

At 36 : Informatique débranchée : construire sa pensée informatique sans ordinateur

Sylvie Alayrangues¹, Samuel Peltier² et Laurent Signac³

¹² Université de Poitiers, XLIM ; sylvie.alayrangues@univ-poitiers.fr, amuel.peltier@univ-poitiers.fr

³ ENSI Poitiers ; laurent.signac@univ-poitiers.fr

Résumé : L'informatique débranchée, appelée également et peut-être plus justement, informatique sans ordinateur ou sciences manuelles du numérique, est une approche qui consiste à appréhender certains éléments de la science informatique par l'utilisation d'objets « concrets » et complètement « déconnectés » (bâtonnets, allumettes, cartes, jetons, ficelles, perle...). Elle permet de s'affranchir de la machine et de la technicité de sa programmation pour mieux saisir les grands principes de la science elle-même. Grâce à elle, il est possible d'aborder de manière ludique des problèmes complexes (recherche d'informations, tri de données, cryptographie, stratégie gagnante...) et leur résolution via la conception d'algorithmes. Elle favorise la mise en œuvre d'une démarche scientifique avec les enfants et adolescents. Elle est propice au travail de groupe. Enfin, lors des activités, il s'agit non pas de donner une solution clé en main mais d'amener les participants à imaginer leur propre solution ou à la co-construire ensemble.

Mots clés : informatique débranchée, science informatique, médiation scientifique, ateliers d'initiation

Introduction

L'informatique, science et technique (Conseil scientifique de la Société informatique de France (SIF), 2014), prend une place de plus en plus importante dans les programmes du primaire à la terminale mais, à défaut d'être reconnue comme discipline au sens scolaire, elle est intégrée à d'autres enseignements (mathématiques, technologie...), ou, lorsqu'elle fait l'objet d'enseignements spécifiques (Informatique et Création Numérique (ICN), Informatique et Sciences du Numérique (ISN)), elle reste enseignée par des professeurs d'autres disciplines. Aussi, une des difficultés auxquelles se heurte cette introduction massive de l'informatique dans le système scolaire est l'absence de formation à cette discipline de la plupart des enseignants dans leur cursus antérieur et la difficulté de former des cohortes entières d'enseignants de tous niveaux et toutes disciplines à la fois à la science informatique et à sa didactique.

Plusieurs approches existent pour faciliter cette acculturation nécessaire à la science informatique. Ainsi, par exemple, il est proposé aux enseignants de l'école primaire d'aborder l'informatique par le biais de la programmation créative qui permet notamment d'inventer des jeux, de raconter des histoires et ainsi de faire travailler aux enfants plusieurs compétences simultanément (Romero, 2016; Romero and Vallierand, 2016). La robotique ludique (Noirpoudre et al., 2017) est également un point d'entrée qui, outre la dimension créative, comporte également un aspect plus technologique et permet de programmer un objet qui réagit à son environnement. Un pas de plus vers le côté technique est franchi avec la démarche des « makers » qui associe conception électronique et développement informatique (Anderson, 2013). Ce courant se traduit notamment par la multiplication de lieux favorisant ces activités, tels les « Fab'lab » (Réseau français des Fab'lab, 2015))

L'informatique débranchée, également appelée informatique sans ordinateur (Di Cosmo, 2015) ou sciences manuelles du numérique (Quinson) est une autre porte d'entrée vers la pensée informatique qui peut être utilisée à tout âge et dans des contextes très divers. Elle présente l'avantage d'aborder l'informatique sans artefact technologique et se situe ainsi davantage au niveau de l'apprentissage des concepts, de ce que d'aucuns nomment la « pensée informatique » (Wing, 2009).

Dans un premier temps, nous présentons les 4 piliers qui fondent la science informatique et expliquons comment des activités sans ordinateur permettent de s'y familiariser. Nous partageons ensuite les éléments à prendre en compte pour construire un tel atelier et nous illustrons ceci sur l'exemple d'ateliers autour de l'algorithmique. Nous poursuivons par quelques retours d'expériences partagés par des collègues du secondaire. Enfin, nous concluons avec des perspectives.

Qu'est-ce que l'informatique ?

