



Enseigner l'informatique débranchée : analyse didactique d'activités

Béatrice Drot-Delange

► To cite this version:

Béatrice Drot-Delange. Enseigner l'informatique débranchée : analyse didactique d'activités. AREF, Aug 2013, France. pp.1-13, 2013. <sic_00955208>

HAL Id: sic_00955208

https://archivesic.ccsd.cnrs.fr/sic_00955208

Submitted on 4 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

380 - Enseigner l'informatique débranchée : analyse didactique d'activités

Béatrice Drot-Delange

Clermont Université, Université Blaise Pascal, EA 4281, ACTé, France

Mots clés : informatique sans ordinateur, enseignement primaire, enseignement secondaire, didactique de l'informatique, système de numération, représentation binaire.

Résumé : L'enseignement de l'informatique en tant que science a repris place dans le système éducatif français sous une forme optionnelle pour les filières scientifiques en terminales au lycée (Drot-Delange, 2012). En marge des disciplines scolaires, des initiatives ont vu le jour pour diffuser cette culture scientifique et technique. L'enseignement de l'informatique sans ordinateur (dite informatique débranchée ou Computer Science Unplugged) en est un exemple. Véritable communauté internationale, fondée par Tim Bell et ses collègues, le mouvement Unplugged est de plus en plus populaire (Bell, Alexander, Freeman, & Grimley, 2009). Il s'agit d'initier aux concepts et méthodes de l'informatique, sous une forme attractive et ludique. Les prescriptions internationales de curriculum en informatique s'y réfèrent explicitement (Tucker, 2003). L'objectif de notre communication est de nous interroger sur les apports de la mise en œuvre dans les classes de ces activités à l'apprentissage de l'informatique, et plus particulièrement sur l'apprentissage de la représentation binaire de l'information.

1. Introduction

La situation de l'enseignement de l'informatique dans l'enseignement primaire et secondaire est très variable d'un pays à l'autre : optionnel ou obligatoire, avec des enseignants spécialisés ou non, etc. L'histoire de cette « discipline » en France, dans l'enseignement général, montre de multiples péripéties (Drot-Delange, 2012; Tort & Drot-Delange, 2013). Actuellement, l'informatique est proposée en enseignement de spécialité dans les lycées en terminales scientifiques, depuis la rentrée 2012, et en option pour les autres filières de terminales à partir de la rentrée 2013. Cette faible présence institutionnelle, partagée par de nombreux pays, s'accompagne d'initiatives de sensibilisation à l'informatique, dont certaines se développent en France : le concours Castor¹ (Tort & Dagiene, 2012) et l'informatique sans ordinateur pour ne citer que ces deux programmes (Bell, Curzon, Cutts, Dagiene, & Haberman, 2011). Le parti pris de ces mouvements est de ne pas proposer la programmation, et l'apprentissage d'un langage, comme porte d'entrée pour l'informatique. L'objectif est de faire découvrir des concepts et des méthodes spécifiques à la science informatique, de montrer en quoi elle se distingue des technologies de l'information et de la communication, généralement bien présentes dans les cursus scolaires et qui donneraient une vision faussée ou stéréotypée de l'informatique.

Notre communication porte sur les apports de cette démarche pour l'apprentissage de l'informatique, et plus particulièrement concernant la représentation binaire de l'information.

Nous présenterons d'abord les principes qui fondent la construction de ces activités d'« informatique sans ordinateur » selon leurs initiateurs, puis nous mènerons une revue de la littérature concernant les évaluations de ces activités en classe (partie 2). Nous étudierons plus précisément le cas de la représentation binaire de l'information dans ces activités sans ordinateur, considérée comme un des savoirs essentiels dans une sensibilisation à l'informatique (partie 3).

2. L'informatique « débranchée » : quels fondements ?

1 <http://castor-informatique.fr/>

2.1 Un mouvement ancien

Le courant de l'informatique dite débranchée (*Computer Science Unplugged*, CSU dans la suite du texte) est né en 1992 à l'université de Canterbury en Nouvelle-Zélande. Il connaît un large essor international depuis 2003, avec une intégration de plus en plus grande dans les classes. A cette date, un rapport de l'*Association for Computing Machinery (ACM)* et de la *Computer Science Teachers Association* (Tucker, 2003) propose un curriculum en informatique. Il s'appuie explicitement sur les activités de l'ouvrage CSU (Fellows, Bell, & Witten, 1996). Celles-ci sont désormais disponibles en téléchargement et traduites dans de nombreuses langues². Le site CSU³ est aussi une plate-forme collaborative et de diffusion des pratiques. La visibilité sur le web s'est accrue par le *sponsoring* de Google depuis 2006, permettant la mise à disposition gratuite des ressources sur le web et de vidéos de séquences filmées. La fréquentation du site web est élevée, plus de 12 000 visiteurs uniques par semaine en octobre 2011 par exemple (Bell, Rosamond, & Casey, 2012).

2.2 Des principes pédagogiques

Selon leurs initiateurs, l'informatique sans ordinateur propose des activités qui permettraient de découvrir les concepts de informatique par des jeux impliquant la manipulation d'objets, des tours de magie, de puzzles. Selon Bell *et alii.* (2012), dans une approche constructiviste, les élèves sont guidés par des questions leur permettant de découvrir les concepts de l'informatique par eux-mêmes. Les activités donneraient suffisamment d'indices pour qu'ils puissent travailler sur les principes par eux-mêmes. On retrouve une idée largement partagée dans le champ de l'enseignement de l'informatique de la nécessité de faire manipuler des objets tangibles, d'expérimenter à partir de situations réelles (Futschek & Moschitz, 2011; Greff, 1998; Komis & Misirli, 2011; Papert, 1980)

