

Rubrique multimédia (version numérique) de Repères-IREM n°141

Gérard Kuntz (g.kuntz@libertysurf.fr)

La rubrique multimédia est confiée dans ce numéro 141 de Repères-IREM à [Marc Agenis-Nevers](#), concepteur et créateur de l'outil d'apprentissage débranché de l'algorithmique **Code en Bois**, objet de cet article.

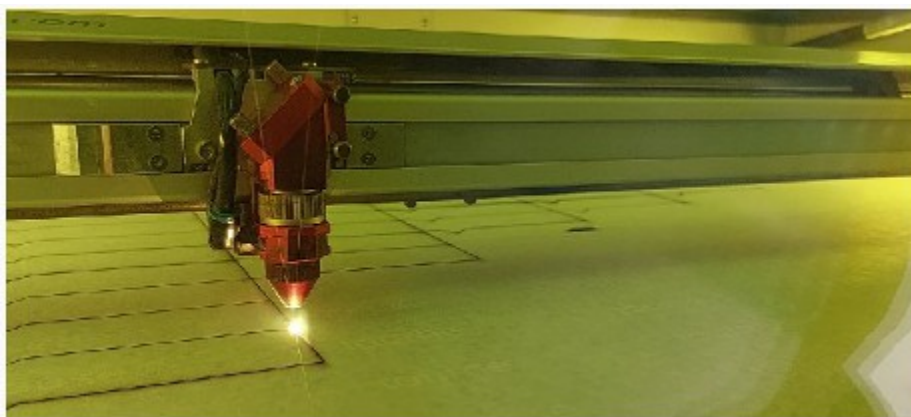
Code en Bois, un outil pour travailler la pensée informatique en mode débranché

Marc Agenis-Nevers (marc@codeenbois.fr)

« Mais, Madame, on ne va pas en salle informatique? » s'interroge un élève de 6ème un peu désespéré, apprenant qu'il allait assister à un atelier de programmation. Il examine alors sur sa table le tas de briques en bois colorées, lève les yeux sur une sorte de labyrinthe dessiné en noir sur le tableau, et comprend, avec sur son visage un mélange de surprise et de curiosité, que l'activité prévue a bel et bien quelque chose à voir avec le cours de Scratch qu'il avait eu en fin de primaire.

Mais ce n'est pas le cas de tous les élèves de sa classe, les initiations à la « pensée informatique » sont encore assez hétérogènes selon les établissements. L'apparition du vocable « pensée informatique » dans les nouveaux programmes est d'ailleurs une bonne nouvelle, et entérine la large utilisation de ce concept dans la littérature scientifique à partir des années 2010. Le terme est mentionné dans le rapport Villani-Torossian (2018) pour arriver depuis l'année dernière dans les programmes de mathématiques (cycle 3 et projet de cycle 4 pour 2026). Les enseignants sont d'ailleurs encouragés à regarder sous l'angle « algorithmique » des exercices de divers domaines, ainsi qu'à débiter ces apprentissages par des activités dites « débranchées » (c'est à dire n'utilisant que des objets simples et non électriques : papier, crayon, ciseaux, bois...). C'est précisément l'objectif du matériel *Code en Bois*, qui vise à former les élèves aux grands concepts de l'algorithmique au moyen de pièces en bois semblables aux briques élémentaires de Scratch (ou Blockly). Sur l'informatique débranchée, la littérature ne manque pas pour expliquer l'intérêt pédagogique, mais également l'intérêt en termes de CPS ([compétences psychosociales](#)).

