
DÉCOUVRIR LE CONCEPT D'ITÉRATION AU COLLÈGE : DES ACTIVITÉS D'INFORMATIQUE DÉBRANCHÉE SUR LES DÉPLACEMENTS ET LES TRACES

Régine HAMM-AUDONNET¹

Collège de Brumath, IREM de Strasbourg

Sophie SCHWARTZ-COMMECY²

Collège Caroline Aigle, IREM de Strasbourg

Charlotte DEROUET³

LISEC UR 2310, IREM de Strasbourg, Université de Strasbourg

Basile SAUVAGE⁴

UFR de mathématique et d'informatique, IREM de Strasbourg, Université de Strasbourg

Résumé. Dans le cadre de l'enseignement des mathématiques au collège en France, il est attendu des élèves, à la fin de la cinquième, d'écrire un script de déplacement ou de construction géométrique utilisant la boucle < répéter... fois >. L'article s'intéresse à l'enseignement de l'informatique, et plus particulièrement à l'introduction du concept d'itération au collège. L'objectif de cet article est de présenter et d'analyser deux activités d'informatique débranchée sur les déplacements et les tracés, pour découvrir le concept d'itération, conçues autour d'un jeu et expérimentées dans des classes de sixième et de cinquième. Les activités s'appuient sur un jeu de cartes avec des instructions Scratch permettant de se déplacer sur un quadrillage. Dans la première activité, les élèves doivent avancer sur le quadrillage librement, à l'aide de cartes tirées au hasard. Dans la deuxième activité, le chemin est imposé, les élèves doivent proposer différentes suites d'instructions permettant de suivre ce chemin. Le défi est d'utiliser un minimum d'instructions. L'utilisation de boucles permet de relever ce défi.

Mots-clés. Itération, informatique débranchée, Scratch, boucle, jeu.

Abstract. As part of the teaching of mathematics in lower secondary schools in France, students are expected, by the end of grade 7 (*cinquième*), to be able to write scripts for movement or geometric constructions using the < repeat ... times > loop. This article focuses on the teaching of computer science, and more specifically on the introduction of the concept of iteration at the lower secondary school level. The goal is to present and analyze two activities in computer science unplugged centered on movement and path drawing, designed to make discover the concept of iteration. These activities, built around a game, were tested in grades 6 (*sixième*) and 7 (*cinquième*) classes. They use a card game with Scratch instructions to move on a grid. In the first activity, students move freely on the grid using randomly drawn instruction cards. In the second activity, the path is predetermined, and students must come up with various sequences of instructions allowing to follow it. The challenge is to use as few instructions as possible. Using loops is a means to meet this challenge.

Keywords. Iteration, computer science unplugged, Scratch, loop, game.

¹ regine.hamm@ac-strasbourg.fr

² sophie.schwartz2@ac-nancy-metz.fr

³ charlotte.derouet@unistra.fr

⁴ sauvage@unistra.fr

Introduction

Dans le cadre de l'enseignement des mathématiques au collège, nous avons constaté dans nos classes que les élèves rencontrent des difficultés dans l'écriture ou l'exécution d'un algorithme de déplacements et de tracés dans le langage du logiciel Scratch, notamment dans les exercices débranchés proposés dans l'épreuve de mathématiques du Diplôme National du Brevet (DNB). Les exercices d'informatique débranchée ne permettent pas aux élèves de mettre en œuvre la stratégie d'essai-erreur qu'ils utilisent souvent lors des activités de programmation avec le logiciel Scratch sur ordinateur. Ce type d'exercices exige de la part des élèves davantage de compréhension et de maîtrise des différentes instructions, en particulier les itérations, aussi appelées boucles.

Le groupe Informatique de l'IREM de Strasbourg est parti de ce constat pour concevoir des activités débranchées autour de plusieurs jeux qui permettent aux élèves de mieux appréhender, dès la fin du cycle 3 ou le début du cycle 4, les instructions de déplacements, de rotations et de boucles. Le but de ces activités est à terme de permettre aux élèves d'être capables d'utiliser à bon escient les différentes instructions et ainsi de compléter ou d'exécuter un algorithme, sans avoir besoin de les tester sur un ordinateur.

