
BONNE DONNE OU MALDONNE, VERS LA CORRECTION PARTIELLE D'UN ALGORITHME¹

Mickaël BARRAUD²

Lycée Jean de Lattre De Tassigny, La Roche-sur-Yon

Annaïck BROCARD³

Lycée Nicolas Appert, Orvault

Pascal CHAUVIN⁴

Lycée François Truffaut, Challans

Jean-Anne COLOMBEL⁵

Lycée Saint-Joseph, Machecoul

Emmanuel DESMONTILS⁶

Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France

Jean-Marie LAMBERT⁷

Lycée Bellevue, Le Mans

Emmanuel LEMAITRE⁸

Collège Pierre de Gennes, Le Mans

Résumé. L'algorithmique est une des entrées du programme officiel de NSI. Il y est inscrit la nécessité de prouver la correction d'un algorithme. Il est attendu d'un élève la capacité à écrire un invariant de boucle et de montrer la terminaison à l'aide d'un variant de boucle. Le sujet de cet article est de proposer une situation didactique permettant d'introduire simplement ces notions pour des élèves de première. Il s'agit de décrire comment des élèves, à travers une activité débranchée, peuvent être amenés à un travail plus théorique autour de la correction d'algorithme. Les élèves travaillent sur un algorithme de division euclidienne, sans qu'il soit formulé ainsi et sans lien direct avec l'arithmétique ni son implémentation : il leur est demandé de réfléchir à la distribution de cartes. Les post-conditions sont introduites en définissant la « bonne donne » et permettent d'engager le travail de correction d'algorithme. Deux mises en œuvre en classe nous ont fait nous focaliser sur la correction partielle. Deux autres mises en œuvre nous permettent une première analyse et d'envisager certaines améliorations.

Mots-clés. Informatique débranchée, Didactique de l'informatique, Algorithmique, Situation didactique, Correction partielle d'algorithme, Invariant, Post-condition

¹ Cet article est le résultat d'un travail effectué par le groupe EIEM de l'IREM des Pays de la Loire (<https://irem.univ-nantes.fr/>) entre 2020 et 2023.

² Mickaël.Barraud@ac-nantes.fr

³ Annaick-Marie.Brocard@ac-nantes.fr

⁴ Pascal.Chaudin@ac-nantes.fr

⁵ Jean-Anne.Colombel@saintjoseph-machecoul.fr

⁶ Emmanuel.Desmontils@univ-nantes.fr

⁷ Jean-Marie.Lambert@ac-nantes.fr

⁸ Emmanuel-J.Lemaitre@ac-nantes.fr

Abstract. Algorithmics is one of the entries in the official NSI program. It states the need to prove the correctness of an algorithm. A student is expected to be able to describe a loop invariant and show the termination using a loop variant. The subject of this article is to propose a didactic situation, allowing these concepts to be simply introduced to first-grade students. This involves describing how students, through an unplugged activity, can be led to more theoretical work around algorithm correction. The students work on a Euclidean division algorithm without it being formulated like this and without a direct link to arithmetic or its implementation: they are asked to think about the distribution of cards. The post-conditions are introduced by defining the “good deal” and allow the algorithm correction work to begin. Two classroom implementations made us focus on partial correction. Two other implementations allow us to conduct an initial analysis and consider some improvements.

Keywords. Computer Science Unplugged, Didactics of Computer Science, Algorithms, Didactic Situation, Partial Algorithm Correction, Invariant, Post-condition

Introduction

Depuis quelques années, la science informatique et la pensée informatique (Wing, 2006 ; Wing, 2008 ; Selby & Woollard, 2014) ont fait leur entrée dans les programmes en collège et au lycée, d’abord en mathématiques, ensuite à travers l’option ISN (Informatique et science du numérique), puis la spécialité NSI (Numérique et Science Informatique) en 2019. Une partie importante et parfois complexe pour un débutant en informatique s’attache à considérer les algorithmes (ou les programmes) en tant qu’objet d’étude (Modeste *et al.*, 2010), en particulier à travers les notions de complexité (spatiale et temporelle), de pré-condition, de post-condition, d’invariant, de terminaison et, plus généralement, de preuve d’algorithme.

Le programme de NSI propose, en particulier en première et dans le cadre d’algorithmes précis, d’aborder les notions d’invariant, de terminaison et de correction. Il est donc demandé aux enseignants de montrer l’intérêt de prouver la correction partielle d’un algorithme pour lequel on dispose d’une spécification précise, en mobilisant la notion d’invariant. La terminaison d’un algorithme doit également être mise en évidence dès l’utilisation d’une boucle non bornée (ou, en terminale, de fonction récursive) grâce à la mobilisation de la notion de variant.

Les notions de variant et d’invariant permettant la correction d’algorithmes sont pointées, dans le programme officiel, sur les algorithmes de tri et de recherche dichotomique (tableau 1). Le programme officiel de NSI indique :

Il est nécessaire de montrer l’intérêt de prouver la correction d’un algorithme pour lequel on dispose d’une spécification précise, notamment en mobilisant la notion d’invariant sur des exemples simples. La nécessité de prouver la terminaison d’un programme est mise en évidence dès qu’on utilise une boucle non bornée (ou, en terminale, des fonctions récursives) grâce à la mobilisation de la notion de variant sur des exemples simples. (MEN, 2019a, annexe p. 9)

Ces algorithmes ayant, pour les élèves, des difficultés intrinsèques, il nous semble qu’introduire les notions de variant et d’invariant doit se faire au préalable, « sur des exemples simples ». D’autre part, ces algorithmes de tri et de recherche sont souvent introduits à l’aide d’activités de manipulation, en particulier avec des cartes à jouer. L’idée de l’activité débranchée présentée ici est donc d’utiliser ce support (les cartes à jouer) et d’introduire ces notions (variant et invariant) à travers un algorithme simple (la distribution des cartes). Enfin, le choix a été fait de ne pas mélanger les deux notions. Cette activité est donc centrée sur la correction partielle (mais permet également très simplement d’engager le travail sur la terminaison).