La construction des activités, principalement kinesthésiques, repose sur un certain nombre de principes explicités par Bell, Alexander, Freeman et Grimley (2008). Nous les reprenons ci-après (traduit librement par nos soins). Le projet se focalise sur la démonstration des concepts de la science informatique plutôt que sur la programmation. Les activités doivent engager le corps, et impliquées, si possible, un travail d'équipe. Elles doivent être agréables et non ennuyeuses. Le matériel nécessaire doit être disponible à faible coût, sous licence *creative commons*. Les activités doivent être neutres du point de vue du genre, favoriser des approches coopératives plutôt qu'individuelles. Elles prennent souvent la forme d'histoire pour retenir l'attention des enfants. Ils sont amenés à découvrir les réponses par eux-mêmes, l'objectif n'étant pas d'enseigner des réponses, mais de faire « jouer » avec les concepts. Une activité doit être raisonnablement robuste à l'erreur, pour qu'une petite erreur d'un enfant ou d'un enseignant ne la rende pas totalement caduque. Selon les auteurs, trois critères permettent d'évaluer si une activité respecte cette philosophie : la simplicité (les règles doivent pouvoir être expliquées rapidement), l'engagement (l'activité est attrayante pour les enfants) et la coopération ou compétition (les enfants doivent être motivés pour atteindre le but). Les résultats attendus de la mise en œuvre de ces activités sont d'une part de donner l'envie d'approfondir ses connaissances en poursuivant par exemple des études en informatique et, d'autre part, de donner à voir en quoi consiste le travail scientifique en informatique pour mieux faire connaître les carrières de l'informatique (Bell et al., 2009).

2.3 Quels savoirs ?

La première édition du manuel (Fellows et al., 1996) livrait des activités, sans instruction particulière pour les enseignants. Mais il s'est vite avéré qu'un minimum d'explications était nécessaire pour des enseignants qui n'avaient pas toujours la culture informatique et mathématique nécessaires à la prise en main et à l'exploitation de ces activités (Bell, Rosamond, & Casey, 2012). Une version écrite par des enseignants pour les enseignants a donc été éditée (Fellows, Bell, & Witten, 2002) comprenant des activités testées en classe. Les activités dans l'ouvrage CSU sont organisées autour de trois concepts informatiques : les données, les algorithmes et la représentation de procédures (voir tableau 1). Le site CSU propose une

2 L'équipe d'Interstices a coordonné la traduction des activités en français. Interstices se définit comme une revue de culture scientifique sur la recherche en informatique, créée et éditée par l'INRIA, animée par des chercheurs.
<https://interstices.info>

3 <http://csunplugged.org/>

classification des activités selon les principes de l'informatique de Denning (2010). Selon lui, le champ de l'informatique se subdiviserait en sept domaines. Les 19 activités présentées sur le site web se répartissent en 8 activités dans le domaine du calcul, 6 dans celui de la communication, 1 dans celui de la coordination, 2 dans la mémorisation (qui inclut la représentation, le stockage et la recherche d'information), 2 dans l'automatisation. Deux domaines sont pour l'instant absent des activités : la conception et l'évaluation.

Chap.	Titre	Sous-titre	Age
1	Compter les points	Écriture binaire des nombres	7 ans et +
2	La couleur par les nombres	Représentation des images	7 ans et +
3	Peux-tu répéter ?	Compression de texte	9 ans et +
4	Tour de cartes	Détection et correction des erreurs	9 ans et +
5	Vingt devinettes	Théorie de l'information	10 ans et +
6	La bataille navale	Algorithmes de recherche	9 ans et +
7	Le plus léger et le plus lourd	Algorithmes de tri	8 ans et +
8	La course contre la montre	Réseaux de tri	7 ans et +
9	La ville embourbée	Arbres couvrants	9 ans et +
10	Le jeu de l'orange	Acheminement et blocage dans les réseaux	9 ans et +
11	La chasse au trésor	Automates d'états finis	9 ans et +
12	Donner et exécuter des ordres	Langages de programmation	7 ans et +

Tableau 1: contenus du manuel *Computer Science Unplugged* (traduction française *Interstices*, 2009)

2.4 Quelles évaluations?

Face à l'engouement pour ce mouvement et son utilisation dans les standards internationaux, on s'attendrait à ce que ce sujet ait suscité de nombreuses recherches sur les apprentissages. Mais une interrogation des bases de données bibliographiques confirme le constat déjà effectué par Thies et Vahrenhold (2013) de la faible quantité de travaux traitant de cette question.

2.4.1 Les recherches sur la satisfaction et la motivation

Les études existantes cherchent principalement à montrer les apports de cette démarche auprès des élèves en termes de satisfaction, de motivation et d'intérêt pour l'informatique. Nous en présentons les résultats ci-après.

Lambert et Guiffre (2009) montrent que les élèves, après les séances, sont plus intéressés par l'informatique, se sentent significativement plus confiants en mathématiques, mais pas significativement plus intéressés par les mathématiques (voir tableau 2).

Public	Activités proposées	Méthode de recueil des données
(États-Unis) 3 classes Grade 4 (équivalent CM1). 3 sessions de 30 mn chacune, une fois par semaine + 2 sessions pré-test (une semaine avant la 1ère activité) et post-test (une semaine après la dernière activité). Mars-Avril 2007	Activité 1 : représentation binaire des nombres, textes, sons et images (ouvrage <i>CSU</i> , édition 2002) Activité 2 : mathématiques (fractions et divisions) (hors manuel) Activités 3 : envoi de données par Internet (détection et correction des erreurs – activité 4, ouvrage <i>CSU</i> , édition 2002) + le protocole TCP/IP (hors manuel)	Questionnaire : 15 questions simples (échelle de Likert, 5 positions) mesurant la confiance dans les capacités cognitives, l'intérêt pour la science informatique, les mathématiques, et l'anxiété par rapport aux mathématiques.

Tableau 2: Protocole utilisé par Lambert & Guiffre (2009)

Nishida et ali. (2008) montrent dans leurs recherches la satisfaction des élèves lors des séances utilisant les activités *CSU*, quel que soit le niveau atteint dans leur scolarité et leur spécialité (voir tableau 3). Ils soulignent que l'intérêt de ces séquences est d'amener les élèves à penser par eux-mêmes, à accroître leur motivation à comprendre.