À ce stade de leur apprentissage, il s'agit seulement d'acquérir une maîtrise élémentaire des boucles. Dans une perspective plus large, c'est l'occasion de découvrir que la longueur d'un algorithme et la longueur de sa trace d'exécution sont deux choses différentes. Ainsi, ce peut être un prélude pour découvrir le concept de complexité des algorithmes, qui pourra être étudiée plus tard dans l'enseignement.

Cet article s'intéresse à l'introduction du concept d'itération avec des activités débranchées. Les objectifs de celui-ci sont de présenter et analyser des activités, autour d'un jeu particulier, conçues et expérimentées en classes de sixième et cinquième. Les activités sont analysées *a priori* puis *a posteriori* à partir des déroulements des expérimentations. Après avoir présenté les éléments théoriques et curriculaires sur lesquels nous appuyons nos propositions (partie 1), nous présenterons le jeu *Trace ton chemin* que nous avons développé et les deux activités associées, accompagnées d'une analyse *a priori* (partie 2). Dans une troisième partie, nous présenterons les retours d'expérimentations relatives à ces activités et en ferons une analyse *a posteriori*. Et enfin, nous conclurons.

1. Contexte institutionnel et éléments théoriques

1.1. Dans les ressources institutionnelles

Pour concevoir nos activités, nous nous sommes appuyés sur le programme de mathématiques du cycle 4 (MENJS, 2020), les attendus de fin d'année de cinquième (publiés sur Eduscol) et également les ressources d'accompagnement du programme de mathématiques (cycle 4) dans la catégorie *Algorithmique et Programmation* (MENESR, 2016).

Les objectifs des deux activités, que nous allons présenter dans cet article, s'inscrivent dans le thème E : *Algorithmique et programmation* du programme de mathématiques du cycle 4 : « Au cycle 4, les élèves s'initient à la programmation, en développant dans une démarche de projet quelques programmes simples, sans viser une connaissance experte et exhaustive d'un langage ou d'un logiciel particulier » (MENJS, 2020, p. 137).

Le document Ressource *Algorithmique et programmation* du programme de mathématiques du cycle 4 (MENESR, 2016) explicite les compétences que les élèves doivent développer en algorithmique et programmation. Il précise que les modalités d'apprentissage peuvent être

variées et, par exemple, peuvent prendre la forme d'activités en mode débranché.

Dans cet article, nous nous focalisons sur quatre des compétences présentées dans le document Ressource (MENESR, 2016), qui sont les suivantes :

- [la] *décomposition* : **analyser un problème compliqué, le découper en sous-problèmes, en sous tâches** ;
- [la] *reconnaissance de schémas* : **reconnaître des schémas, des configurations, des invariants, des répétitions**, mettre en évidence des interactions ;
- [la] *généralisation et [l'] abstraction* : repérer les enchaînements logiques et les traduire en instructions conditionnelles, **traduire les schémas récurrents en boucles**, concevoir des méthodes liées à des objets qui traduisent le comportement attendu ;
- [la] *conception d'algorithme* : **écrire des solutions modulaires à un problème donné**, réutiliser des algorithmes déjà programmés, programmer des instructions déclenchées par des événements, concevoir des algorithmes se déroulant en parallèle. (p. 2)

Les parties du texte en gras soulignent les éléments précis de cet extrait sur lesquels nous nous focalisons dans cet article.

1.2. L'itération dans des travaux et recherches antérieurs

Pour faire émerger le concept d'itération ou boucle, nous nous appuyons sur le besoin d'« économie de la communication homme – machine » (Nguyen et Bessot, 2010, p. 47). Les auteurs notent que « l'intention d'exécuter des calculs répétitifs avec un invariant opératoire est une condition pour concevoir et élaborer une communication à la machine qui économise la communication de tous les calculs effectués par elle. Le concept de boucle est une réponse à cette recherche d'économie » (ibid., p. 47). Pour cela, les activités que nous proposons mettent en évidence que plusieurs algorithmes engendrent un même déplacement, et valorisent les algorithmes les plus courts. Dans un contexte d'activité débranchée, puisqu'il n'y a pas de machine, l'« économie de la communication homme-machine » n'est pas à comprendre *stricto sensu* : l'élève éprouve l'effort à concevoir et écrire l'algorithme, et constate que la longueur de l'algorithme peut affecter la difficulté à le concevoir, la fastidiosité de son écriture et de sa vérification. Cette expérience se retrouve dans les activités avec programmation sur machine.