Pour prouver la correction d’un algorithme, il est nécessaire de vérifier, quelles que soient les données manipulées, la terminaison et la correction partielle.

La terminaison est le fait que l’algorithme s’arrête. Dans le cas de la présence de boucles, on utilise un variant de boucle. C’est une expression entière, positive et qui décroît strictement à

chaque itération. Elle dépend des variables de la boucle et permet de prouver que l'exécution de la boucle se termine.

La correction partielle vérifie qu'à partir d'entrées vérifiant les pré-conditions, l'algorithme produit correctement des sorties vérifiant les post-conditions. Une pré-condition est une assertion attachée à l'algorithme qui doit être vraie avant son exécution. Une post-condition est une assertion qui doit être vraie après terminaison de l'algorithme et qui garantit que les résultats sont conformes aux attentes. Dans le cas de la présence de boucles, on utilise un invariant de boucle. C'est une expression logique qui dépend des variables de la boucle et qui doit être vraie avant la boucle et après chaque itération de la boucle.

Contenus	Capacités Attendues	Commentaires
Tris par insertion, par sélection	Écrire un algorithme de tri. Décrire un invariant de boucle qui prouve la correction des tris par insertion, par sélection.	La terminaison de ces algorithmes est à justifier. On montre que leur coût est quadratique dans le pire cas.
Recherche dichotomique dans un tableau trié	Montrer la terminaison de la recherche dichotomique à l'aide d'un invariant de boucle .	Des assertions peuvent être utilisées. La preuve de la correction peut être présentée par le professeur.

Tableau 1. Programme de numérique et sciences informatiques de première générale - section Algorithmique (MEN, 2019a, Annexe, extrait).

Le programme de NSI propose d'aborder ces notions dans la section Langages et programmation dans le cadre de la spécification de fonctions (tableau 2).

Le programme officiel de NSI indique alors :

L'importance de la spécification, de la documentation et des tests est à présenter, ainsi que l'intérêt de la modularisation qui permet la réutilisation de programmes et la mise à disposition de bibliothèques. Pour les programmes simples écrits par les élèves, on peut se contenter d'une spécification rapide mais précise. (MEN, 2019a, annexe p. 8)

Le programme de terminale NSI (MEN, 2019b) propose quant à lui d'aborder les questions de calculabilité et de décidabilité. Nous n'aborderons pas ces questions ici.

Contenus	Capacités Attendues	Commentaires
Spécification	Prototyper une fonction. Décrire les préconditions sur les arguments. Décrire des postconditions sur les résultats.	Des assertions peuvent être utilisées pour garantir des préconditions ou des postconditions .

Tableau 2. Programme de numérique et sciences informatiques de première générale - section Langages et programmation (MEN, 2019a, Annexe, extrait).

Dans ce travail, nous cherchons donc à montrer qu'il est possible d'introduire assez simplement certaines notions importantes pour l'enseignement de la preuve d'algorithme comme demandé dans le programme (MEN, 2019a), en particulier les notions de pré/post-condition et la notion d'invariant. Pour cela, nous avons choisi de proposer une activité d'informatique débranchée en utilisant des cartes à jouer classiques.

L'objectif de cet article est de décrire l'activité que nous proposons pour l'introduction de ces notions et sa mise en œuvre dans le cadre de la classe de première NSI. Nous allons d'abord poser le contexte théorique de notre travail en section 1. Puis, en section 2, nous décrirons l'activité proposée. En section 3, nous commenterons la mise en œuvre et proposerons une analyse avant de conclure en section 4.

1. Contexte théorique

Pour satisfaire nos objectifs, nous nous sommes tournés vers une approche débranchée et avons envisagé la situation sous l'angle de la théorie des situations didactiques de Brousseau (1998). Une situation didactique est un ensemble de conditions et de circonstances (appelé milieu) plus ou moins accompagné par l'enseignant, ayant pour objet de favoriser l'apprentissage par un élève d'une connaissance nouvelle. Elle intègre une progression pédagogique structurée et des rétroactions du milieu en fonction des actions des élèves. Elle favorise l'émergence de questions, d'interrogations et de débats entre les élèves. Elle intègre parfois des moments de régulation par l'enseignant pour faciliter la compréhension et aider les élèves à progresser. Lorsque l'enseignant n'intervient pas, la situation est alors dite « adidactique ». Un composant important d'une situation (a)didactique est la notion de variable didactique. C'est un élément qui peut être modifié pour provoquer des comportements différents chez l'élève, et donc renforcer ou étendre l'apprentissage.

L'informatique débranchée ou informatique sans ordinateur (Crozet & Gropserrin, 1987 ; Drot-Delange, 2013 ; Bell *et al.*, 1998 ; Nishida *et al.*, 2009 ; Alayrangues *et al.*, 2017) est très en vogue depuis une quinzaine d'années pour l'enseignement de l'informatique. Nishida, Kanemune, Idosaka, Namiki, Bell et Kuno (2009) proposent la définition de l'informatique débranchée suivante :

"Computer Science (CS) Unplugged" is an educational method for introducing non-specialists to concepts of CS through hands-on activities that don't require the use of a computer.
(Nishida *et al.*, 2009, p. 231)

Autrement dit, l'informatique débranchée est une méthode éducative permettant d'introduire, pour des non-spécialistes, des concepts d'informatique à travers des activités manuelles qui ne nécessitent pas d'ordinateur. Pour les auteurs, ce genre d'activité doit comporter les caractéristiques suivantes :

1. sans ordinateur ;
2. ludique ;
3. kinesthésique (la manipulation d'objets physiques est préférable) ;
4. gérés par les étudiants/élèves ;
5. facile à mettre en œuvre ;
6. extensible ;
7. conte une histoire.

Nous pouvons noter que la caractéristique 4 s'aligne avec la notion de situation adidactique telle que définie par Brousseau (1998).

