.271-281. Université de Patras, Grèce. <http://dx.doi.org/10.13140/2.1.3321.8569>
- Komis, V., & Misirli, A. (2015). Apprendre à programmer à l'école maternelle à l'aide de jouets programmables. In G.-L. Baron, É. Bruillard & B. Drot-Delange (dir.) *Informatique en éducation : perspectives curriculaires et didactiques*, pp.210-226. Clermont-Ferrand : Presses Universitaires Blaise-Pascal.
- Léonard, M., Peter, Y., Secq, Y., Alvarez, J., & Fluckiger, C. (2021). MOTIF..MOTIF.: initier à la notion de répétition en maternelle sans mobiliser de repérage spatial. *Sticef*, 28(3), pp.13-38. <http://sticef.univ-lemans.fr/num/vol2021/Sticef-V28-3-final.pdf>
- Spach, M. (2017). *Activités robotiques à l'école primaire et apprentissage de concepts informatiques : quelle place du scénario pédagogique ? Les limites du co-apprentissage*. Thèse de doctorat, Université Sorbonne Paris Cité. <https://tel.archives-ouvertes.fr/tel-02271924>
- Spach, M. (2018). Robotique pédagogie à l'école primaire, découverte de la touche « pause », des possibilités intéressantes. *Adjectif.net* Mis en ligne mercredi 20 juin 2018. En ligne : <http://www.adjectifs.net/spip/spip.php?article470>
- Tchounikine, P. (2016). *Initier les élèves à la pensée informatique et à la programmation avec scratch*, Université Grenoble-Alpes. En ligne : <http://lig-membres.imag.fr/tchounikine/PenseeInformatiqueEcole.pdf>
- Vaithilingam, P., Tianyi Zhang, T., & Glassman, E. (2022). Expectation vs. Experience: Evaluating the Usability of Code Generation Tools Powered by Large Language Models. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI '22 Extended Abstracts)*, April 29-May 5, 2022, New Orleans, LA, USA. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3491101.3519665>
- Vandeveld, I., Fluckiger, C., & Nogry, S. (2022). Resources and textbooks for computer science education in French primary schools. *IARTEM e-journal*, 2022, 14(1)
- Ziegler, A. (2022). Research: How GitHub Copilot helps improve developer productivity. En ligne : <https://github.blog/2022-07-14-research-how-github-copilot-helps-improve-developer-productivity/> Consulté le 7 octobre 2022.