Notre objectif dans les activités proposées est de découvrir le concept de boucle et d'en comprendre le fonctionnement élémentaire. Ce travail se situe à l'entrée dans l'algorithmique et la programmation, et peut contribuer à l'« alphabétisation informatique » (Lagrange et Rogalski, 2017, p. 141). Aussi, nous proposons des activités qui évitent deux autres obstacles rapportés par Lagrange et Rogalski (ibid., p. 146), à savoir la notion de variable et la notion d'invariant de boucle, qui exigent de se représenter l'état de la mémoire comme une fonction temporelle de l'exécution du programme. Nous nous inscrivons ainsi dans la stratégie d'enseignement avec « absence de machine de référence », évitant ainsi « la connaissance préalable d'un langage de programmation » (Nguyen et Bessot, 2010, p. 34). Bien qu'elles fassent appel au langage Scratch, nos activités relèvent de l'informatique débranchée¹ : les élèves n'utilisent pas d'ordinateur, mais manipulent des objets tangibles. Il convient alors de faire attention à deux écueils, soulignés par Drot-Delange (2013), si l'on veut dépasser la simple sensibilisation et intégrer ces activités dans un curriculum. Le premier écueil serait de faire exclusivement des activités débranchées. En effet, les activités avec et sans ordinateur sont complémentaires. De ce point de vue, l'utilisation d'instructions en langage Scratch peut ici s'avérer être un atout. Le second écueil serait de proposer des activités isolées, sans articulation explicite entre elles, au risque de ne pas leur donner de sens. Nous prendrons donc soin d'institutionnaliser le concept de

¹Voir article de synthèse sur l'informatique débranchée de Drot-Delange (2013).

boucle à l'issue des activités proposées.

Des recherches en didactique de l'informatique en France ont déjà étudié la reconnaissance de motifs et la répétition (notamment Peter et al., 2020 ; Léonard et al., 2020 ; Léonard et al., 2022). Les activités proposées sont dans le cadre d'un déplacement d'un personnage ou d'un curseur. L'étude de Léonard et al. (2020) a montré que les élèves de 6-7 ans sont capables de détecter des motifs et de les exprimer sous forme de répétition, dans le cas où le code utilisé et les éléments du motif se ressemblent. Dans ces travaux, ils utilisaient notamment le langage Scratch. Les chercheurs ont ensuite essayé dans d'autres travaux de catégoriser certaines difficultés des élèves relatives à l'identification de motifs (Léonard et al., 2022).

1.3. Ressources pour l'enseignement du concept de boucle dans le secondaire

Peu de ressources pédagogiques existent pour l'enseignement et l'apprentissage du concept de boucle en informatique au collège. Par exemple, le site Eduscol² n'en propose pas.

Nous pouvons tout de même citer le jeu sérieux Pyrates qui vise l'introduction du langage Python en seconde, dans lequel l'utilisateur contrôle un personnage à l'aide de programmes Python (Branthôme, 2021). Au fur et à mesure des niveaux, différents concepts informatiques sont introduits, notamment la boucle. Cependant le langage Python n'est pas au programme du collège, de plus nous cherchons ici à aborder le concept en mode débranché.

En informatique débranchée, selon Baron & Drot-Delange (2016), « le principe des activités [...] est de reposer principalement sur le jeu, la manipulation d'objets, des tours de magie. Ce sont les principes de la science en informatique qui sont centraux dans la démarche proposée, et pas seulement la programmation. Les activités doivent être simples, susciter l'engagement de l'enfant et favoriser les situations de coopération ou de compétition » (p. 58).

L'IREM de Clermont-Ferrand³ propose une activité d'informatique débranchée, l'activité du Gobot, dans laquelle le besoin d'économie de la communication émerge d'une difficulté à écrire et exécuter à la main, sans erreur, plusieurs dizaines d'instructions. Les élèves sont alors amenés à concevoir une instruction de répétition. La reconnaissance de motifs est ici peu présente, il s'agit de répéter une seule instruction.