Comme le dit Jeannette Wing (2008) : « computational thinking does not require a machine ». Ce type d'activité permet de se détacher de l'aspect programmation et matériel de l'informatique pour se concentrer sur des concepts particuliers. Cette approche est particulièrement intéressante dans notre cadre d'étude d'algorithme. Dans ce cadre, il existe de nombreuses approches sur papier (Modeste *et al.*, 2010 ; Meyer & Modeste, 2020 ; Meyer & Modeste, 2022), en jeux de rôle (Crozet & Gropserrin, 1987) ou à l'aide de matériel tangible comme des aiguilles à tricoter (Duflo-Kremer, s. d.), des cartes (Bell *et al.*, 1998 ; Alayrangues *et al.*, 2017), des gobelets, des Legos ou des ficelles (Alayrangues *et al.*, 2017). Nous allons un peu plus loin que des activités sur papier, puisque nous proposons la manipulation de cartes à jouer. Notre approche, parfois un peu abstraite dans sa présentation habituelle, est introduite avec un outil très familier pour beaucoup d'élèves. Il n'y a donc pas de phase d'appropriation du matériel pédagogique ni de la démarche du « jeu ». Elle répond aux différentes caractéristiques proposées par Nishida, Kanemune, Idosaka, Namiki, Bell et Kuno (2009), excepté le point 7 à propos de l'histoire contée. Cependant, ce

dernier point est surtout important pour des élèves plus jeunes (primaire ou collège) et l'est moins pour ceux qui sont la cible de notre travail (des lycéens de première).

La manipulation des cartes apporte une facette kinesthésique à notre situation. Cette approche encourage les étudiants à s'impliquer davantage dans la matière et les uns avec les autres. Favoriser un environnement d'apprentissage collaboratif et interactif peut être particulièrement bénéfique (Mobley & Fisher, 2014). L'approche kinesthésique permet aux étudiants d'aborder des concepts informatiques par le biais d'activités physiques, de jeux et de tours de magie, rendant ainsi le processus d'apprentissage interactif et agréable (Sellarès & Toussaint, 2003 ; Bell *et al.*, 2009).

À l'instar de travaux précédents (Modeste *et al.*, 2010 ; Meyer & Modeste, 2020 ; Meyer & Modeste, 2022), nous pouvons adopter la théorie des situations didactiques de Brousseau (1998) comme cadre d'analyse de l'activité. L'enseignant accompagne les groupes pour les amener à comprendre les notions visées. Dans le cadre de cette expérimentation, nous proposons une situation didactique, c'est-à-dire avec une intervention explicite de l'enseignant, basée sur la manipulation de cartes. Il n'est pas attaché d'importance à la couleur et à la valeur des cartes. Cependant, l'utilisation des cartes est intéressante dans un souci de cohérence avec d'autres activités. En particulier, les valeurs des cartes seront utilisées lors de l'introduction aux algorithmes de tri pour lesquels les notions de preuve et de terminaison seront à nouveau abordées (MEN, 2019a).

2. Présentation de la situation et analyse *a priori*

Dans les ressources NSI (ouvrages, manuels, web), les notions de variant et d'invariant ne sont pas abordées ou simplement décrites en quelques lignes. Malgré la complexité du sujet, il n'existe pas d'activités spécifiques dans le cadre de NSI. La situation didactique que nous proposons vise à introduire les notions de variant, d'invariant, de pré-condition et de post-condition afin d'amener à des réflexions sur la correction d'algorithmes. Elle est destinée à des élèves de première suivant la spécialité NSI en France (entre 16 et 17 ans).

Le principe est de travailler sur la distribution de cartes à un ensemble de joueurs. L'activité est prévue pour une séance d'une heure. Le rôle de l'enseignant est de s'assurer du bon déroulement des distributions et d'amener les élèves à déterminer les concepts visés.

Notre situation didactique utilise des cartes à jouer comme support. Ce choix comporte plusieurs avantages. Tout d'abord, le matériel pédagogique est facile à trouver, à stocker et peu coûteux. Il peut être utilisé dans beaucoup d'autres situations didactiques comme le tri ou la recherche dichotomique, par exemple. De plus, un jeu de cartes est facile à manipuler et assez peu fragile. Pour terminer, c'est un outil que beaucoup d'élèves connaissent bien dans le cadre de différents jeux, ce qui peut, nous le verrons, amener à contextualiser la distribution.

Avant toute chose, l'enseignant doit donc se munir de paquets de cartes à jouer (1 paquet de 32 ou 52 cartes par groupe d'élèves), qui servent de support à la réflexion des élèves tout au long de l'activité. Les cartes peuvent être de différentes natures puisque nous ne les utilisons pas pour leur valeur, mais simplement dans le but de les distribuer. Il faut également répartir les élèves en groupes. Il est plus simple, pour engager les discussions, que tous les groupes n'aboutissent pas à une distribution identique. Il est donc préférable de former des groupes de trois à cinq élèves et de donner à chacun un paquet de cartes à jouer. La diversité des configurations sur le nombre d'élèves dans un groupe et le nombre de cartes à distribuer vise à permettre à la classe de converger plus rapidement vers des résultats généraux pour les post-conditions et, plus tard, l'invariant. Il faut éviter d'avoir uniquement les configurations où le nombre de cartes est un

multiple du nombre d'élèves dans le groupe. Les variables didactiques sont donc le nombre de participants dans un groupe et le nombre de cartes à distribuer. Ces variables vont pouvoir faire émerger chez l'élève les concepts de variant et d'invariant.

L'activité se décompose en trois phases permettant d'introduire respectivement les notions de post-condition, d'invariant et de correction partielle (tableau 4) :

- une première distribution complète ;
- une seconde distribution interrompue ;
- la fin de la seconde distribution après échange des groupes.