Piliers de l'informatique

Plutôt que de partir d'une définition de la science informatique comme celle proposée par la Société informatique de France (Conseil scientifique de la Société informatique de France (SIF), 2014), nous adoptons ici le point de vue de l'Académie des sciences, qui, dans un rapport publié en 2013 (Académie des sciences, 2013), met en avant ce qu'elle définit comme les quatre piliers de la science informatique : algorithmes, données / information, langages, machines.

En quelques mots, un **algorithme** est une méthode qui permet de résoudre un problème de manière systématique. Les **données** appelées à être manipulées par un ordinateur permettent de représenter des informations de nature très diverses : des images, des sons, du texte, des nombres. Pour ce faire, elles doivent d'abord être codées de manière appropriée pour pouvoir être traitées.

Avant l'ordinateur, **machine** « universelle » imaginée par Turing, les machines existantes ne permettaient de traiter qu'un type d'algorithme sur un jeu de données de nature unique. Parmi ces machines, on peut citer le métier à tisser Jacquart, et les bien plus anciens bouliers pour calculer. Au 20^{ème} siècle, est donc apparue la notion de machine « universelle », machine qui peut exécuter, *n'importe quel algorithme traitant n'importe quel type d'information* tant qu'algorithmes et informations sont codés de manières adéquates. Ces ordinateurs sont maintenant partout et de toutes tailles : des grands calculateurs aux objets connectés (télévision, grille-pain, etc.) en passant par les ordinateurs de bureau, les ordinateurs portables, les tablettes et les Smartphones. Concrètement, ces machines ont deux composants principaux leur permettant de fonctionner : un processeur (autrement dit une unité de calcul) et une mémoire. Ces deux éléments centraux sont bien entendu couplés à des mécanismes d'entrée / sortie permettant de fournir les données d'entrée du problème et de récupérer les résultats après traitement.

Enfin, pour implanter les algorithmes, généralement exprimés en langage naturel, donc ambigu, il est nécessaire de disposer de **langages** de programmation précis et sans ambiguïté.

Activités débranchées

« L'informatique n'est pas plus la science des ordinateurs que l'astronomie n'est celle des télescopes. » disaient Michael R. Fellows et Ian Parberry (Fellows and Parberry, 1993). En effet, si, pour s'incarner, l'informatique a besoin de la machine, les principes qui sous-tendent cette science peuvent se penser sans elle.

Le courant actuel des activités débranchées prend sa source dans la publication d'un cahier de telles activités, publié par Tim Bell (Bell, 2015) en Nouvelle-Zélande, même si des travaux avaient déjà été menés en France à la fin des années 80 (Crozer and Groperrin, 1987). En France, le mouvement s'est récemment développé à la fois dans le contexte de la médiation scientifique et dans celui de l'enseignement scolaire. Dans le premier cas, les activités sont souvent imaginées par des enseignants-chercheurs en informatique pour partager des grains de sciences, lors de la fête de la science, de visites de laboratoires, d'actions ponctuelles dans le cadre scolaire ou périscolaire (Duflot-Kremer; Quinson). Elles le sont aussi parfois par des associations qui développent des activités de médiation scientifique (par exemple, la fondation *la main à la pâte*). Davantage tournés vers l'enseignement, plusieurs groupes des IREM, notamment ceux de Clermont-Ferrand et Grenoble, créent et mettent à disposition des supports pédagogiques pour réaliser de telles activités en classe (Groupe IREM Grenoble, 2017; IREM Clermont-Ferrand). Un groupe de travail hébergé par la Société informatique de France, le groupe InfoSansOrdi (SIF), est un lieu d'échanges pour toutes celles et tous ceux qui s'investissent dans la conception de telles activités quelle que soit leur provenance.

Ainsi des notions liées à chacun des piliers évoqués précédemment peuvent être illustrées par le biais d'activités sans ordinateur, dont voici quelques exemples (la liste est très loin d'être exhaustive et si, pour chaque activité, une référence vers un fiche activité est fournie, il ne s'agit pas forcément de la version originale). Il est à noter qu'une même activité peut permettre d'aborder plusieurs piliers. Le découpage choisi ici vise juste à donner quelques exemples concrets de problématiques abordées par l'informatique sans ordinateur. Pour les activités citées, des liens sont donnés vers des pages web proposant des descriptions plus précises.