La naissance de l'outil *Code en Bois* remonte à fin 2021, avec la proposition d'une enseignante de CE2 à Clermont-Ferrand, de faire présenter leurs métiers aux parents d'élèves de la classe (c'était mon cas, [j'ai expliqué mon parcours professionnel atypique...](#)) lors d'une pause méridienne. L'idée de proposer une version de Scratch sur des blocs rigides en bois m'est alors venue assez naturellement. C'est seulement ensuite que je me suis posé la question « ce genre de chose doit sûrement déjà exister », la réponse était non, pas vraiment, puisqu'aucune initiative suffisamment poussée pour proposer un produit fini n'existait alors. Avec un collègue enseignant de physique et féru de Python, nous avons passé de longs moments à décider ce qui ferait partie d'un ensemble d'instructions et d'actions de base. Le logiciel Scratch, dans sa version minimale (sans les extensions), compte près de 110 briques d'instructions différentes. Or il n'était pas question d'avoir plus d'une trentaine de morceaux de bois sur une table de classe, ni des programmes trop longs qui déborderaient vite vers le bas pour finir par terre ! De fait, les contraintes créées par la transposition d'un outil 100% numérique vers un objet purement physique - et en bois - étaient nombreuses. Exemple: le bois ne s'étirant pas, comment créer un bloc qui puisse contenir un nombre variable d'autres blocs ? Nous avons introduit des pièces « fin de bloc » qui rendent le code proche d'un vrai langage de programmation avec indentation. Un long tâtonnement technologique pour fabriquer efficacement de telles briques en masse avec une production française a abouti début 2023 à une combinaison d'impression sur bois et de découpe laser : il faut imaginer un faisceau de 130 Watts concentré sur à peine 1mm² de surface, qui va vaporiser jusqu'à 1 cm d'épaisseur de bois comme si c'était du beurre. D'ailleurs, la présence de petites imprimantes laser dans de nombreux fablabs de collège permet d'imaginer demain des projets de fabrication « maison » de briques de code ! Le matériel a régulièrement évolué au cours des 2 dernières années avec l'apparition de pictogrammes pour les non-lecteurs et d'une gamme de tailles adaptées aux situations. [L'expertise de l'IREM de Clermont-Ferrand](#) a aussi été très utile pour orienter ou écarter certains choix ([voir l'annexe 2 en fin de document](#)).



La découpe laser de plaques de bois pour les pièces Code en Bois



Les pièces Code en Bois finies

Les kits scolaires sont communs aux cycles 2, 3 et 4, c'est-à-dire que c'est le même matériel qui pourra accompagner l'élève pendant plusieurs années. Certaines pièces ne serviront donc pas avant le cycle 3, et en fin de collège on pourra s'aider d'une extension spécialement dédiée à la découverte des variables et des routines. Ainsi, un même kit pourra être partagé entre plusieurs niveaux, ou entre un primaire et un collège (cités éducatives, groupes scolaires) et servir un projet de liaison école-collège.

- Se déplacer et décrire des déplacements dans la classe en s'orientant et en utilisant des repères. - Construire et utiliser un plan de la classe pour communiquer un déplacement. - Utiliser et produire une suite d'instructions qui codent un déplacement en utilisant un vocabulaire spatial précis.	L'élève comprend et utilise les instructions suivantes : avancer, reculer, tourner à droite, tourner à gauche, monter, descendre. L'élève sait représenter sur un plan de la classe un itinéraire qu'il a effectué. L'élève sait coder un déplacement qu'un autre élève doit ensuite effectuer, par exemple : « avancer de deux pas, tourner à droite, reculer de trois pas ». Si un robot est disponible, l'élève peut programmer son déplacement sur un tapis quadrillé. Pour coder ces déplacements, il utilise les instructions : « avancer d'une case », « pivoter d'un quart de tour à droite », « pivoter d'un quart de tour à gauche ». Les déplacements à programmer comprennent au maximum dix instructions, dont deux virages.
---	--

(programme 2025 cycle 2 mathématiques)

Effectuer une boucle simple avec une répétition.	L'élève sait que les boucles permettent de répéter une ou plusieurs instructions plusieurs fois sans les réécrire. Il sait répéter une séquence linéaire d'instructions un nombre précis de fois en la plaçant dans une boucle inconditionnelle simple.
---	---

(projet de programme 2026 de mathématiques, 5ème)

Utiliser une boucle conditionnelle.

L'élève définit des boucles conditionnelles.

L'élève définit et utilise un bloc personnalisé pour structurer un programme.