Public	Activités proposées	Méthode de recueil des données
(Japon, élèves de 14-15ans) Première période : 4 classes de 16 élèves, 10 heures, optionnel. Seconde période : 4 classes de 12 élèves, 9 heures, obligatoire.	Activités 1 à 4, 6, 8 à 11. (ouvrage <i>CSU</i>)	Questions fermées et ouvertes.
(Japon, élèves de 16-17 ans), dans des cours d'informatique 33 élèves, 3 séances	Activités 6, 7 et 9. (ouvrage <i>CSU</i>) Activités « classiques »	Questionnaire.
(Japon, élèves de 17-18 ans), cours de programmation. 10 élèves.	Activités 2, 4, 6 et 11 en introduction d'exercice de programmation (ouvrage <i>CSU</i>).	Questionnaire.

Tableau 3: Protocole utilisé par Nishida et ali. (2008)

Taub, Armoni et Ben-Ari (2012) étudient si les objectifs assignés à l'« informatique sans ordinateur » sont remplis auprès d'élèves de collèges (grade 7), à savoir s'ils ont une représentation plus claire de l'informatique et s'ils ont envie d'en continuer l'étude plus tard (voir tableau 4). Les résultats obtenus montrent que les élèves sont moins attirés par l'informatique et la considèrent comme moins intéressante que ce qu'ils déclareraient avant de faire les activités sans ordinateur. Une explication donnée par les auteurs serait que les élèves ne comprennent pas de quelle manière les activités proposées sont représentatives de l'informatique étudiée dans l'enseignement secondaire ou d'une éventuelle future carrière. Les liens entre les activités et les concepts informatiques ne sont pas toujours explicités pour les élèves, même si les documents les précisent de manière plus formelle pour les enseignants.

Public	Activités proposées	Méthode de recueil des données

(Israël, élèves de 12-13 ans) 2 écoles, dans la 1 ^{ère} : 2 classes (filles uniquement), 52 élèves ; dans la 2 ^{ème} (filles et garçons), 1 classe (26 élèves), 2 heures par semaine, sur un semestre	Activités effectuées par les enseignants des classes concernés : 1 à 7 et 14 pour la 1 ^{ère} , de 1 à 7 et 8 pour la 2 ^{ème} (CSU, édition 1996)	Questionnaires (points de vue sur la nature de l'informatique, sur le travail en informatique et sur les femmes et l'informatique ; attitudes envers l'informatique, les scientifiques et les activités débranchées ; intentions d'étudier et de travailler en informatique) : 22 propositions, échelle de Likert, 5 positions. Avant le premier cours et après le dernier. Entretiens : recueil des réponses à partir d'images, de lecture de textes, questions fermées et ouvertes.
---	--	--

Tableau 4: Protocole utilisé par Taub, Armoni et Ben-Ari (2012)

2.4.2 Les recherches sur les apprentissages

Feaster, Segars, Wahba et Hallstrom (2011) comparent l'effet des activités sans ordinateur à un enseignement traditionnel sur les attitudes des élèves envers la science informatique et sur leur perception de leur compréhension des contenus. Les résultats obtenus (voir tableau 5) montrent qu'il n'y a pas de gain significatif en termes de confiance ou d'intérêt pour l'informatique, ni en termes de compréhension des contenus. Les auteurs émettent plusieurs hypothèses pour expliquer ce résultat : les activités basées sur les jeux et la manipulation conviendraient mieux aux plus jeunes élèves qu'aux plus âgés ; les élèves concernés ici étaient déjà engagés dans des cours en informatique, ils se percevaient comme expérimentés et moins concernés par l'apprentissage de ces concepts. Ils ont l'expérience de programmation de jeux, etc., les activités proposées leur ont peut-être paru moins gratifiantes voire inutiles.

Public	Activités proposées	Méthode de recueil des données
(Etats-Unis) 1 lycée. 2 groupes au 1 ^{er} semestre : groupe expérimental de 14 élèves (ouvrage CSU), groupe contrôle de 15 élèves (méthode traditionnelle). Au second semestre, les élèves du groupe contrôle ont les séances avec l'ouvrage CSU.	10 sessions d'une heure, répétées pendant deux semestres dans un cours d'introduction à la programmation. 9 des 10 sessions sont basées sur l'ouvrage CSU.	Avant la première session et le dernier jour, le même questionnaire auprès des 2 groupes. Échelle de Likert (6 degrés), 18 questions sur les attitudes et la compréhension des contenus et 2 questions ouvertes.

Tableau 5: Protocole utilisé par Feaster et ali.(2011)

Fort des résultats précédents, Thies et Vahrenhold (2013) veulent valider l'hypothèse qu'utiliser les activités CSU peut être adéquate et efficace, lorsque cette démarche est soigneusement intégrée dans l'enseignement secondaire au niveau du collège (voir tableau 6). Les évaluations réalisées pour chaque sujet abordé devaient permettre de répondre aux questions suivantes :

- dans quelle mesure les élèves peuvent-ils résoudre avec succès des exercices relatifs à chaque sujet ?
- à quel niveau d'abstraction les élèves peuvent-ils décrire chaque sujet abordé ?
- dans quelle mesure les élèves peuvent relier chaque sujet abordé à la science informatique?

Les résultats obtenus par les auteurs montrent qu'il n'y a pas de différence statistique significative entre le groupe CSU et le groupe traditionnel, tant dans les évaluations à court terme qu'à moyen terme. Les auteurs en concluent qu'il n'y a donc pas de contre-indication à employer cette méthode d'enseignement.