2.1. Phase 1 : première distribution

Dans cette première phase, nous cherchons à présenter l'algorithme, mais surtout à introduire la notion de post-condition dans une situation où cette condition n'est pas trop complexe. Dans notre cas, comment doit être le jeu une fois que toutes les cartes ont été distribuées tout en respectant l'équité ? La vérification des conditions est aisée puisque les conditions de distribution sont optimales (un seul distributeur et distribution sans interruption).

La séance commence donc par une première distribution des cartes. La distribution est menée jusqu'au bout (figure 1), c'est-à-dire que la distribution est égalitaire, avec éventuellement un talon pour les cartes restantes. Mais aucune consigne n'est donnée a priori sur le nombre de cartes à distribuer à chacun.



Figure 1. Distribution par les élèves.

Cette ambiguïté devrait permettre de soulever la question de « bonne donne », ou, au minimum, de ce qui est attendu de la distribution. Chaque groupe est amené à formuler un algorithme de distribution d'un paquet de cartes. La classe se met ensuite d'accord sur l'algorithme qui est recopié au tableau.

Ensuite, la question est posée aux élèves de savoir si la distribution réalisée est correcte, selon l'algorithme établi précédemment. En posant cette question, nous leur demandons de définir ce qu'est une « bonne donne » (et, par opposition, une « maldonne ») et donc de décrire les post-conditions de l'algorithme. L'objectif est d'arriver à la définition suivante :

Il y a « bonne donne » si :

- chacun a obtenu autant de cartes ;
- le nombre de cartes du « talon » est strictement inférieur au nombre de joueurs.

Nous expliquons alors que nous cherchons à prouver que l'algorithme amène bien aux post-conditions énoncées.

À l'issue de cette phase, les élèves sont amenés à expliciter la post-condition de l'algorithme. D'une description approximative dans un premier temps (« toutes les cartes sont distribuées »), la classe doit arriver à une définition rigoureuse de la post-condition.

2.2. Phase 2 : seconde distribution interrompue

Après la découverte de la notion de post-condition, nous cherchons ensuite à faire émerger la notion d'invariant chez les élèves. Nous allons donc provoquer une distribution perturbée pour amener la nécessité d'avoir un critère, une condition à vérifier, permettant de s'assurer que la distribution se déroule bien à tout moment, c'est-à-dire à chaque tour.

Pour cela, nous lançons une seconde distribution que nous interrompons avant qu'elle se termine, mais à la fin d'un tour complet de distribution. Nous demandons alors aux élèves si, dès ce stade de la distribution, nous pouvons dire qu'il y a déjà maldonne ou si, pour l'instant, la distribution se passe sans erreur. L'idée, ici, est de faire formuler aux élèves qu'à chaque tour de distribution, les joueurs ont le même nombre de cartes et que le nombre total de cartes est constant.

Il faut penser à indiquer que c'est bien l'égalité des comptes de chacun, et pas leur nombre de cartes, qui valide la réponse : chaque groupe n'en est pas au même tour de distribution, mais peut conclure de la même façon.

Nous introduisons à ce moment de l'activité le mot invariant. C'est l'occasion d'esquisser un premier tableau de suivi de la situation (tableau 3). Les groupes adaptent le nombre de colonnes du tableau à leur situation.

Nombre de cartes du	joueur 1	joueur 2	joueur 3	talon
Entre deux tours de distribution	n	n	n	r

Tableau 3. Suivi de la situation de la distribution.

Selon le déroulement de cette phase, il se peut qu'un second invariant ait émergé : le nombre total de cartes. Sinon, c'est l'occasion de le mettre en valeur. Si la remarque ne vient pas, nous faisons aussi remarquer aux élèves que la proposition « le nombre de cartes du talon est inférieur au nombre de joueurs » n'est pas vérifiée en cours de distribution.

Nous demandons ensuite aux élèves de changer de table de jeu. Chaque groupe se retrouve alors à la place d'un autre, qui a laissé en place son paquet partiellement distribué. Nous leur demandons alors de faire un nouveau tour de distribution (en respectant le nombre de tas présents sur la table de jeu). Puis, nous demandons s'ils peuvent, sans toucher aux cartes cette fois, affirmer que la distribution est toujours bonne. Nous voulons, à ce moment, décrire plus finement un tour de distribution et contrôler à chaque étape l'invariant énoncé précédemment.

Pour expliciter le mécanisme de l'invariant de boucle aux élèves qui ne connaissent pas le raisonnement par récurrence, l'enseignant peut subtiliser quelques cartes d'un paquet au moment des changements de place. Cette astuce permettra, lors de la discussion suivant la fin de la phase 2 (le détail d'un tour de distribution), de montrer aux élèves que l'hérédité ne suffit pas à prouver que l'invariant est vrai pour toutes les itérations de la boucle. Effectivement, si on est capable de montrer qu'une itération de la boucle ne modifie pas l'invariant, il faut encore être certain qu'il est vrai avant cette itération.

À l'issue de cette étape, nous donnons une définition d'un invariant de boucle : « *Un invariant de boucle est une proposition : vraie avant la première itération de la boucle et qui reste vraie à la fin d'une itération si elle l'était au début.* »

Nous faisons également remarquer deux choses :

- L'invariant n'a pas besoin de rester vrai à tout moment dans la boucle.
- L'invariant est un prédicat (une proposition booléenne) sur des valeurs qui, elles, peuvent varier.

La conséquence importante de cette définition est qu'« un invariant est donc automatiquement vrai quand la boucle se termine ». On vient donc de réduire la correction partielle de l'algorithme à trouver et justifier un « bon » invariant. On rejoint ainsi les attendus du programme de 1^{re} NSI qui demandent aux élèves de savoir « décrire un invariant de boucle qui prouve la correction » et non de réellement dérouler cette preuve.

À la fin de cette phase, la notion d'invariant de boucle est donc introduite. La situation est suffisamment simple pour que cet invariant soit assez facile à trouver et à spécifier.