La construction de dessins, à la main, en suivant la philosophie de la tortue logo (Alayrangues et al.), autrement dit en utilisant un jeu d'instructions limité (tourne, avance, dessine) représentées par des cartes, se prête bien à une découverte de la notion d'algorithme (**Figure 1**). Une autre approche passe par l'analyse d'un jeu et la formalisation de stratégies gagnantes comme le jeu de Nim, ou les haricots de Dijkstra (Alayrangues et al.). Une approche plus physique, autrement dit qui nécessite que les participants se déplacent, peut être mise en œuvre pour illustrer certains algorithmes de tri (Algo-rythmics). Il est également possible d'aborder des notions plus avancées, en réfléchissant aux caractéristiques des algorithmes trouvés pour résoudre un problème. Comment savoir s'ils sont corrects ? Existe-t-il une solution unique ? S'il existe plusieurs solutions, l'une d'elles est-elle meilleure que les autres ? Ces interrogations sont notamment présentes dans les activités « Alice déménage » (Duflot-Kremer) et le « base-ball multicolore » (Quinson).

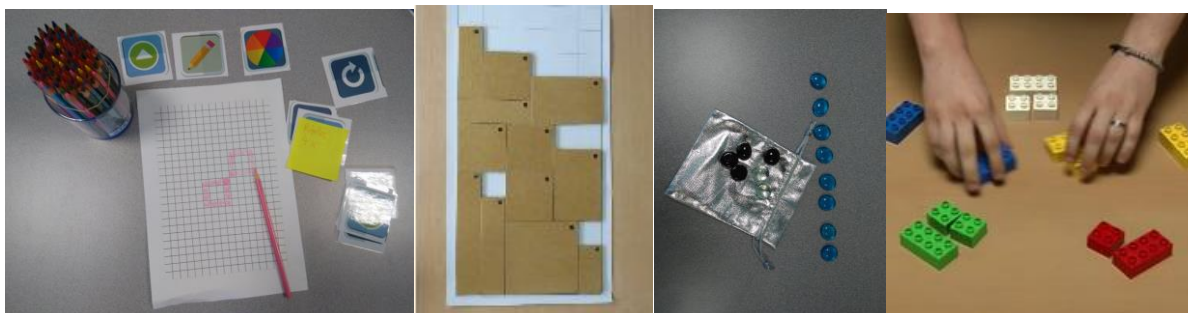


Figure 1 : matériel pour des activités liées aux algorithmes

(image 1 et 3 issues de (Alayrangues et al.), image 2 issue de (IREM Grenoble), image 4 issue de (Pixees))

Les activités autour des codes secrets et plus particulièrement de la cryptographie (Alayrangues et al.) permettent d'aborder la notion du codage (et en l'occurrence même de chiffrement) d'informations. Elles peuvent traiter de cryptographie symétrique, où la clé de chiffrement et la même que celle de déchiffrement (chiffre de César, scytale) ou de cryptographie asymétrique, où les clés de chiffrement et déchiffrement sont différentes (activité du coffre et des cadenas). Un autre terrain de jeu intéressant est le codage des images (IREM Clermont-Ferrand). On peut s'intéresser aux images binaires, autrement dit en noir et blanc, par exemple via un tableau de 0 et de 1. On peut également se poser la question de la compression de telles images, autrement dit de trouver un codage moins gourmand en mémoire. On peut aussi étendre l'activité à la représentation des images en couleur. Enfin, il est possible de réaliser des activités autour de la stéganographie (Alayrangues et al.), qui consiste à dissimuler une information dans une autre information. Une telle activité peut, par exemple, consister à cacher des chiffres ou des lettres dans une « image » réalisée avec des perles à repasser.



Figure 2 : matériel pour des activités liées à la représentation des données

(image 1 à 3 issues de (Alayrangues et al.))