Jutand et Kirch (2020) de l'IREM de Poitiers proposent une activité où les élèves décrivent un déplacement avec des cartes représentant des instructions. Pour introduire la structure de boucle, ils s'appuient aussi sur l'aspect fastidieux de la tâche manuelle, mais aussi sur le repérage de motifs qui se répètent. Nos propres activités peuvent se rapprocher de cette proposition.

2. Présentation et analyse *a priori* de deux activités autour d'un jeu

Dans cette partie, nous allons présenter les deux activités d'algorithmique en mode débranché prenant appui sur un jeu de cartes conçu par le groupe IREM pour répondre à la problématique de l'introduction du concept d'itération. Nous précisons les objectifs de celles-ci et justifierons nos choix.

Le jeu *Trace ton chemin* a été conçu pour des élèves de classes de sixième et de cinquième car ce sont les classes où le concept d'itération est censé être introduit. Il n'est pas nécessaire que les élèves aient utilisé le logiciel Scratch avant la mise en œuvre de ces activités. Le seul prérequis indispensable pour le bon fonctionnement du jeu et des activités associées est la compréhension des actions réalisées par les instructions < avancer de 1 pas >, < répéter ... fois >, < tourner vers

² <https://eduscol.education.fr/280/mathematiques-cycle-4>, consulté le 29/01/2025.

³ <http://www.irem.univ-bpclermont.fr/>

la gauche de 90 degrés >. Une première manipulation de ces instructions lors de questions flash par exemple devrait suffire comme préliminaire.

2.1. Présentation du jeu *Trace ton chemin*

Il s'agit d'un jeu avec des cartes et un plateau. Les cartes « instruction » (exemples en figure 1) représentent les instructions Scratch suivantes :

- avancer de 1 pas (18 exemplaires)
- tourner à droite (dans le sens horaire) de 90 degrés (12 exemplaires)
- tourner à gauche (dans le sens anti-horaire) de 90 degrés (12 exemplaires)
- répéter 2 fois (8 exemplaires)
- répéter 3 fois (4 exemplaires)
- répéter 4 fois (2 exemplaires)



Figure 1. Exemples de cartes « instruction ».

Nous avons fait le choix de cartes sous forme d'instructions Scratch pour familiariser les élèves avec ces instructions qui font partie des choix de langage informatique pour les cycles 3 et 4. Cela permettra aux élèves une reconnaissance et un gain de temps d'appropriation de ces instructions lors des premières séances en salle informatique. Nous avons bien conscience que pour l'instruction < répéter ... fois >, il n'est pas possible avec les cartes d'insérer les instructions à répéter « à l'intérieur » du bloc comme sur le logiciel Scratch, cependant nous faisons l'hypothèse que les élèves trouveront un moyen de matérialiser en posant leurs cartes d'une certaine façon ou de verbaliser à quelles instructions se rapportent la répétition.

Le plateau est un quadrillage de 19 lignes et 24 colonnes avec une colonne bleue à une extrémité et une colonne rouge à l'autre extrémité (figure 2). Les déplacements dans le jeu s'effectuent sur les lignes du quadrillage et s'arrêtent sur un sommet.

Le jeu se joue à deux joueurs. Le joueur bleu part de la zone bleue et le joueur rouge part de la zone rouge. Au départ, chaque joueur prend, à tour de rôle, les quatre cartes face cachée du dessus de la pioche (les cartes étant mélangées au préalable). Il choisit son point de départ, un des sommets du quadrillage se situant sur le bord de sa zone, ainsi que son orientation de départ. Lorsque c'est son tour, le joueur peut poser jusqu'à quatre cartes < instruction > en sachant qu'il n'a pas le droit de couper le chemin de son adversaire. Il trace sur le plateau le déplacement correspondant et indique son orientation finale à l'aide d'une petite flèche. À la fin de son tour, il pioche le nombre de cartes nécessaires pour avoir à nouveau quatre cartes en main. L'objectif du jeu est d'être le premier à rejoindre la zone de l'adversaire. Dans la figure 2, nous présentons un exemple de début de partie, où le joueur bleu commence.