2.3. Phase 3 : fin de la distribution

Sur cette dernière phase, nous cherchons à faire le lien entre l'invariant (phase 2) et la post-condition (phase 1) pour amener à la correction partielle. Nous demandons donc aux élèves de terminer la distribution et nous leur demandons comment montrer que la dernière partie de la post-condition, sur le nombre de cartes présentes dans le talon, est vérifiée à la fin de l'algorithme. Finalement, nous expliquons ce qu'est la correction partielle, et pourquoi elle est seulement partielle. Si on démontre que la post-condition est vérifiée quand la boucle termine, il reste à démontrer qu'elle termine effectivement. Cela fait un lien évident avec la preuve de terminaison d'un algorithme vue dans une autre activité.

À la fin de cette troisième phase, les élèves ont découvert les notions de post-condition, d'invariant et de correction d'un algorithme (tableau 4).

Étape	Activité enseignant	Activité Élève
Mise en place	Présente l'activité et fournit des paquets de cartes à jouer (32 ou 52 cartes).	Répartition en groupe (les effectifs des groupes peuvent varier de 3 à 5)
Première distribution	Lance une distribution. Conduit aux post-conditions, formalise un algorithme de distribution.	Un donneur par groupe distribue l'ensemble des cartes. Participe à définir : la « bonne donne » et un algorithme de distribution.
Seconde distribution interrompue	Lance et interrompt une distribution. « Peut-on savoir si, jusqu'ici, il y a bonne donne ? » (vers un invariant)	Un (autre) donneur distribue une partie des cartes (finit un tour entamé). Détermine un procédé de vérification et l'effectue (comptage des cartes).
Seconde distribution : suite	Supervise les changements de tables. Lance un tour de distribution. "Comment montrer qu'un nouveau tour de distribution n'entraîne pas une maldonne ?" Introduit le vocabulaire : « invariant ». Propose de suivre l'état des variables pour justifier de l'invariant sur une itération.	Les groupes permutent en laissant les cartes sur place. Un donneur réalise un tour de distribution. Propose une argumentation sans recomptage. Complète le tableau proposé.
Seconde distribution : fin	Prouve la correction partielle. Ouvre vers la notion de terminaison.	Finit la distribution. Précise sa compréhension.

Tableau 4. Synthèse de la situation.

3. Mise en œuvre et analyse *a posteriori*

Cette activité a été expérimentée sur des groupes de première NSI dans deux lycées au cours des années 2021, 2022 et 2023 : le lycée Saint-Joseph à Machecoul en Loire-Atlantique (avec respectivement 27, 16 et 13 élèves) et le lycée De Lattre de Tassigny à La Roche-sur-Yon en Vendée (avec respectivement 16, 14 et 22 élèves). Les données récoltées sont des photos de prises de notes et du tableau (dont certaines sont présentées dans cet article). Pour tous ces groupes, les élèves ont déjà l'habitude d'utiliser des objets variés (manipulations de cartes, jeux...). Initialement prévue pour une séance d'une heure, nous avons parfois prolongé la séance d'une demi-heure pour commencer des exercices d'application ou bien poursuivre sur la preuve de terminaison de l'algorithme (qui n'est pas l'objet de cet article).

3.1. Phase 1 : première distribution

L'objectif de cette étape est d'écrire un algorithme simple et sa spécification (en particulier ses post-conditions) à partir des constatations de chacun des groupes. Nous avons demandé, sans autre explication préalable, à un élève par groupe de distribuer les cartes à son groupe. Puis, nous avons noté au tableau : « *Comment peut-on être certain qu'il n'y a pas maldonne ?* »



Figure 2. Distribution initiale par les élèves.

Contrairement à notre attente, nous avons constaté que certains élèves n'ont jamais vraiment joué à un jeu de cartes, et d'autres, qu'à certains jeux pour lesquels la notion de maldonne n'est pas réellement présente. Ceci nous a obligé à formuler avec plus de précision nos attentes. Par exemple, un élève qui a surtout joué au « président », jeu pour lequel le gagnant est le premier qui se débarrasse de toutes ses cartes, a demandé : « *À quel jeu on joue ?* ». Comme la distribution, dans son groupe, apportait plus de cartes à certains qu'à d'autres, il y avait un problème d'équité. Nous lui avons proposé de nous dire s'il connaissait d'autres jeux et les conséquences de ce genre de distribution. Nous lui avons rappelé que certains jeux (« le barbu » par exemple) demandent un nombre précis de joueurs et que, dans d'autres (le tarot, par exemple), la distribution met de côté un chien. D'autres jeux ne distribuent tout simplement pas toutes les cartes et utilisent un talon.

À la question de la maldonne, la plupart des élèves comptaient leurs cartes, mais, même pour ceux qui ont laissé de côté les cartes en surnombre, personne n'a pensé à compter ces cartes pour vérifier que la distribution est finie. Ils n'ont pas pensé non plus à faire la somme pour vérifier le nombre total de cartes (sans parler de l'intégrité du paquet de départ). Par contre, certains élèves ont tout de suite parlé de division euclidienne. Nous avons donc écrit « $2 \times 6 + 8 = 20$ » et nous avons demandé si c'était bien la division euclidienne de 20 par 6 ? La condition sur le reste est alors arrivée immédiatement.

À un moment de la discussion avec les élèves sur le sujet, il a été nécessaire d'aider à trancher et de se mettre d'accord sur ce qu'on considère être une « bonne donne ». Cette décision est importante, car elle peut induire des raisonnements plus ou moins compliqués pour la correction partielle. Nous avons noté au tableau :

Il y a « bonne donne » si :

- chacun a obtenu autant de cartes ;
- le nombre de cartes du « talon » est strictement inférieur au nombre de joueurs.