Public	Activités proposées	Méthode de recueil des données
(Allemagne, élèves de 11-12 ans) Une classe de 25 élèves option Sciences, Technologie, Ingénierie et Mathématiques, en 2 groupes.	Dans la même séance, le groupe 1 travaille sur une activité pendant 45mn selon l'ouvrage <i>CSU</i> , puis le second groupe avec d'autres ressources. Pour l'activité 8, les groupes sont intervertis. Activités 1, 6 et 8 (manuel <i>CSU</i> , édition 2002).	Évaluation le lendemain de la séance + rappel des contenus (sans référence aux supports utilisés) 3 semaines après + évaluation identique à la première 3 semaines après

Tableau 6: Protocole utilisé par Thies et Vaherenhold (2013)

D'autres recherches ont été menées mais concernent des activités qui se déroulent hors les classes (camps de vacances par exemple) ou auprès de publics ciblés (les filles, les élèves handicapés, les seniors, les parents ou les enseignants) (voir pour une synthèse Bell *et ali.* (2012).

3. Le cas de la représentation de l'information en binaire

Le système binaire semble être le B-A-BA de toute initiation à l'informatique. Les cursus des différents pays et de différents niveaux de classe le mentionnent, des panels d'experts le considèrent comme un des savoirs importants lors d'une initiation à l'informatique, des moyens pédagogiques ont été depuis fort longtemps inventés pour tenter d'en faciliter la compréhension.

3.1 La représentation binaire de l'information : un savoir à enseigner ?

Parmi les savoirs à enseigner dans un cours introductif en informatique figure en bonne place la représentation binaire de l'information. Dans les standards proposés par l'ACM, la représentation binaire doit être enseignée très tôt : dès le grade K2, les élèves doivent comprendre « comment les 0 et les 1 peuvent être utilisés pour représenter l'information, comme les images numériques ou les nombres » (Tucker, 2003, p.10). De même, en France, les programmes en terminale scientifique de la spécialité « Informatique et Sciences du Numérique » (ISN)⁴ commencent par la représentation de l'information. Le programme précise, concernant la représentation binaire, qu'un « ordinateur est une machine qui manipule des valeurs numériques représentées sous forme binaire ».

Des chercheurs ont utilisé la méthode Delphi, en réunissant une vingtaine d'experts, pour déterminer les sujets qui devaient être selon eux enseignés à des élèves débutants les cours d'informatique, évaluer leur niveau d'importance et leur degré de difficulté (Goldman et al., 2010). Malgré des dissensions, selon ce panel d'experts, la représentation des nombres est un des concepts importants que les élèves doivent comprendre, mais aussi le plus facile.

3.2 Quelles recherches didactiques sur ce point ?

Aigle (1989) s'interroge, lors du premier colloque francophone de 1988 à Paris sur la didactique de l'informatique, sur la manière de mener, au niveau de l'école élémentaire, une étude du codage de l'information dans un ordinateur. Il lui semble incontournable de montrer « qu'à divers endroits de l'ordinateur (mémoire, écran, imprimante, etc.) l'information a une structure binaire ». Il souligne que les enfants connaissent le binaire depuis le cours moyen 1ère année, mais « celui du mathématicien (...) [qui] s'applique à des nombres ». Il conviendrait donc selon lui de « mettre en évidence la nature électronique des signaux reçus, traités ou émis par l'ordinateur ». Il propose alors l'utilisation d'un chenillard, pour mettre en évidence les niveaux de tension, ou d'un afficheur 7 segments, pour mettre en évidence la notion de code.

Herman, Zilles et Loui (2011) étudient les erreurs des étudiants en informatique concernant la représentation des nombres, après avoir rappelés des résultats connus en didactique des mathématiques. Les erreurs concernent plus particulièrement le système de numération de position, la notation en complément à deux des entiers relatifs et le concept de débordement (*overflow*). Les auteurs mènent des entretiens auprès de 26 étudiants en informatique ayant obtenu des notes moyennes à leur examen. Les entretiens consistent en la résolution de

4 Ministère de l'Éducation Nationale, Bulletin Officiel spécial, n°8, 11 octobre 2011

problèmes (comparaison de nombres écrits dans différentes bases, addition et soustraction de nombres binaires, complément à deux, etc.) et à penser à voix haute en expliquant son raisonnement. Les représentations repérées dans les entretiens ont ensuite servi de base à la construction d'un test sous forme de questionnaire à choix multiple, administrés à 700 étudiants de 6 universités des Etats-Unis. Parmi les résultats obtenus, les représentations erronées concernent les erreurs liées à la maîtrise du système de numération de position. La notion de complément à deux pose également des difficultés d'interprétation. Les étudiants font des confusions entre la notation en complément à deux des entiers relatifs et le calcul de cette représentation. De même, les étudiants ont des représentations erronées concernant la notion de débordement, lié à la longueur fixe des registres.

Les hypothèses émises par les auteurs pour comprendre les erreurs tiennent principalement à l'ordre et à la manière dont sont présentés ces concepts. Par exemple, comprendre le concept de débordement implique d'avoir des connaissances sur l'architecture d'un ordinateur et les registres. Or ces sujets peuvent être traités à des moments éloignés dans le temps, ne permettant pas à l'étudiant de faire le lien. Les auteurs concèdent qu'aborder d'abord la représentation binaire puis les contraintes produites par l'architecture d'un ordinateur peut sembler un ordre logique, mais qu'il serait, au vu de leurs résultats, plus profitable de procéder à l'inverse. Une autre source possible d'explication est la prédominance de l'acquisition de savoirs procéduraux plutôt que conceptuels. C'est, selon les auteurs, ce qu'illustre le fait que les étudiants soient parfois capables de convertir un nombre d'une base dans une autre mais aient des difficultés à les comparer. Finalement, les auteurs concluent que les erreurs des étudiants concernant la représentation des nombres sont assimilables aux difficultés des élèves plus jeunes concernant la numération (Margolinas & Wozniak, 2012).

3.3 Activités d'informatique débranchée et représentation binaire de l'information

3.3.1 Description des activités de l'ouvrage CSU (Fellows et al., 2002)

L'activité concernant la représentation binaire des nombres est la première à avoir été conçue. Il s'agissait pour leur auteur, Tim Bell, d'expliquer des concepts essentiels de l'informatique (ici, la représentation binaire des nombres) à des enfants de 5 ans en 1992. Il fait délibérément le choix de ne pas utiliser d'ordinateur, mais de les présenter sous forme ludique et engageante pour les élèves (Bell et al., 2012).