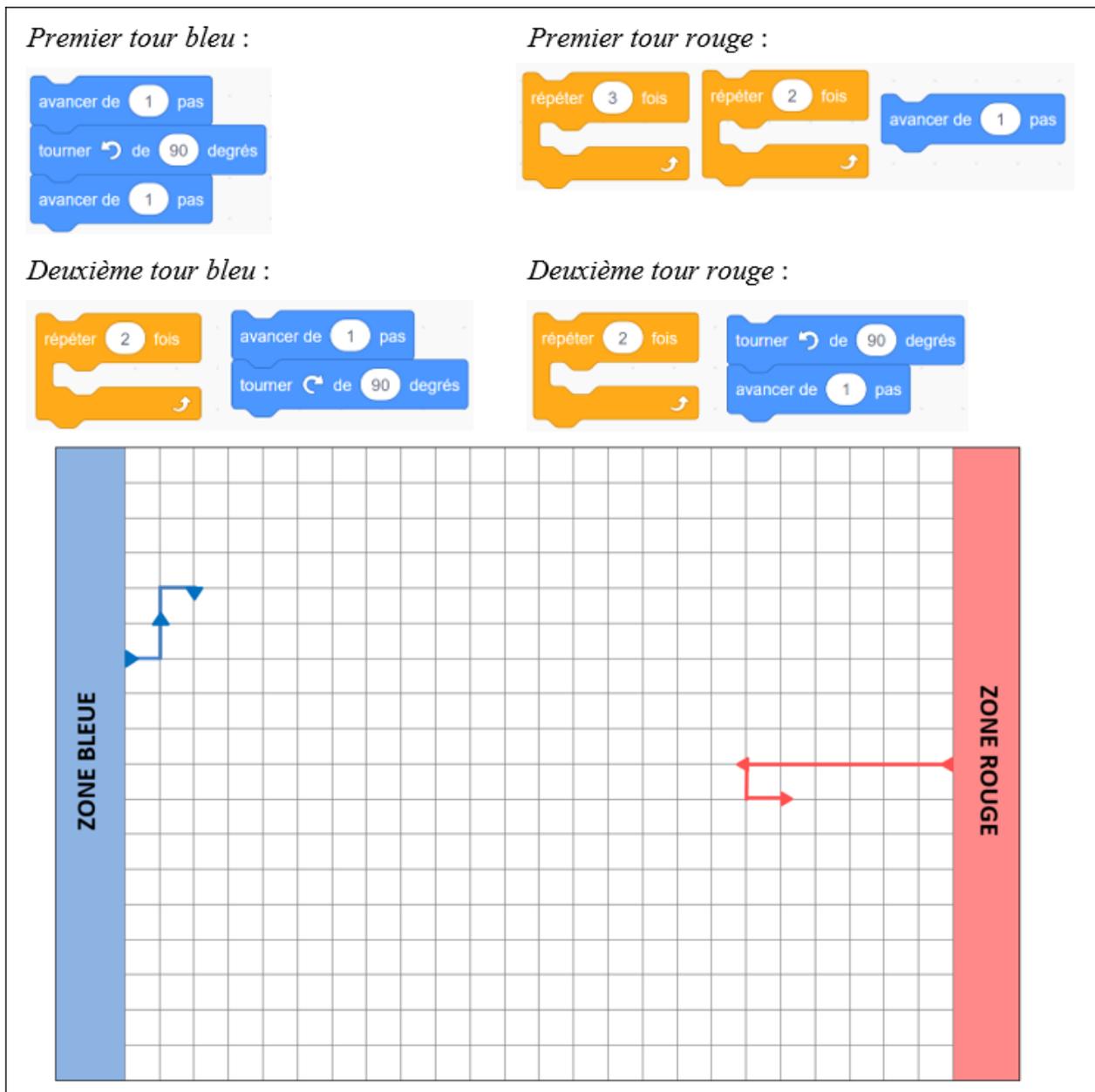


Figure 2. Exemple de début de partie avec deux tours par joueur.