Nous avons ensuite (re)précisé le vocabulaire lié à la spécification, en particulier les post-conditions et nous avons terminé cette partie en demandant un algorithme de distribution de cartes. Chaque groupe a dû se mettre d'accord sur une réponse. Sous la dictée de l'un d'eux, on a obtenu par exemple :

Tant qu'il reste des cartes, on distribue

La mise en commun a permis d'amener les élèves à une réponse plus précise :

Tant qu'il reste plus de cartes que de joueurs,
| on effectue un tour de distribution

3.2. Phase 2 : seconde distribution interrompue

L'objectif de cette étape est de conduire à l'identification d'un invariant de boucle. Nous avons donc demandé aux élèves de recommencer une nouvelle distribution. Cette fois-ci, nous avons interrompu la distribution avant la fin et nous avons demandé de ne finir que le tour de distribution en cours. Nous avons écrit au tableau : « *Peut-on savoir si, pour l'instant, la distribution est bonne, ou s'il y a déjà maldonne ?* » Une réponse est venue facilement : « *chacun compte ses cartes pour savoir s'il en a autant que les autres.* » Nous avons alors indiqué que c'est un invariant et nous avons présenté le tableau 3. L'invariant « le nombre total de cartes est constant. » est arrivé assez souvent.

Nous avons cherché ensuite à préciser la notion d'invariant de boucle en reprenant la distribution. Nous avons donc demandé aux élèves de laisser leurs cartes à leur place et de changer de table de jeu. Au profit des hésitations et des déplacements, au moment du changement de table, nous avons, comme prévu, subtilisé discrètement quelques cartes du paquet d'un joueur.

Une fois chacun à sa nouvelle place, nous avons demandé aux nouveaux donneurs de faire un tour de distribution seulement et aux autres d'observer sans toucher à rien. Nous avons noté alors : « *Peut-on savoir si la distribution est toujours bonne, sans toucher aux cartes ?* » Nous avons constaté que les élèves réfléchissent essentiellement sur le tour qui vient de se passer. Ils répondent souvent qu'« *il faut être certain de faire un tour complet et conserver le nombre de tas présents* ». Certains demandent où (sur quel joueur) la distribution s'était interrompue. Cependant, la mise en situation est suffisamment claire pour en tirer rapidement le raisonnement : « *comme chacun avait autant de cartes et en a obtenu une de plus, alors chacun en a toujours autant* ». Nous avons alors demandé (si besoin) de préciser l'algorithme au tableau pour qu'ils suivent ce raisonnement :

Tant qu'il reste plus de cartes que de joueurs,
| on distribue une carte à chacun.

Sur une séance, une proposition moins précise d'un élève « *on distribue autant de cartes à chacun* » a été discutée. Nous avons alors posé la question : « *si on distribue les cartes deux par*

deux, est-on sûr que ce sera toujours possible ? » La réponse a été, bien sûr : « *pas s'il ne reste qu'une carte par joueur au dernier tour* ». Si, d'autre part, on s'autorise à distribuer un nombre différent à chaque tour, on perd trop de contrôle sur la situation pour espérer démontrer facilement que la distribution est valide.

Pour préciser le travail sur l'invariant, nous avons demandé à l'oral : « *Peut-on dire qu'à tout moment dans la distribution, chaque joueur a autant de cartes ?* » La réponse la plus fréquente a été, bien évidemment, « *non* », car physiquement, les cartes ne sont pas distribuées simultanément à chaque joueur. Pour aider les élèves à décomposer toutes les étapes d'un tour de distribution, nous avons été obligés de leur proposer de remplir le tableau 5 (figure 3), ajustement du tableau 3.

Nombre de cartes du	joueur 1	joueur 2	joueur 3	Talon	Autant chacun ?
Début d'une itération	n	n	n	r	oui
Une carte pour Joueur1					
Une carte pour Joueur2					
Une carte pour Joueur3					

Tableau 5. Suivi la situation de la distribution pour l'ensemble du groupe.

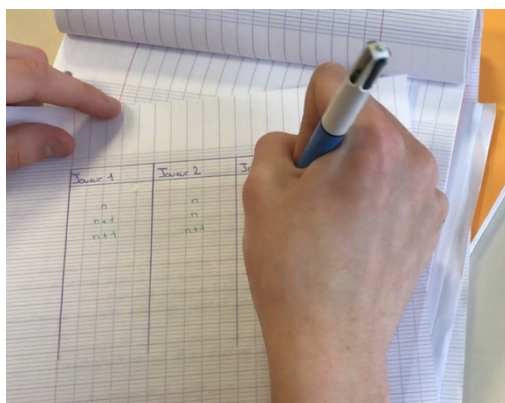


Figure 3. Un élève sur le tableau de suivi lors d'une distribution.

Il n'y a pas eu de difficulté particulière ici. On obtient $n+1$ cartes pour chaque joueur à la fin de l'itération (figure 4). Nous avons alors pris soin de préciser le vocabulaire : une proposition qui est vraie au début d'une itération de boucle, dont on peut démontrer qu'elle est de nouveau vérifiée à la fin de l'itération, est dite « invariante » pour la boucle.

Nous avons ensuite posé la question : « En admettant que le donneur respecte bien cet algorithme (au tableau), ce que l'on vient d'écrire suffit-il à démontrer que chacun aura finalement autant de cartes ? » Cette question est plus difficile. Nous avons utilisé alors la supercherie effectuée lors du changement de table. Nous avons demandé aux élèves de compter les cartes, celles de chaque joueur et celles du talon. Un groupe s'aperçoit du manque de cartes et les élèves concluent assez facilement que la donne a été faussée. La dernière vérification n'a donc pas permis de détecter la maldonne et les élèves rectifient : « *si la distribution était correcte à l'étape précédente, alors elle l'est une fois de plus.* »

On a alors remonté le raisonnement à son origine pour constater qu'à l'initialisation, si aucun joueur n'a de carte, alors la proposition est vérifiée. Ce fut le moment de formaliser, en donnant une définition plus précise de l'invariant : « *Un invariant de boucle est une proposition vraie avant la première itération de la boucle et qui reste vraie à la fin d'une itération si elle l'était au*

début » (figure 5). Un invariant est donc naturellement vrai quand la boucle se termine. Nous avons alors fait remarquer que cela permet de démontrer seulement une partie de la post-condition proposée : « chacun a obtenu autant de cartes ».