De façon peut-être plus inattendue, il est aussi possible de réaliser des activités débranchées autour du fonctionnement des machines. Une approche jeu de rôle (IREM Clermont-Ferrand) peut, par exemple, permettre aux participants de se familiariser avec les différents composants d'un ordinateur en se mettant à leur place et en coopérant pour exécuter un petit programme (par exemple, la saisie et l'affichage d'un caractère). Des activités permettent également de s'immerger dans les principes du parallélisme, c'est le cas de l'activité réseau de tri (Duflot-Kremer). D'autres donnent à expérimenter le fonctionnement d'un réseau d'ordinateurs (Duflot-Kremer) pour comprendre comment les messages y circulent de point à point, comment on peut pallier les défaillances d'une machine du réseau, comment la structure du réseau (on parle aussi de sa topologie) influe sur le temps d'acheminement des messages, etc.

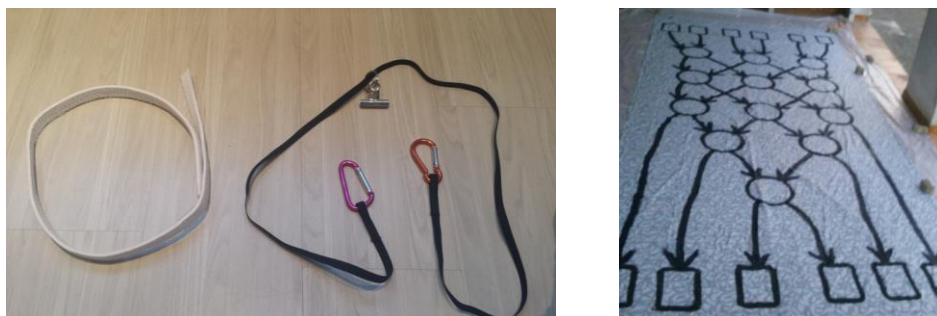


Figure 3 : matériel pour des activités liées aux machines

(image 1 issue de (Duflot-Kremer), image 2 issue de (Pixees))

Les activités débranchées illustrant la notion de langage sont très souvent liées aux activités algorithmiques. Dans l'activité du robot-idiot (Duflot-Kremer), on peut notamment réfléchir à la manière d'exprimer les instructions qui permettront de diriger le robot pour atteindre le but fixé. L'activité Gobot (IREM Clermont-Ferrand) met à disposition un langage simplifié dont les instructions sont exprimées par des cartes représentant des flèches. Enfin, il est aussi possible de s'intéresser à la manière dont un langage peut être reconnu par un ordinateur, autrement dit, étant donné un mot, comment savoir s'il appartient ou non à un langage donné. Il s'agit d'aborder la notion d'automate, ce que propose notamment l'activité « jeu du labyrinthe » (IREM Clermont-Ferrand).



Figure 4 : matériel pour des activités liées aux langages

(image 1 issue de (IREM Grenoble), image 2 issue de (IREM Clermont))

Construire un atelier

Généralités

Selon qu'un atelier est imaginé dans le cadre d'activités de médiation scientifique ou dans le cadre scolaire, le processus de création sera différent : les créateurs ne sont pas les mêmes, les objectifs non plus.

Pour construire une activité dans le cadre de la médiation scientifique, on choisira probablement en premier un thème informatique puis on fixera autour de lui un certain nombre d'objectifs. Ici la difficulté, étant donné un thème, consiste à identifier les aspects que l'on peut travailler en fonction du contexte dans lequel se déroulera l'activité (atelier court ou long, ponctuel ou régulier, devant un public « homogène » (par rapport à l'âge par exemple) ou « hétérogène »...).

Dans le contexte scolaire, il sera sans doute plus naturel de commencer par fixer les objectifs en fonction des attendus du programme puis de trouver le thème informatique qui permettra de les illustrer au mieux. Cette approche nécessite une culture informatique solide pour trouver le thème le plus approprié en fonction des objectifs fixés.

Dans les deux cas, il est primordial de connaître le public auquel on s'adresse, ses capacités cognitives, et quand cela est possible les compétences censées avoir été acquises précédemment.

Pour rendre ceci plus concret, considérons comment il est possible de construire un atelier autour de l'algorithmique.