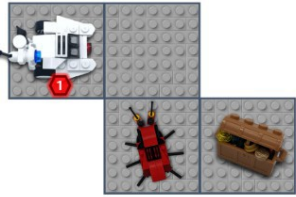
L'élève utilise un opérateur logique pour combiner deux conditions simples.

(projet programme 2026 de mathématiques, 3ème) + Extension Code en Bois variables & fonctions

Mission 6 : Zig zag

Situation déclenchante : On va faire une petite variante pour vérifier que tout est bien compris avec les PIVOTE.

Objectif : Récupère le trésor
On remet le petit symbole de la batterie du laser pour indiquer qu'un seul tir est autorisé.



Familles de briques nécessaires : bleues

Nombre de lignes minimum de la solution : D + 7 lignes

Solution





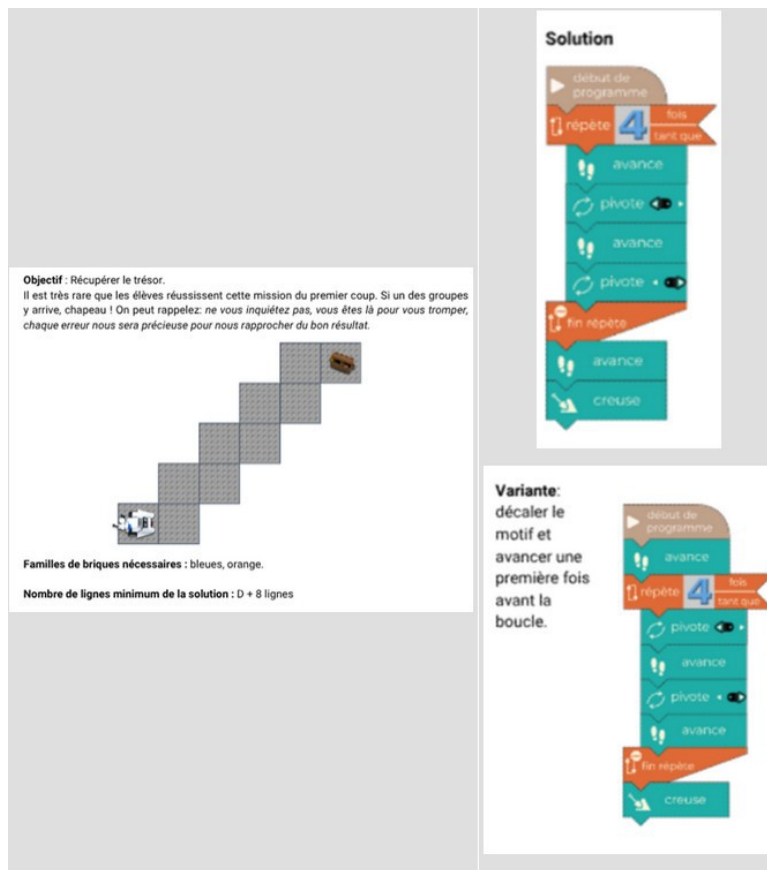
(À gauche le livret enseignant niveau cycle 2, à droite l'extension pour cycle 4 !)

L'exemple de la mission (ou activité) 12 [du livret enseignant](#)

Prenons à titre d'exemple ci-dessous *la mission 12 du livret enseignant* (le livret de 81 pages est en accès libre, [on peut le feuilleter ici](#)), qui est particulièrement féconde en enseignements et accessible dès le CE2 : l'objectif est d'amener le robot jusqu'au trésor en parcourant cette série de zigzags qui requièrent tantôt de pivoter à gauche, tantôt à droite.

Puisqu'on dispose de seulement deux briques PIVOTE et quatre AVANCE, il est absolument nécessaire de les utiliser dans une boucle. Au moins deux solutions possibles existent, et les élèves vont souvent faire plusieurs erreurs qu'ils verront en essayant au tableau : ils proposent par exemple le motif AVANCE+PIVOTE À GAUCHE + PIVOTE À DROITE, qui échoue, car les deux rotations s'annulent et le robot continue tout droit.