Activité : Distribution d'un jeu de cartes.

supposant que les n premières étapes de la distribution sont bien passées. Comment vérifier qu'elle se passe bien une fois de plus ?

Joueur 1	Joueur 2	Joueur 3	Joueur 4	Talon
n	n	n	n	R
$n+1$	n	n	n	$R-1$
$n+1$	$n+1$	n	n	$R-2$
$n+1$	$n+1$	$n+1$	n	$R-3$
$n+1$	$n+1$	$n+1$	$n+1$	$R-4$

On vient de montrer que si l'invariant était vrai au début d'un tour, alors il le reste vrai à la fin du tour.

→ invariant ok.

Comment peut-on vérifier que les joueurs ont pu se faire un jeu de cartes ?

→ invariant ok.

Figure 4. Bilan d'un tour de distribution et émergence de l'invariant.

L'invariant d'une boucle, permet de montrer qu'un résultat reste vrai à l'issue de la boucle.

Si l'invariant est vérifié au début de la boucle.

Et, si une itération de la boucle conserve l'invariant.

Alors l'invariant est vrai à la fin de la boucle.

Figure 5. Bilan d'un tour de distribution et émergence de l'invariant.

3.4. Phase 3 : suite et fin de la seconde distribution

L'objectif est de compléter la correction partielle et d'ouvrir vers la preuve de terminaison. Nous avons redonné les cartes subtilisées, puis nous avons demandé aux groupes de terminer la distribution. La distribution terminée, nous avons posé la question : « Comment prouver la post-condition ? » Nous avons remarqué que cette question est facilement confondue par les élèves avec une preuve de terminaison. Pour eux, comme « il y a un nombre fini de cartes », au bout d'un moment, il n'y en aura plus à distribuer. On a fait remarquer que cela signifie que la distribution se termine : c'est l'autre partie de la preuve de correction. Pour la correction partielle, il ne s'agit pas de savoir ici si ça se termine, mais si ça se termine bien. Concernant les classes dans lesquelles cette activité a été jouée, nos élèves de première suivaient majoritairement la spécialité mathématique. Comme la preuve par récurrence est au programme de cette spécialité à partir de la terminale, ce fut l'occasion d'en évoquer le vocabulaire. L'invariant est l'hypothèse de récurrence. Prouver qu'il est vrai au début de la première itération est l'initialisation ; puis, à chaque itération, qu'il reste vrai est l'hérédité.

4. Conclusion et perspectives

Dans cet article, nous avons présenté une situation didactique basée sur une activité en informatique débranchée. Cette activité se propose d'introduire des concepts utiles pour la preuve d'algorithme comme la post-condition ou l'invariant. Pour cela, nous exploitons l'activité de distribution de cartes à jouer pour un ensemble de joueurs. Nous avons mis en œuvre cette activité en deux étapes. Une première distribution normale pour introduire la notion de post-condition, puis une distribution interrompue et reprise avec permutation des groupes. Cette seconde étape nous a permis d'introduire la notion d'invariant.

Si nous reprenons les caractéristiques de Nishida *et al.* (2009), nous émergeons principalement sur 4 critères (sans ordinateur, kinesthésique, géré par les élèves et facile à mettre en œuvre). Notre proposition n'est pour l'instant que très modérément ludique. Cependant, nous envisageons d'améliorer cet aspect en introduisant un jeu simple basé sur cette distribution, comme la présence dans la main d'un joueur de certaines cartes, etc. Nous n'avons pas mis en évidence l'extensibilité, mais nous envisageons évidemment d'introduire des réflexions autour de la terminaison d'un algorithme. Pour le dernier critère, l'histoire, nous n'avons pas cherché particulièrement à le développer. L'âge des élèves et la simplicité de l'activité ne nécessitaient pas cette caractéristique.

Concernant notre ambition d'un support abordable permettant de centrer notre propos sur l'introduction du concept de correction partielle d'algorithme, il nous semble que l'activité d'informatique débranchée a bien rempli son rôle. Chacun s'est bien prêté au jeu et l'apport kinesthésique est indéniable (chacun a au moins dû, à un moment, compter ses cartes, voire jouer le rôle de donneur). Nous regrettons cependant que ces phases de manipulation soient un peu trop courtes et que les incessantes mises en commun soient indispensables à l'émergence des notions. Nous nous écartons d'une situation adidactique en ne donnant le problème et les consignes qu'au fur et à mesure. Nous faisons en réalité une succession de petites activités apportant chacune rapidement des réponses, par la simple prise de conscience des problèmes qui se posent. Le temps dévolu aux phases de manipulation, finalement assez court, a été suffisant pour permettre des discussions fructueuses et a permis de faire émerger les notions ciblées. Cette rapidité dans la résolution des problèmes envisagés vient en fait, paradoxalement, de la simplicité (voulue) de la situation. Peut-être choisir les supports les plus simples n'est-il pas toujours la meilleure approche, car elle ne permet pas aux élèves d'intégrer la pertinence de l'abstraction qu'on souhaite en déduire. Il est d'autant plus difficile de formaliser une notion qu'elle paraît évidente.

En ce qui concerne la facilitation de l'introduction des notions théoriques, l'objectif est atteint. Le vocabulaire a pu être introduit sans difficulté. Les notions, retravaillées aussitôt sur des exercices moins contextualisés, ont été correctement remobilisées par les élèves. Parler d'invariant lors de la présentation des algorithmes de tri a été plus facile. Par contre, les faire vivre sur le long terme est une gageure. Attendre trop longtemps pour n'en reparler que sur les algorithmes au programme assure à chacun d'avoir oublié le vocabulaire travaillé (mais permet de réactiver à peu de frais les notions). Il nous paraît donc important, dans la progression, de prévoir peu de temps entre cette activité et son utilisation. La difficulté majeure concerne la réutilisation et peut être la compréhension du mécanisme mathématique du raisonnement par récurrence. L'implication au sens mathématique reste une notion difficile pour certains élèves, elle est travaillée de la 6^e à la terminale au second degré. Pourtant, même les élèves de terminale spécialité maths éprouvent des difficultés avec celle-ci (et la logique en général) lors de l'introduction du raisonnement par récurrence.