Exemple de construction d'un atelier autour de l'algorithmique

L'algorithmique est un vaste sujet et les objectifs visés peuvent être de natures très différentes. Par exemple, on peut, étant donné un problème, se contenter de trouver un algorithme qui permet de le résoudre. Mais on peut assez naturellement se poser également la question de prouver que

l'algorithme trouvé fonctionne. Plus difficile, comment savoir si l'algorithme est efficace, voire optimal ? L'algorithme peut-il être adapté pour résoudre un problème similaire ?

Ainsi, un même atelier peut se décliner de manière différente selon le public auquel il s'adresse et du temps dont on dispose.

Nous avons conçu notre première activité, « Pixel art et algorithmique » (Alayrangues et al.), pour un atelier destiné à un public « à partir de 8 ans » dans le centre de culture scientifique et technique de Poitiers, l'espace Mendès France (Espace Mendès France : culture & médiation scientifiques).

Pour cette première expérience, nous avons choisi d'explorer un thème classique, l'algorithmique et de nous inspirer d'activités déjà réalisées par d'autres : nous avons ainsi repris le principe de la tortue Logo (Papert, 1992).

Nous nous sommes fixés plusieurs objectifs :

- Réflexion sur les informations que l'algorithme va manipuler et sur leur mémorisation via des variables
- Découverte de la notion d'algorithme (en l'illustrant par la réalisation de dessins) :
 - avec un jeu d'instructions très limité
 - avec des structures de contrôle pour répéter des instructions
- Exécution d'un algorithme « à la main »
- Construction d'un algorithme pour obtenir le résultat souhaité
- Découverte de la notion de « bug » (ici, erreur au niveau de l'algorithme)

Pour l'aspect concret, nous avons classiquement représenté les instructions par des cartes. Dans un premier temps, seules les instructions « tourner » et « avancer » sont disponibles. Elles sont représentées par des flèches. On ajoute plus tard dans l'atelier la possibilité de lever et baisser le crayon pour réaliser des dessins non connexes. Il est aussi prévu de disposer d'une instruction permettant de changer la couleur du crayon.

Pour les boucles, on dispose d'élastiques pour rassembler les séquences d'instructions que l'on souhaite répéter et des papiers autocollants pour écrire le nombre d'itérations à effectuer.

On dispose aussi de cartes « variable » pour mémoriser à tout instant l'orientation courante (autrement dit, la direction dans laquelle avance le crayon) et éventuellement la couleur et le fait que le crayon soit baissé ou levé.

Au-delà de la découverte des notions, il existe une difficulté supplémentaire. Lorsqu'un humain exécute un algorithme à la place de l'ordinateur, il faut faire attention à ce qu'il ne corrige pas involontairement des bugs présents dans l'algorithme (ou qu'il n'en introduise pas d'autres). Une manière de limiter le problème consiste à faire exécuter l'algorithme par un participant autre que celui qui l'a inventé.

Retours d'expérience

Nous avons lancé un petit sondage informel auprès de plusieurs personnes ayant réalisé des ateliers dans le cadre scolaire (de la maternelle au lycée). Nous n'avons pas la place ici de présenter l'ensemble des réponses obtenues. Nous commençons par une courte synthèse générale, puis présentons les trois activités qui ont été le plus citées dans les retours. Nous présentons pour chacune

un lien vers sa description détaillée, le(s) public(s) avec lequel elle a été jouée, la ou les catégories à laquelle elle appartient, un bref résumé, sa durée et quelques remarques des personnes qui les ont préparées et animées.

Il est à noter que les activités ont été selon les cas réalisées par des enseignants, des enseignants-chercheurs en informatique ou des étudiants en informatique. La plupart des enseignants se sont fondés sur des activités mises à disposition sur Internet.

Les retours obtenus correspondent essentiellement au ressenti des personnes qui ont animé l'activité et à l'appréciation des participants. Dans certains cas, les activités ont été immédiatement suivies d'un temps de bilan avec les élèves.

Dans tous les cas, les participants se prêtent globalement bien au jeu et apprécient l'« originalité » des activités et leur côté ludique. Ils se montrent actifs et s'ils sont accompagnés dans la démarche, ils construisent eux-mêmes les solutions aux problèmes qui leur sont posés. Ils apprécient également de pouvoir travailler par petits groupes. Pour certaines activités, la possibilité de « rebrancher » a été particulièrement appréciée. Il a aussi été remarqué que ce ne sont pas forcément les élèves d'ordinaire les plus à l'aise dans la classe qui réussissent le mieux les activités.