Certains voudront mettre un 9 comme nombre d'itérations, ayant compté 9 cases à parcourir, oubliant que chaque répétition comporte 2 AVANCE ! Après plusieurs essais, la classe comprend les erreurs possibles et comment les éviter. C'est enfin un défi propice également à l'introduction des blocs personnalisés en 3e (qu'on appelle routines) et à l'introduction aux boucles conditionnelles (4e) si on fait varier le nombre de virages.



À gauche, le dessin du parcours à reproduire au tableau avec les magnets. À droite : deux solutions possibles avec Code en Bois.

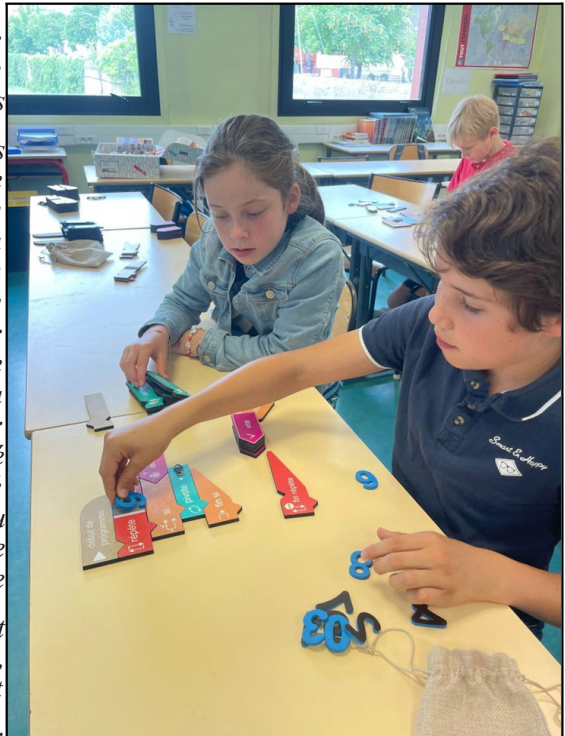
Témoignage de Mme Estève, enseignante de mathématiques dans l'académie de Clermont-Ferrand : « J'ai contacté Code en Bois pour mettre en place une expérimentation dans mon collège. Mon idée première était de faire des activités lors de la semaine des mathématiques en mars 2023. Finalement j'ai intégré les

activités directement dans mon cours de 4^{ème}. Je suis dans un établissement où les élèves sont équipés d'iPad, aussi je trouvais intéressant de travailler l'algorithmique sans numérique. Les élèves ont tout de suite bien accroché à l'univers du robot et du trésor. Les règles étaient simples à expliquer et à mettre en place. Même si tous avaient des notions des années antérieures sur Scratch, le passage sur code en bois a permis de souligner des défauts de maîtrise des instructions conditionnelles et des boucles, et on a pu ainsi mettre en place de la remédiation, insister sur l'importance du positionnement des briques dans le déroulé d'un programme. Le retour sur Scratch par la suite a été plus aisé. L'année suivante, la mise en place en cours n'a pu se faire, mais j'ai pu expérimenter une autre forme d'utilisation à l'occasion du départ en voyage scolaire

d'une grande partie des 4^{ème}. Tous les élèves restants ont été regroupés, et plutôt que de faire des révisions classiques, j'ai mis en place les activités Code en Bois avec pour but

sur la dernière séance, que les 4^{ème} encadrent à leur tour une classe de CE1-CE2. Cette expérimentation a été une jolie découverte sur plein de points, tant sur le

plan mathématique que sur le plan relationnel. Les 4^{ème} ont très bien su faire découvrir la programmation aux plus jeunes, ils se sont mieux approprié des notions car ils devaient à leur tour les transmettre aux plus jeunes. Désormais j'utilise code en bois en classe de 4^{ème} pour amorcer la programmation, en 3^{ème} plutôt en remédiation. Je continue à intervenir dans la classe de primaire d'une



collègue. Nous faisons des séances d'initiation en demi-groupe, puis en classe entière, nous proposons aux élèves (mis en binômes) de créer des énigmes à leur tour. Le projet de programme de Cycle 4 pour la rentrée 2026, imposera une progression de l'introduction des notions, en fonction du niveau de classe. Peut-être rajouterai-je des énigmes dédiées à mes niveaux de classe, afin de ne pas anticiper les notions de l'année suivante, mais Code en Bois fera encore partie intégrante de mon enseignement de la programmation! »