La trame est restée la même depuis la création, même si des modifications mineures, ou culturelles selon les pays pour lesquels elles ont été adaptées, ont été apportées depuis. Sur les 12 activités proposées dans le manuel enseignant, 5 concernent la représentation des informations.

La première activité « Compter les points » commence par une démonstration devant la classe, avec 5 élèves volontaires, tenant chacun une carte avec des points sur le recto et rien sur le verso (voir illustration 1).

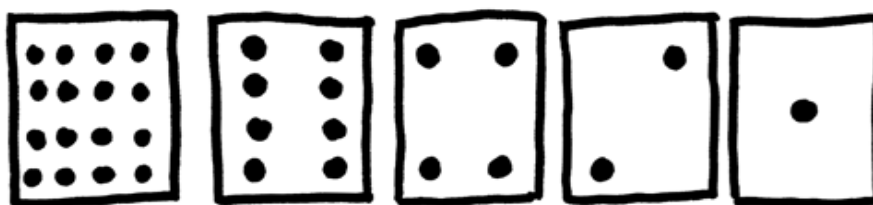


Illustration 1 : Cartes pour l'activité « Compter les points » (Fellows et al., 2002)

Des questions sont posées aux enfants :

- Que remarque-t-on concernant le nombre de points sur les cartes ?
- Combien de points devrait avoir la prochaine carte si on en ajoute une à gauche ? Et la suivante ? Etc.
- On demande ensuite de faire afficher des nombres aux enfants...

Puis l'animateur explique le fait que si la carte ne présente aucun point, elle est représentée par un 0 en binaire, par un 1 si elle a des points (voir illustration 2). On propose d'afficher des nombres en binaire et de faire trouver à la classe leur équivalent en décimal. Et inversement.

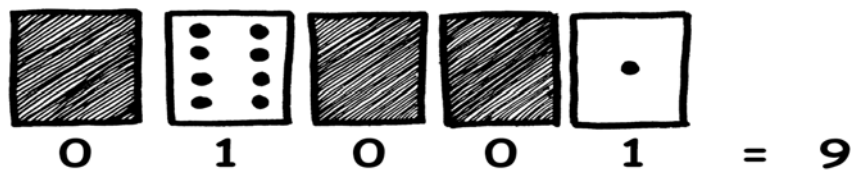


Illustration 2 : Cartes pour l'activité « Compter les points » (Fellows et ali., 2002)

L'exercice suivant consiste à faire travailler en binaire les élèves avec des jeux de carte individuels, de déchiffrer des nombres codés avec d'autres symboles. Les élèves sont ensuite amenés à envoyer des messages secrets codés en binaire, chaque signal lumineux correspondant à une lettre, à coder/décoder un message en binaire émis avec des sons aigus ou graves pour simuler le fonctionnement d'un modem. L'exercice suivant propose à compter au-delà de 31 avec ces doigts (chaque doigt représente une position, le doigt levé correspond à 1, baissé à 0).

L'activité 2 (la couleur par les nombres) aborde la question du stockage, et donc du codage, d'une image par l'ordinateur. Elle introduit la notion de pixels. Les enfants doivent dessiner dans un quadrillage des images codées en 0 et 1. ils peuvent aussi dessiner une image, la coder, transmettre le code à un camarade pour que celui-ci à son tour dessine l'image.

L'activité 3 (Peux-tu répéter ?) introduit la notion de compression de texte. Les enfants sont amenés à repérer dans un texte les groupes de lettres ou les mots qui se répètent, puis à supprimer les répétitions en les matérialisant par des cases vides et des flèches indiquant le motif à reproduire (voir illustration 3).

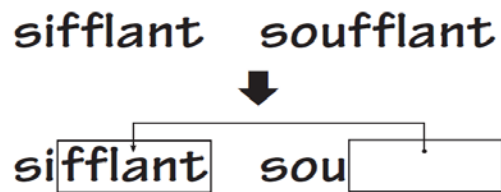


Illustration 3 : Formalisme pour la compression. Activité 3 (Fellows et ali., 2002)

L'activité 4 (Tour de cartes) présente sous forme de magie le contrôle de parité, permettant de détecter les erreurs et de corriger les erreurs survenues lors de la transmission des données.

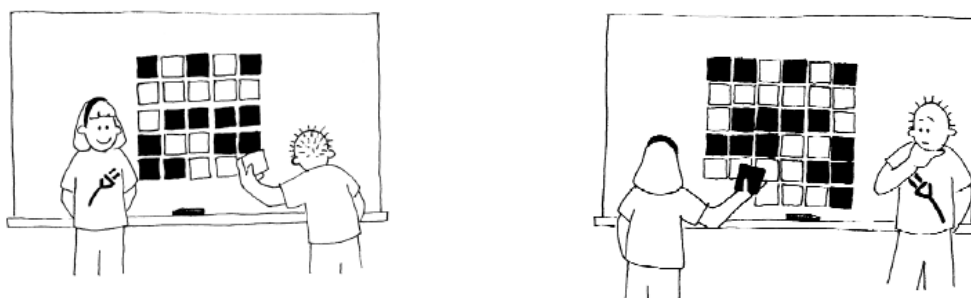


Illustration 4 : Tour de magie. Activité 4 (Fellows et ali., 2002)

Un élève dispose des cartes en carré de 5x5 au tableau. L'enseignant ajoute une sixième ligne et colonne « pour compliquer les choses » (voir illustration 4). Il se détourne et demande à l'élève de retourner une seule carte. L'enseignant alors devine quelle est la carte retournée. Le tour repose sur la ligne et la colonne ajoutées par l'enseignant qui permettent d'avoir un nombre pair de cartes colorées et donc de retrouver la carte retournée. L'astuce est donnée aux élèves qui doivent utiliser par petits groupes un jeu de cartes similaire et réaliser le « tour de magie ». Des exercices

complémentaires sont proposés avec d'autres codes (ISBN, code barre, etc).