Dans plusieurs cas, les activités ont été animées par deux enseignants, parfois de disciplines différentes. Pour certains, il s'agissait aussi de premières expériences d'enseignement de l'informatique et, dans l'ensemble, l'expérience leur a donné envie de recommencer. Souvent, les ateliers ont été réalisés en petits groupes, ce qui peut demander, pour fonctionner au mieux, un bon taux d'encadrement. Les activités en groupes ont régulièrement été suivies d'un temps en classe entière pour dresser un bilan voire reconstruire ensemble une solution au problème posé.

Les collaborations avec des étudiants en informatique ou des enseignants-chercheurs ont été également très appréciées à la fois par les participants et par les enseignants.

En ce qui nous concerne, nous avons mené la plupart de nos activités dans un cadre périscolaire, au centre de médiation de Culture Scientifique Technique et Industrielle de Poitiers, l'espace Mendès France. La plupart des remarques précédentes peuvent s'appliquer à ce que nous y avons vécu. La différence essentielle en ce qui nous concerne est la diversité du public. Les ateliers sont ouverts à partir de 8 ans sans limite d'âge supérieure. Nous avons ainsi été amenés à réaliser des ateliers avec des groupes d'enfants, des groupes d'adultes, des groupes mixtes (enfants, ados, adultes). Si cette diversité est une richesse et source d'échanges parfois inattendus, la principale difficulté est que nous découvrons la constitution du groupe le jour de l'atelier.

Gobelets (Cargobot ou Gobot)

Descriptions de l'activité (deux versions) :

<http://www-irem.ujf-grenoble.fr/spip/spip.php?article146#cargobot>

<http://www.irem.univ-bpclermont.fr/Informatique-sans-Ordinateur#ancre-5>

Public : Cycle 3 en demi classe, ou Cycle 4 en classe complète.

Catégorie : Algorithmes, Langages

Résumé : Écrire des suites d'instructions, autrement dit des petits programmes, pour commander un robot (une grue) qui déplace des containers (représentés par des gobelets). Cette activité est

notamment l'occasion d'introduire les notions de programmation, d'exécution d'un programme et de débogage.

Durée : de 50 minutes à 2h.

Remarques :

- L'activité est particulièrement adaptée pour une première découverte de l'algorithmique et illustre bien que l'ordinateur ne fait rien d'autre qu'exécuter les instructions qui lui sont fournies ;
- Elle constitue un point d'entrée intéressant à la fois pour les enfants et pour les enseignants qui n'auraient pas de connaissance préalable en informatique et quelques inquiétudes à l'enseigner ;
- Après une phase de « jeu » avec les gobelets, les élèves comprennent la nécessité de mettre en place une démarche rigoureuse ;
- Certaines versions de l'activité peuvent être envisagées dès la maternelle ;
- L'activité peut aussi servir de support pour travailler le langage ;
- Il est aisé de « rebrancher » l'activité avec Lightbot (soit avec Lightbot Flash, soit avec une application pour smartphones et tablettes : <http://lightbot.com/>) ;
- En Cycle 3, un groupe complet (20 ou plus) rend l'activité difficile à gérer.

Images

Description de l'activité :

<http://www.irem.univ-bpclermont.fr/Informatique-sans-Ordinateur#ancree-6>

Public : Cycle 3 en petit groupe, Cycle 4 en classe complète.

Catégorie : Information/ Données

Résumé : Coder une image à la manière d'un ordinateur (avec ou sans compression)

Durée : de 1h15 à 2h.

Remarques :

- Les principes ont été compris assez rapidement et la synthèse est venue des élèves ;
- Les élèves ont vite eu envie de coder et décoder leurs propres images ;
- L'activité peut donner lieu à des activités sur ordinateur en manipulant des images, par exemple au format PPM qui stocke les valeurs de tous les pixels de l'image

Crêpier

Description de l'activité :

<http://www-irem.ujf-grenoble.fr/spip/spip.php?article146#crepier>

Public : cycle 4 ou cycle 3 (en demi groupe)

Catégorie : Algorithmique (Tri)

Résumé : Trouver un algorithme pour trier une pile de crêpes en fonction de contraintes (tailles....)