Nous avons fait le choix (pour la première activité) de proposer un plateau de taille suffisamment grande (19 lignes et 24 colonnes) pour permettre aux joueurs de pouvoir avancer sans être véritablement contraints par le joueur adverse, tout en gardant l'aspect motivationnel du jeu en duel. Cela permet aux joueurs de se concentrer sur leur trajectoire et leur objectif de rejoindre la zone opposée le plus rapidement possible. Certains élèves peuvent essayer de bloquer leur adversaire et cela permet une différenciation du jeu selon le niveau de maîtrise des instructions de déplacement et de la capacité de l'élève à établir une stratégie.

Avec nos choix de cartes, il serait envisageable qu'un joueur gagne dès le premier tour en imbriquant trois boucles (figure 3), cependant il nous semble qu'il est peu probable que cela arrive (dans le sens où il faudrait avoir ce tirage, mais surtout que l'élève soit en mesure de proposer cette combinaison pour avancer). Cette taille de grille permet donc de trouver un bon équilibre dans la durée d'une partie (de l'ordre de 15 minutes). Si cela venait tout de même à arriver, il suffirait de redémarrer une nouvelle partie. En revanche, c'est pour cette raison de

durée de jeu que nous n'avons pas mis de cartes < répéter n fois > avec $n > 4$.



Figure 3. Exemple de combinaison permettant de gagner du premier coup

2.2. Activité 1 : découverte du jeu

La première activité en classe consiste à découvrir le jeu et, à cette occasion, les différentes instructions disponibles et la combinaison d'instructions en acte. Cette phase peut avoir lieu sur une séance d'une heure, en classe entière ou en demi-classe, en mathématiques ou en accompagnement personnalisé par exemple. Les règles du jeu sont présentées et distribuées aux élèves (annexe 1), l'exemple de début de partie en figure 2 est aussi proposé. En revanche, il n'est pas véritablement expliqué aux élèves comment utiliser les cartes < instruction >, notamment la carte < répéter ... fois >. Il est attendu que les élèves se questionnent à ce sujet pendant le déroulement du jeu. Cependant le langage Scratch nous semble assez transparent et accessible aux élèves pour que les discussions riches entre pairs émergent

Dans cette phase, les élèves sont répartis en petits groupes de trois élèves constitués d'un joueur bleu, d'un joueur rouge et d'un observateur. Les élèves changent de rôle à chaque nouvelle partie. Le rôle de l'élève observateur est de vérifier la correspondance du tracé avec les cartes < instruction > posées et de signaler aux deux joueurs leurs éventuelles erreurs. L'expérimentation a été organisée sans intervention, validation ou correction systématiques des propositions par l'enseignant, l'intérêt étant aussi de susciter la discussion en cas de désaccord sur le déplacement correspondant aux cartes posées et obliger les trois élèves à argumenter. Il peut cependant être envisagé de laisser la possibilité aux élèves de solliciter l'enseignant pour un arbitrage si l'accord ne se fait lors de la discussion entre l'observateur et les deux joueurs.

Le plateau de jeu est inséré dans une pochette transparente et les élèves tracent leur trajectoire avec des feutres d'ardoise (figure 4), ce qui permet à la fois la réutilisation de la grille et la correction des erreurs éventuelles.