Cette activité est devenue un incontournable dans les cours de première NSI des auteurs, qui s'en servent en jouant au maximum avec les variables didactiques. La distribution ne concerne alors pas le paquet de cartes entier, mais une partie du paquet, divisée (non aléatoirement) entre plusieurs tables. La subtilisation de cartes peut être faite dans le talon plutôt que dans le paquet d'un joueur, où une carte peut être ajoutée ou déplacée... Chaque décision peut donner lieu à un échange entre élèves ou avec l'enseignant et est susceptible de faire avancer les choses.

Le nombre de cartes distribuées à chaque joueur et le nombre restant de cartes sont le quotient et le reste dans la division euclidienne de la taille du paquet de départ par le nombre de joueurs. Il peut être intéressant de demander, à la fin de l'activité, après la phase d'institutionnalisation indispensable, d'ajouter ce fait comme post-condition et de proposer un invariant pour le démontrer. On peut également poursuivre le travail sur cette activité par l'introduction de la terminaison, en considérant le nombre de cartes encore à distribuer comme variant. La situation peut éventuellement être réutilisée dans des niveaux inférieurs, pour formuler un algorithme (sans le programmer), appréhender le problème de la terminaison ou même introduire la division euclidienne en cycle 3. Elle peut aussi être exploitée en mathématiques expertes, en apportant des preuves de correction d'algorithmes, tels que celui d'Euclide, la décomposition dans une base ou le test de primalité.

Références bibliographiques

- Alayrangues, S., Peltier, S. & Signac, L. (2017) Informatique débranchée : construire sa pensée informatique sans ordinateur. *Colloque Mathématiques en Cycle 3 IREM de Poitiers*, IREM de Poitiers, Poitiers, France. 216-226. <https://hal.science/hal-01868132v1>
- Bell, T., Witten, I. H. & Fellows, M. (1998) *Computer Science Unplugged: off-line activities and games for all ages*. <https://classic.csunplugged.org/documents/books/english/unplugged-book-v1.pdf>
- Bell, T. Alexander, J., Freeman, I. & Grimley, M. (2009). Computer science unplugged: school students doing real computing without computers. *Journal of Applied Computing and Information Technology*, 13(1):20-29.
- Brousseau, G. (1998) *Théorie des Situations Didactiques*, La pensée sauvage.
- Crozet G. & Gropserrin, R. (1987) *Informatique sans ordinateur*, Centre National de Documentation Pédagogique (CNDP) <https://files.inria.fr/mecsci/gropserrin-et-al/informatique-sans-ordinateur.pdf>
- Duflot-Kremer, M. (s. d.) *Les bases de données... à tricoter*, consulté le 11/02/2025 à l'adresse <https://members.loria.fr/MDuflot/files/med/BDatricoter.html>
- Drot-Delange, B. (2013). Enseigner l'informatique débranchée : Analyse didactique d'activités. *Actes du congrès de l'Actualité de la Recherche en Éducation et Formation (AREF - AECSE)*, 1-13. https://archivesic.ccsd.cnrs.fr/sic_00955208v1
- Grenier, D. (2012) Une étude didactique du concept de récurrence, *Petit x*, n°88, 27-47 <https://irem.univ-grenoble-alpes.fr/revues/petit-x/consultation/numero-88-petit-x/>
- Meyer, A. & Modeste, S. (2020) Analyse didactique d'un jeu de recherche : Vers une situation fondamentale pour la complexité d'algorithmes et de problèmes, *Colloque Didapro 8 – DidaSTIC L'informatique, objets d'enseignements – enjeux épistémologiques, didactiques et de formation*, Lille, France.

- Meyer, A. & Modeste, S. (2022) Situation didactique autour d'un jeu de recherche : expérimentation en classe de NSI, *Didapro* 9, Le Mans, France.
- Kayce, Mobley. & Sarah, Fisher. (2014). Ditching the Desks: Kinesthetic Learning in College Classrooms. *The Social Studies*, 105(6):301-309.
- Modeste, S., Gravier, S. & Ouvrier-Bufferet, C. (2010) Algorithmique et apprentissage de la preuve. *Repères IREM*, 79, 51-72.
- Nishida, T., Kanemune, S., Idosaka, Y., Namiki, M., Bell, T. & Kuno, Y. (2009) A CS unplugged design pattern. *Proceedings of the 40th ACM technical symposium on Computer science education (SIGCSE '09)*. Association for Computing Machinery, New York, NY, USA, 231–235. <https://doi.org/10.1145/1508865.1508951>
- Selby, C. & Woollard, J. (2014) Computational thinking: The developing definition. *Actes du SIGCSE*.
- Sellarès, J. A., & Toussaint, G. (2003). On the role of kinesthetic thinking in computational geometry. *International Journal of Mathematical Education in Science and Technology*, 34(2), 219–237. <https://doi.org/10.1080/0020739031000071511>
- Wing, J. M. (2006) Computational Thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717-3725.
- MEN (2019a) *Programme d'enseignement de spécialité de numérique et sciences informatiques de la classe de première de la voie générale*, Bulletin officiel spécial n°1 du 22 janvier 2019. https://cache.media.education.gouv.fr/file/SP1-MEN-22-1-2019/26/8/spe633_annexe_1063268.pdf
- MEN (2019b) *Programme de l'enseignement de spécialité de numérique et sciences informatiques de la classe terminale de la voie générale*, Bulletin officiel spécial n°8 du 25 juillet 2019 https://cache.media.education.gouv.fr/file/SPE8_MENJ_25_7_2019/93/3/spe247_annexe_1158933.pdf