En termes **pédagogiques**, un atelier d'algorithmique débranchée est assez différent d'un atelier sur ordinateur : lorsqu'on appuie sur « GO » dans Scratch, on ne sait pas ce qui se passe, et toute la partie importante de l'exécution va se dérouler en une fraction de seconde sans que l'élève s'aperçoive de rien. Sur du code débranché, il va devoir faire ce travail pas à pas. Comme il faut un élève pour lire le code et un autre pour l'exécuter au tableau avec les personnages aimantés, les instructions vont devoir être transmises oralement et cet exercice d'élocution participe à lever toute éventuelle ambiguïté dans le code : en effet, le résultat est très différent si l'on dit « répète 5 fois les actions avance et creuse », ou « répète 5 fois avance, répète 5 fois creuse ». On préconise d'ailleurs de faire lire un code par un autre élève que celui qui l'a écrit.



Les autres groupes ainsi que l'enseignant observent et surveillent la cohérence et l'absence d'erreur. Si l'erreur est détectée par celui qui lit (« répète... ah tu as oublié de mettre le chiffre de la répétition! ») : c'est une erreur de syntaxe. Si l'erreur est détectée par celui qui est au tableau (ex : « impossible je n'ai plus assez de munitions pour tirer! ») c'est une erreur d'exécution. Les élèves comprennent vite qu'un *bug* est une erreur (presque toujours) humaine. Mais qu'une erreur est toujours une source d'apprentissage et un passage obligé, selon la tradition informatique bien connue *qu'un code ne fonctionne jamais du premier coup !*

L'**attribution de rôles** tournants à certains élèves vient étoffer cette implication de toute la classe : on se propose de nommer un responsable du compteur des répétitions (vérifier qu'on ne répète pas 6 fois quand on mis un 5 sur la boucle), et responsable de chaque capteur : l'élève capteur de trésor doit, lorsqu'il est interrogé sur la mesure de présence d'un trésor sur la case, répondre par VRAI ou FAUX.

L'utilisation de briques de code, en particulier celles adaptées pour aimantation au tableau et avec des trous sous la forme « ? », sont pertinentes pour travailler en pédagogie **PRIMM** (**p**redict, **r**un, **i**nvestigate, **m**odify, **m**ake), qui permet de concevoir des situations d'enseignement dans lesquelles les élèves s'engagent de manière progressive dans leur compréhension du code.

Ainsi, on peut commencer par présenter un code correct au tableau et l'exécuter, puis placer une briques mystère « ? » en demandant de trouver l'action manquante. On peut également mélanger les briques ou les blocs pour que les élèves retrouvent le bon ordre des instructions, plutôt que de partir d'un programme totalement vierge ce qui peut créer des situations d'échec.

L'organisation d'activités avec Code en Bois peut s'intégrer dans des séquences à raison de séances de 40 minutes à 1h30, deux ou trois fois **dans** l'année. L'enseignant peut suivre le livret qui propose une progression dans les défis par grande thématique (séquences d'actions et déplacements dans le plan, boucles et répétitions, capteurs et ~~m~~esures, structures conditionnelles). Les activités seront plus ou moins dirigées :

- Soit un travail **en classe entière synchrone** : l'enseignant présente une mission de son livret (= un labyrinthe avec un objectif à atteindre), laisse les élèves réfléchir et commencer à coder, puis fait passer 3 binômes successivement au tableau pour tester leur programme. La correction se fait **ensemble**, avec présentation d'une solution au tableau au moyen des briques géantes aimantées, puis avec un débrief sur les concepts appris.

- Soit un travail **en îlots en autonomie** : l'enseignant a préparé des fiches mission (par exemple en imprimant certaines missions du carnet, ou en les recréant à sa guise à l'aide des slides), et les distribue plastifiées aux groupes, en fournissant des personnages à déplacer. Les solutions peuvent être proposées sur le bureau de l'enseignant pour vérification.