Durée : de 1h30 à 2h.

Remarques :

- Niveau assez avancé. Il vaut mieux prévoir un taux d'encadrement suffisant (ex : 2 enseignants pour 15 élèves en 5e) pour compenser la différence de vitesse d'avancement entre les groupes ;
- La notion de boucle arrive très tôt, et les alternatives (si-alors-sinon) viennent ensuite ;
- Rentre bien dans le cadre de la réforme.

Conclusion et perspectives

L'informatique sans ordinateur est un point d'entrée de la science informatique particulièrement riche, en premier lieu, parce qu'elle permet d'en aborder les quatre piliers : algorithmes, données, machines et langages. Elle présente également l'avantage de ne pas nécessiter de matériel coûteux pour être mise en œuvre.

Monter une telle activité demande à la fois des connaissances en informatique, et des compétences en pédagogie. Aussi, la création d'une activité peut-elle être le fruit de dialogues fructueux entre informaticiens et enseignants. Les informaticiens maîtrisent les concepts à transmettre tandis que les enseignants connaissent à la fois le « public » de l'activité et le cadre imposé par les programmes.

En outre, il existe une vaste littérature sur laquelle s'appuyer, et plusieurs communautés se développent pour partager, échanger, « remixer » et inventer de nouvelles activités (Groupe InfoSansOrdi hébergé par la SIF (SIF, groupe InfoSansOrdi) ou communauté Class'code (Class'code, 2017) qui utilise entre autres, mais pas seulement, de telles activités). Ainsi, outre les aspects scientifiques, on retrouve des aspects de travail collaboratif qui font aussi partie de la culture informatique.

Ces activités (et de manière générale l'initiation à l'informatique) présentent d'autres avantages : elles contribuent notamment à développer des compétences qui ne sont pas propres à l'informatique mais s'avèrent utiles dans bien d'autres apprentissages : rigueur, langage, logique, créativité, résolution de problème, travail collaboratif, etc.

Deux questions relatives à l'utilisation de cette approche débranchée pour se familiariser avec l'informatique sont cependant encore insuffisamment explorées.

On peut tout d'abord se demander comment évaluer l'impact de cette approche. Ces activités ont, initialement, été développées dans un contexte de médiation scientifique où l'objectif n'est pas d'acquérir des connaissances et compétences mais de susciter la curiosité et l'envie d'en découvrir davantage. Aussi les évaluations classiquement réalisées dans ce contexte s'intéressent-elles plutôt au ressenti des participants qu'à l'estimation de ce qu'ils ont appris et à la manière de réinvestir ces connaissances et compétences. Des analyses didactiques autour de l'enseignement de l'informatique, notamment par le biais de l'informatique débranchée, commencent à se développer pour tenter d'aller plus loin et de déterminer l'impact de cette approche en terme d'apprentissage (Drot-Delange, 2013).

En outre, l'informatique débranchée, si elle donne un certain nombre de clés de la science informatique, ne peut être suffisante lorsqu'il s'agit d'approfondir les connaissances et d'acquérir des compétences techniques. Pour atteindre ces objectifs, il est nécessaire de rebrancher les activités, que ce soit avec un langage de blocs (comme Scratch) ou un langage de programmation textuel

(comme Python). Un bon dosage est à trouver en fonction des attendus des programmes des différents niveaux, en gardant à l'esprit qu'il ne s'agit pas de former des informaticiens mais des citoyens capables d'appréhender les enjeux informatiques de la société dans laquelle ils sont appelés à vivre.

Remerciements

Marie Duflot-Kremer, Séverine Fleury, Noëlle Lacourt, Émilie Marcon, Fabien Marsollat, Marianne Mognos, Malika More, Martin Quinson.