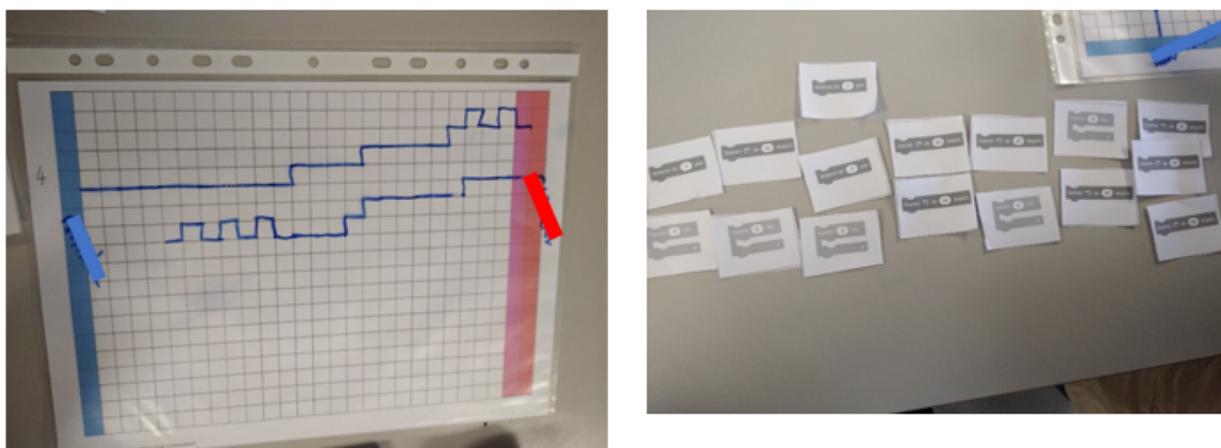


Figure 4. Exemple de partie de l'activité 1.

Dans l'activité, il est attendu que les élèves soient capables de :

- comprendre les instructions et les exécuter à la main ;

- manipuler des boucles ;
- assembler quelques instructions pour réaliser un déplacement ;
- exécuter à la main un algorithme contenant des boucles.

Ces objectifs répondent aux objectifs de contenus algorithmiques et de compétences du programme de mathématiques du cycle 4 (MENJS, 2020). Pour cette première activité, nous nous sommes concentrés en particulier sur les connaissances « notions d’algorithme et de programme », et « séquences d’instructions, boucles [...] » du programme. Le travail sur la compétence associée « écrire [...] un programme en réponse à un problème donné » sera amorcé dans cette activité et sera approfondi dans l’activité 2.

Nous nous attendons, dans cette activité, à quelques écueils ou difficultés des élèves. Comme nous l’avons précisé plus haut, l’impossibilité matérielle d’insérer les instructions de la boucle à l’intérieur du corps de la boucle peut constituer une difficulté liée au matériel. Cependant cela ne nous semble pas véritablement problématique, car les élèves pourront verbaliser les instructions concernées par la répétition ou bien trouver une façon de matérialiser avec le positionnement des cartes cette répétition. Il est possible aussi que certains élèves ne s’autorisent pas (consciemment ou non) à mettre plusieurs instructions à répéter dans une boucle. L’imbrication de boucles, possible et même attendue, sera aussi certainement source de difficulté pour certains élèves. Notamment sur la compréhension de l’imbrication de l’instruction < répéter 2 fois > < répéter 3 fois > comme correspondant à répéter 6 fois les instructions à répéter. Cette interprétation qui est l’interprétation standard en informatique nous semble cependant accessible aux élèves, en appui sur leurs connaissances liées à la multiplication (école primaire) et le fait qu’il soit écrit « 2 fois » (qui relève du champ multiplicatif et non additif) sur les cartes. Nous considérerons comme « erronées » une interprétation différente, notamment du type répéter 5 fois. Les discussions dans le groupe d’élèves permettront au moins d’ouvrir le débat, si ce n’est de le clôturer.

Nous ne pouvons donc pas garantir que les élèves, à l’issue de cette séance, n’ont pas encore certaines conceptions erronées, notamment concernant le concept de boucle. C’est pour cette raison qu’il est important de poursuivre le travail initié avec l’activité 2, qui vise notamment à l’institutionnalisation des notions travaillées.

2.3. Activité 2 : plusieurs suites d’instructions pour un même tracé

Après avoir découvert le jeu dans l’activité 1, manipulé les instructions Scratch pour effectuer un déplacement, l’objectif de la deuxième activité est de faire prendre conscience aux élèves de la variété des possibles pour obtenir un déplacement, d’écrire différents algorithmes pour chaque déplacement, et ensuite d’identifier l’intérêt des boucles pour raccourcir un algorithme et donc d’amener les élèves à écrire une suite d’instructions pour obtenir le déplacement souhaité en utilisant la boucle < répéter ... fois >.