L'**évaluation** peut se faire de façon classique, via des fiches conçues par l'enseignant, ou bien mobiliser des outils comme Plickers ou QCMCam (open source) que nous mettons à disposition pour collecter les réponses des élèves sur des questions relatives à la séance. [Les élèves répondent en montrant un QR-code tourné dans le sens adéquat correspond à l'une des réponses A, B, C ou D.](#)

Conclusion et avenir

Code en Bois est un outil polyvalent et facile à prendre en main pour s'initier à la programmation débranchée, du cycle 2 au cycle 4. Utilisable dans de nombreuses situations, avec des profils d'élèves très variés (une adaptation Braille est même disponible pour les non-voyants), et pourquoi pas en DNL (version anglaise) ou en classe dehors (jeu de l'enfant-robot).

Annexe 1 : exemples de mini-dialogues :

[*Élève-Programmeur*] Tu tournes !

[*Enseignant*] Il faut dire vers où!

[*Élève-Programmeur*] Ah, ben, j'ai mis... vers la droite!

[*Élève au tableau*] Euh.. non ! Vers la droite du robot c'est comme ça !

[*Enseignant*] Imagine que tu es dans le cockpit du robot, tu tournes le volant dans quel sens ?

[*Élève-Programmeur*] Répète tant qu'il y a un ennemi : Tire et tourne ! Bah vas-y, tu attends quoi ?

[*Élève au tableau*] Le capteur d'ennemi a pas fait « BIP », je peux pas répéter.

[*Élève-Programmeur*] Non mais il reste des ennemis là bas !

[*Enseignant*] Le capteur a raison, il sont derrière, alors il ne peut pas les détecter, la boucle s'arrête.

Annexe 2 : Apport de l'IREM de Clermont-Ferrand dans l'élaboration de Code en Bois

[L'IREM de Clermont-Ferrand](#), et en particulier le [groupe Informatique sans Ordinateur](#), coordonné par Malika More, a contribué à tester et orienter certains choix de conception. Par exemple, ils nous ont convaincus d'ajouter une brique « début de programme » et de remplacer les actions « tourne » par des « pivote » (moins ambigus). Par la suite, notre extension « variables&fonctions » a été soumise aux membres du groupe dès sa conception. Le groupe nous a aidé à sélectionner les pièces à retenir dans l'extension et leur nom :

- suppression du registre des variables en bois au profit d'un tableau de variables à remplir au feutre ;
- ajout d'une brique « diminuer la variable de (...) », plus facile que de faire « augmenter la variable » avec une valeur négative qui aurait été pénible en termes didactiques ;
- enfin l'idée d'introduire une distinction entre les variables modifiables et les variables systèmes en lecture seule, distinction opérée par une encoche-détrompeur qui empêche mécaniquement d'insérer les jetons variables dans les briques « augmenter » par exemple.

Nous sommes heureux d'avoir eu accès à ce type d'expertise précieuse dans le développement de nos outils pédagogiques.

Remarque finale : la présence de petites imprimantes laser dans de nombreux fablabs de collège permet d'imaginer demain des projets de fabrication « maison » de briques de code ! Le matériel a régulièrement évolué au cours des 2 dernières années avec l'apparition de pictogrammes pour les non-lecteurs et d'une gamme de tailles adaptées aux situations.

Annexe 3 : Des outils alternatifs :

Les lecteurs sont invités à consulter d'autres articles qui traitent de pensée algorithmique dans un cadre d'informatique débranchée, comme « [Algorithmes et codage au collège](#) » ou « Comportements algorithmiques, [I universalité](#) et [II verbalisation](#) », et plus récemment le [projet Expire](#). Ces articles, ainsi que d'autres qui y sont cités, décrivent l'état de la réflexion didactique sur l'initiation à l'informatique. Ils soulignent (comme le présent article) l'importance de la verbalisation et donnent de nombreux exemples d'activités débranchées qui ne nécessitent aucun investissement matériel (en ces temps de disette financière...).