Références

- Académie des sciences, 2013. L'enseignement de l'informatique en France - Il est urgent de ne plus attendre, <http://www.academie-sciences.fr/fr/Rapports-ouvrages-avis-et-recommandations-de-l-Academie/l-enseignement-de-l-informatique-en-france-il-est-urgent-de-ne-plus-attendre.html>
- Alayrangues, S., Peltier, S., Signac, L., poitiers-infosansordi [WWW Document]. GitLab. URL <https://framagit.org/poitiers-infosansordi>.
- Algo-rythmics, Technologically and artistically enhanced multi-sensory computer science education (Inter-cultural method for teaching-learning sorting algorithms) [WWW Document]. URL <http://algo-rythmics.ms.sapientia.ro/dance/insertion>.
- Anderson, C., (2013). The Maker Movement: Tangible Goods Emerge From Ones and Zeros. WIRED, <https://www.wired.com/2013/04/makermovement/>.
- Bell, T., (2015). Computer Science Unplugged [WWW Document]. URL <http://csunplugged.org/>
- Class'code, (2017). Class'Code v2 [WWW Document]. URL <https://pixees.fr/classcode-v2/>
- Conseil scientifique de la Société informatique de France (SIF), 2014. L'informatique, la science au coeur du numérique. 1024 Bull. Société Inform. Fr. 2, 13--20. <http://www.societe-informatique-de-france.fr/wp-content/uploads/2014/02/1024-2-science-au-coeur-du-numerique.pdf>
- Crozer, G., Groperrin, R., (1987). Informatique sans ordinateur. URL <https://files.inria.fr/mecsci/groperrin-et-al/informatique-sans-ordinateur.pdf>
- Di Cosmo, R., (2015). Enseigner et apprendre les sciences informatiques à l'école [WWW Document]. Interstices. URL <https://interstices.info/informatique-ecole>.
- Drot-Delange, B., (2013). Enseigner l'informatique débranchée : analyse didactique d'activités. Presented at the AREF, pp. 1–13. https://archivesic.ccsd.cnrs.fr/sic_00955208/document
- Duflot-Kremer, M. La page médiation de Marie [WWW Document]. URL <https://members.loria.fr/MDuflot/files/med/index.html>.
- Espace Mendès France : culture & médiation scientifiques [WWW Document] URL <https://emf.fr/>.
- Fellows, M.R., Parberry, I., (1993). SIGACT trying to get children excited about CS. Comput. Res. NEWS 5, 7. <http://archive.cra.org/CRN/issues/9301.pdf>
- Groupe IREM Grenoble, (2017). Action informatique débranchée [WWW Document]. URL <http://www-irem.ujf-grenoble.fr/spip/spip.php?article146#cargobot>.
- IREM Clermont-Ferrand. Informatique sans Ordinateur [WWW Document]. URL <http://www.irem.univ-bpclermont.fr/Informatique-sans-Ordinateur>.
- Noirpoudre, S., Roy, D., Desprez, T., Segonds, T., Caselli, D., Oudeyer, P.-Y., (2017). Poppy Education : un dispositif robotique open source pour l'enseignement de l'informatique et de

la robotique. Presented at the EIAH 2017 - Environnements Informatiques pour l'Apprentissage Humain, p. 8. <https://hal.inria.fr/hal-01517941/document>

Papert, S., (1992). Jaillissement de l'esprit. *Flammarion*.

Pixees [WWW Document]. URL <http://pixees.fr>

Quinson, M., Sciences manuelles du numérique [WWW Document]. URL <http://people.irisa.fr/Martin.Quinson/Mediation/SMN/>

Réseau français des Fab'lab, (2015). RFFLabs – Le site du réseau Français des Fablabs [WWW Document]. URL <http://www.fablab.fr/>

Romero, M., (2016). De l'apprentissage procédural de la programmation à l'intégration interdisciplinaire de la programmation créative. *Form. Prof. Rev. Sci. Int. En Éducation* 24, 87–89. <http://formation-profession.org/pages/article/24/13/a92>

Romero, M., Vallerand, V., (2016). Co-creative activities for 21st Century Kids. <https://www.slideshare.net/margarida.romero/romero-vallerand-2016-cocreative-activities-for-the-21st-century-kidsr02>

SIF, groupe InfoSansOrdi [WWW Document]. URL <http://www.societe-informatique-de-france.fr/mediation/infosansordi/>

Wing, J., (2009). La pensée informatique. *Interstices*.