3.3.2 Observation de séances

Nous avons mené deux séances dans deux classes différentes de CM1/CM2 de 24 élèves chacune. Il ne s'agissait pas de mettre en place un protocole expérimental, mais de tester les activités 1 (45 minutes) et 4 (1 heure 30) en vue d'une formation future pour des enseignants. L'auteur a conduit l'activité 1 dans une classe, une autre intervenante a conduit l'activité 4 dans l'autre classe, pendant que l'auteur l'observait.

Le déroulement de l'activité 1 (45 mn) a consisté d'abord à faire les démonstrations avec l'aide de cinq élèves volontaires et les cartes à points puis des exercices collectifs de conversion. Ensuite, une fiche mettant en œuvre un algorithme de conversion du décimal en binaire a été distribuée et expliquée aux élèves. Ils l'ont utilisé pour convertir des nombres, qui correspondaient à une « définition » d'une grille de mots croisés. Les élèves devaient ensuite colorier chaque case qui contenait un 1 pour visualiser une image. Les élèves ont dans l'ensemble réussi cette activité.

Dans l'apprentissage des nombres, des cartes à points sont parfois utilisées, mais elles permettent de visualiser les nombres eux-mêmes (voir par exemple l'illustration 5).

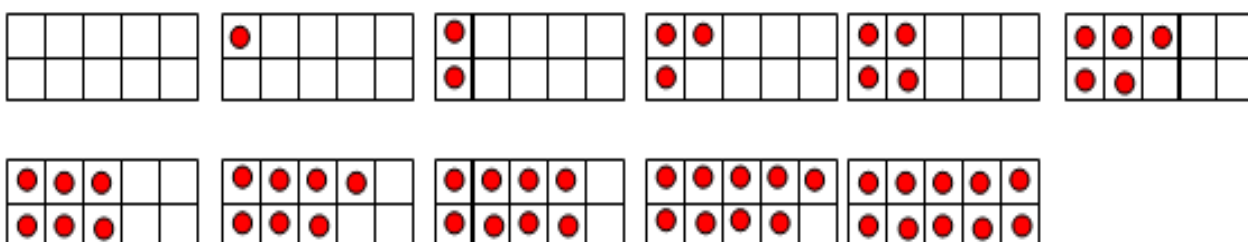


Illustration 5 : Cartes à points, d'après Bregeon (2003)

Or, les travaux de Herman, Zilles et Loui (2011) montrent que mêmes des élèves assez âgés ont des difficultés dans la conversion entre les différentes bases utilisées en informatique (binaire et hexadécimale), en particulier avec des confusions sur le poids de la position la plus à droite (2 au lieu de 1 en base 2 par exemple). Le choix fait dans cette activité de représenter le poids de la position (voir illustration 3) et non le nombre lui-même permet de réduire les erreurs. Ce que semble montrer les travaux de Kordaki, dans un autre environnement. Kordaki (2011) a créé un jeu de cartes informatique pour l'apprentissage des aspects simples du système de représentation binaire des nombres auprès d'élèves d'école primaire. Le système permet la conversion d'un nombre décimal en binaire et inversement, avec ou sans aide, figurée sous forme de cartes à points, selon la logique mise en œuvre dans les activités *CSU*.

Dans l'activité 4, pour rendre les élèves plus actifs, l'intervenante avait choisi de ne pas « donner l'astuce » mais de la faire découvrir aux élèves. Les élèves ont été amené à travailler par groupe de 4, chaque groupe disposant d'un jeu de cartes leur permettant de reproduire le tour de magie. L'intervenante a conçu un document qui permettait aux élèves de garder la trace de la disposition des cartes avant et après l'introduction de l'« erreur ». Une septième colonne a été ajoutée à droite demandant d'indiquer le nombre de cartes noires et blanches. De même une septième ligne en dessous du tableau permettait aux élèves d'indiquer le nombre de cases blanches et noires de la colonne. Ce tableau était à compléter par les élèves une première fois avant la modification et une seconde fois après la modification. L'intervenante supposait qu'il serait alors plus facile pour les élèves de réfléchir et de faire des hypothèses.

Le déroulement de la séance a montré que si les élèves étaient incités par le document à compter les cartes noires et blanches, peu ont fait d'eux-mêmes le lien avec la quantité paire ou impaire des cartes. De même, lors de la séance, il s'est avéré que peu d'élèves ont réussi à trouver le rôle joué par la colonne et la ligne ajoutées par l'enseignant. Lors de la phase de validation, l'intervenante a demandé à un des élèves qui avait réussi à faire le tour d'expliquer aux autres en quoi il consistait. Alors même qu'il avait été capable de faire le tour à plusieurs reprises, la formulation a été difficile pour lui. A la suite des explications données par l'intervenante, tous les élèves étaient à même d'effectuer le tour. Certains élèves ont introduit d'eux-mêmes des variantes : au lieu de changer une seule carte, ils en ont changé deux, voire trois. Une discussion

avec eux a été engagée pour savoir s'ils pensaient que toutes les erreurs étaient détectables. Les élèves pensaient que oui. Un élève n' a changé aucune carte, laissant ses camarades dans la perplexité ...

L'une des difficultés rencontrées par les intervenantes est l'introduction de la séance auprès des élèves. Comment donner du sens pour eux à ce sujet, qui n'a pas forcément de résonance immédiate dans leur vie quotidienne ? Les élèves ont accepté sans difficulté le sujet et se sont montré intéressés et attentifs lorsqu'il leur a été annoncé qu'ils allaient travailler des notions d'informatique. On peut imaginer que le fait que la séance ne soit pas menée par l'enseignant habituel et a été observée par une autre personne, à jouer un rôle dans cet intérêt.

3.3.3 Évaluation dans la littérature de ces activités

Des chercheurs ont mené ces activités dans des classes et les ont évaluées. Les principaux résultats obtenus sont les suivants.

Chiang et ali. (2008) proposent à des élèves canadiens de 5 classes différentes une leçon mêlant des activités mathématiques et informatique. Les auteurs évaluent les sentiments des élèves concernant l'informatique après la séance sur la représentation binaire des nombres. Plus de 80 d'entre eux (sur 134 répondants) ont fortement aimé la séance, environ 45 ont aimé, les autres ont moyennement aimé.

L'une des activités proposées par Thies et Vahrenhold (2013) concernait les nombres binaires. Les auteurs ont comparé la réussite des élèves ayant utilisé le matériel pédagogique *unplugged* et ceux ayant utilisé les manuels traditionnels à des exercices de conversion du binaire vers le décimal et inversement. Les résultats obtenus ne montrent pas de différence statistique significative concernant la réussite aux tests des deux groupes.

Nishida et ali. (2008) après chaque séance de la première expérience (voir tableau 3) demandent aux 28 élèves s'ils ont trouvé la séance plaisante. Les résultats sont présentés dans le tableau 7. Dans la troisième expérience, 9 élèves sur 10 disent avoir compris le contrôle de parité (activité 4).

Chapitre	Titre	Activité	4	3	2	1
2	La couleur par les nombres	Représentation des images	55,3	38,8	5,8	0
3	Peux-tu répéter ?	Compression de texte	39,1	45,7	15,2	0
4	Tour de cartes	Détection et correction des erreurs	34	48,9	17	0
1	Compter les points	Écriture binaire des nombres	41,7	43,8	12,5	2,1

Tableau 7: Satisfaction des élèves en % pour chacune des activités (Nishida et ali. 2008)

Légende : 4 : Plaisant, 3 : Relativement plaisant, 2 : Relativement peu plaisant, 1 : Pas plaisant

4. Discussion

Au regard de l'analyse de la littérature, tirer des conclusions des différents travaux concernant l'apport des activités débranchées à l'apprentissage de l'informatique s'avère un exercice délicat. Les résultats ne sont en effet pas tous concordants, concernant l'intérêt de ce type d'activité en fonction de l'âge des élèves par exemple. Les expérimentations sont diverses tant par les publics concernés, les connaissances déjà acquises par les élèves, les situations d'enseignement, la manière dont est utilisé ou modifié le matériel, la durée des séances, la personne qui mène l'activité (l'enseignant habituel ou une autre personne), etc. Bref, beaucoup de variables rendent les conclusions difficilement généralisables. De plus, les résultats de certaines expérimentations sont lacunaires : nombre d'élèves concernés, taux de réponse etc. sont parfois manquants.

De même, les résultats concernant principalement les déclarations des élèves sur leur motivation, leur intérêt, leur plaisir. Cette orientation des recherches peut aisément se comprendre dans le

cadre d'une sensibilisation – ce qui est l'objectif premier des activités telles qu'elles ont été conçues -, mais n'est plus suffisante si ces activités sont amenés à être incluses dans un curriculum officiel. Soulignons à nouveau que ces activités n'ont pas été pensées pour cet usage. Il conviendrait alors de s'interroger sur les conséquences de ce changement de statut sur la conception de ce matériel. Des activités éparses, ne couvrant pas l'ensemble des sujets possibles, devraient être articulées pour donner du sens. Il ne s'agit pas ici de prétendre que tout un enseignement d'informatique devrait se faire sans ordinateur, cela n'aurait pas de sens, et serait particulièrement démotivant pour les élèves ! Les fondateurs eux-mêmes de ce programme plaident pour la complémentarité des approches. L'expérience montre que pour mener une activité de ce type, les connaissances de l'enseignant doivent être suffisantes pour faire face aux questions et remarques des élèves, qui ne manqueront pas de survenir (un exemple parmi d'autres, lors de l'activité 1, un élève demande si on peut représenter des nombres à virgule).

Le question du sens à donner à une activité s'est posée lors des deux séances. C'est aussi une question soulevée par (Nishida et al., 2009). Les auteurs soulignent que pour l'activité 4, certes les élèves comprennent que le contrôle de parité permet de détecter et éventuellement de corriger des erreurs, mais que le besoin de ce contrôle n'est pas immédiatement apparent. Ce qui peut poser problème avec des élèves assez jeunes pour qui, selon les auteurs, l'activité sera leur principale expérience et que les explications que l'on pourra leur apporter n'auront pas le même impact.

Enfin, il nous semble que l'analyse de ces activités peut apporter à la didactique de l'informatique, qui se consacre essentiellement, et ce qui est nécessaire, à l'apprentissage de la programmation et à l'utilisation d'environnement tels que Scratch, Alice, et Baltie, à l'utilisation de jouets programmables, mais finalement assez peu à d'autres concepts et méthodes de la science informatique. Le développement de jeux sérieux permettant par exemple l'apprentissage de la représentation binaire des nombres est une des pistes ouvertes. Si les travaux de Kordaki (2011) s'inscrivent dans la lignée de l'approche du binaire comme système de numération, d'autres travaux cherchent à construire des jeux sérieux qui, comme l'indiquait Aigle (1989), mettent en œuvre une représentation électronique du système binaire (Feaster, Ali, & Hallstrom, 2012).

5. Conclusion

Nous avons souhaité faire le point sur les apprentissages réalisés en informatique avec des activités *CSU* dans un contexte de classe. Finalement, la revue de littérature a montré que peu de travaux étaient consacrés à cette question. De plus, les travaux existants sont peu comparables et non cumulables par la diversité des situations étudiées.

Qu'apprennent les élèves avec les activités *CSU* ? Aident-elles les élèves à construire de représentations exactes des concepts de l'informatique ? Quel est le niveau de compréhension réelle des élèves ? C'est un programme de recherche qui s'ouvre, mais qui nécessite des réflexions concernant les protocoles à mettre en œuvre. La recherche sur ces apprentissages ne pourra pas faire l'impasse sur la réflexion à conduire en parallèle sur la didactique de l'informatique et ses champs conceptuels.

Références :

Aigle, M. (1989). « Vous avez dit binaire ? » In Colloque francophone sur la didactique de l'informatique, Université René Descartes Paris les 1, 2 et 3 septembre 1988 (p. 177 192). Présenté à Colloque francophone sur la didactique de l'informatique. Consulté à l'adresse <http://edutice.archives-ouvertes.fr/edutice-00362447>

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2008). Computer science without computers: new outreach methods from old tricks. In Proceedings of the 21st Annual Conference of the National Advisory Committee on Computing Qualifications.

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, 13(1), 20–29.

Bell, T., Curzon, P., Cutts, Q., Dagiene, V., & Haberman, B. (2011). Overcoming obstacles to CS education by using non-programming outreach programmes. In Proceedings of the 5th international

- conference on Informatics in Schools: situation, Evolution and Perspectives (p. 71–81). Berlin, Heidelberg: Springer-Verlag.
- Bell, T., Rosamond, F., & Casey, N. (2012a). Computer Science Unplugged and Related Projects in Math and Computer Science Popularization. In H. L. Bodlaender, R. Downey, F. V. Fomin, & D. Marx (Éd.), *The Multivariate Algorithmic Revolution and Beyond* (p. 398–456). Springer Berlin Heidelberg.
- Bell, T., Rosamond, F., & Casey, N. (2012b). Computer science unplugged and related projects in math and computer science popularization. In *The Multivariate Algorithmic Revolution and Beyond* (p. 398–456). Springer.
- Bregeon, J.-L. (2003). Les cartes à points : pour une meilleure perception des nombres. *Activités Mathématiques et Scientifiques. Les revues pédagogiques de la Mission Laïque Française*, (50), 11–20. Consulté à l'adresse : <http://www.ia94.ac-creteil.fr/math/reflexion/nombre/bregeon.pdf>
- Denning, P. J. (2010). The great principles of computing. *American Scientist*, 98(5), 369–72.
- Drot-Delange, B. (2012). Enseignement de l'informatique, éducation aux technologies de l'information et de la communication en France, dans l'enseignement général du second degré. *Spirale*, 50, 25–37.
- Feaster, Y., Ali, F., & Hallstrom, J. O. (2012). Serious toys: teaching the binary number system. In *Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education* (p. 262–267).
- Feaster, Y., Segars, L., Wahba, S. K., & Hallstrom, J. O. (2011). Teaching CS unplugged in the high school (with limited success). In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education* (p. 248–252).
- Fellows, M., Bell, T., & Witten, I. (2002). Computer science unplugged. *Computer Science Unplugged*.
- Fellows, M. R., Bell, T., & Witten, I. (1996). *Computer Science Unplugged... offline activities and games for all ages: Original Activities Book*. Computer Science Unplugged.
- Futschek, G., & Moschitz, J. (2011). Learning algorithmic thinking with tangible objects eases transition to computer programming. In *Informatics in Schools. Contributing to 21st Century Education* (p. 155–164). Springer.
- Goldman, K., Gross, P., Heeren, C., Herman, G. L., Kaczmarczyk, L., Loui, M. C., & Zilles, C. (2010). Setting the Scope of Concept Inventories for Introductory Computing Subjects. *Trans. Comput. Educ.*, 10(2), 5:1–5:29.
- Greff, É. (1998). Le « jeu de l'enfant-robot » : une démarche et une réflexion en vue du développement de la pensée algorithmique chez les très jeunes enfants. *Revue Sciences et Techniques Educatives*, 5(1), 47–61.
- Herman, G. L., Zilles, C., & Loui, M. C. (2011). How do students misunderstand number representations? *Computer Science Education*, 21(3), 289–312.
- 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. In *Sciences et technologies de l'information et de la communication en milieu éducatif: Analyse de pratiques et enjeux didactiques. Actes du quatrième colloque international DIDAPRO 4-Dida&Stic*, 24–26 octobre 2011, Université de Patras. (p. 271–281). Consulté à l'adresse : <http://edutice.archives-ouvertes.fr/edutice-00676143>
- Kordaki, M. (2011). A computer card game for the learning of basic aspects of the binary system in primary education: Design and pilot evaluation. *Education and Information Technologies*, 16(4), 395–421.

- Margolinas, C., & Wozniak, F. (2012). *Le nombre à l'école maternelle. Une approche didactique*. Bruxelles: De Boeck.
- Nishida, T., Idosaka, Y., Hofuku, Y., Kanemune, S., & Kuno, Y. (2008). New Methodology of Information Education with « Computer Science Unplugged ». In R. T. Mittermeir & M. M. Sysło (Éd.), *Informatics Education - Supporting Computational Thinking* (p. 241-252). Springer Berlin Heidelberg.
- Nishida, T., Kanemune, S., Idosaka, Y., Namiki, M., Bell, T., & Kuno, Y. (2009). A CS unplugged design pattern. In *ACM SIGCSE Bulletin* (Vol. 41, p. 231–235).
- Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. Basic Books, Inc.
- Thies, R., & Vahrenhold, J. (2013). On plugging « unplugged » into CS classes. In *Proceeding of the 44th ACM technical symposium on Computer science education* (p. 365–370). New York, NY, USA: ACM.
- Tort, F. & Dagiene, V. (2012). *Concours Castor : découvrir l'informatique autrement. E-Dossier de l'audiovisuel : l'éducation aux cultures de l'information*.
- Tort, F., & Drot-Delange, B. (2013). Informatics in the French Secondary Curricula: Recent Moves and Perspectives. In I. Diethelm & R. T. Mittermeir (Éd.), *Informatics in Schools. Sustainable Informatics Education for Pupils of all Ages* (Vol. 7780, p. 31-42). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Tucker, A. (2003). *A model curriculum for K–12 computer science. Final report of the ACM K–12 task force curriculum committee*.