Ces objectifs ont été pensés pour correspondre aux attendus de fin d’année de 5ème (MENJS, 2020) :

Niveau 1 :

- *[L’élève] réalise des activités d’algorithmique débranchée.*
- *Il met en ordre et/ou complète des blocs fournis par le professeur pour construire un programme simple sur un logiciel de programmation.*
- *Il écrit un script de déplacement ou de construction géométrique utilisant des instructions conditionnelles et/ou la boucle « Répéter ... fois ». (p. 12)*

Le document stipule que « le Niveau 1 en Algorithmique et programmation est attendu en fin de

cinquième et qu'il est possible que certains élèves aillent au-delà » (ibid., p. 12). L'activité 2 est proposée lors d'une deuxième séance d'une heure en classe entière ou en demi-classe, en mathématiques ou en accompagnement personnalisé. Les élèves reçoivent, une fiche contenant deux plateaux de jeu de taille réduite (8 lignes et 8 colonnes) contenant un déplacement déjà tracé ainsi qu'un rappel visuel des différentes instructions disponibles dans le jeu *Trace ton chemin* (figure 5). Les cartes disponibles sont exactement les mêmes qu'à l'activité 1, notamment les élèves ont à disposition les instructions < répéter n fois > avec $n = 2, 3, 4$, ce qui leur permettra de les combiner. En revanche, dans cette activité, les élèves ne sont plus contraints en termes de quantité de cartes de chaque sorte à disposition. C'est une des grandes différences avec l'activité 1. Ici, l'objectif est en effet de trouver un maximum d'algorithmes possibles (question 1) et non plus un objectif de « rapidité » comme dans l'activité 1.

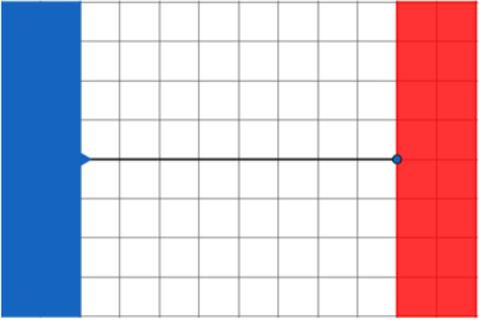
Tu trouveras ci-dessous deux chemins du bleu au rouge. Pour chacun d'eux :

1. Écris un maximum de suites d'instructions utilisant les cartes ci-dessous permettant d'arriver à ce résultat.
2. Essaie de trouver une suite d'instructions qui utilise le moins de cartes possible.

Rappel des cartes du jeu :



Chemin 1 :



Chemin 2 :

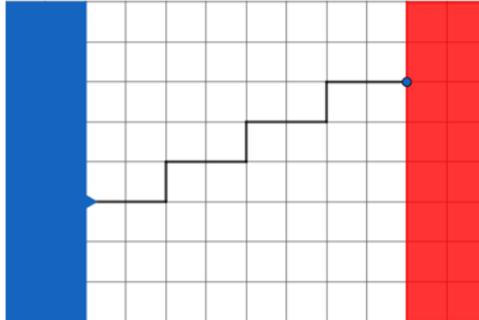


Figure 5. Énoncé de l'activité 2.

Dans la première question, il est demandé, pour chaque chemin, de proposer un maximum de « suites d'instructions » différentes (en gardant le contexte du jeu *Trace ton chemin*), ce qui correspond aux enchaînements de cartes < instruction >. Dans la deuxième question, il est demandé de trouver une suite d'instructions qui utilise le moins de cartes possible. Plusieurs modalités de choix de codage à utiliser pour retranscrire les enchaînements d'instructions ont été testées. Nous y reviendrons dans l'analyse *a posteriori*, cependant nous conseillons de laisser les élèves recopier les instructions des cartes à la main.

Le premier chemin proposé est une ligne droite, ce qui offre un grand nombre de possibilités d'enchaînements d'instructions (une quinzaine à l'ordre des instructions près, voir quelques exemples en figure 6) et a pour objectif de permettre à tous les élèves de rentrer dans l'activité et de trouver plusieurs solutions différentes. En effet, ils sont tous capables de trouver plusieurs enchaînements d'instructions qui permettent d'obtenir une ligne droite. La mobilisation de l'instruction de boucle semble accessible aux élèves, du fait que cette activité fait suite à l'activité 1. De plus, dans ce déplacement, les instructions liées aux rotations ne sont pas nécessaires, bien que possibles. Par ailleurs, les élèves qui, après la séance de découverte du jeu, sont déjà à l'aise avec les boucles et même les boucles imbriquées peuvent les utiliser de multiples façons.

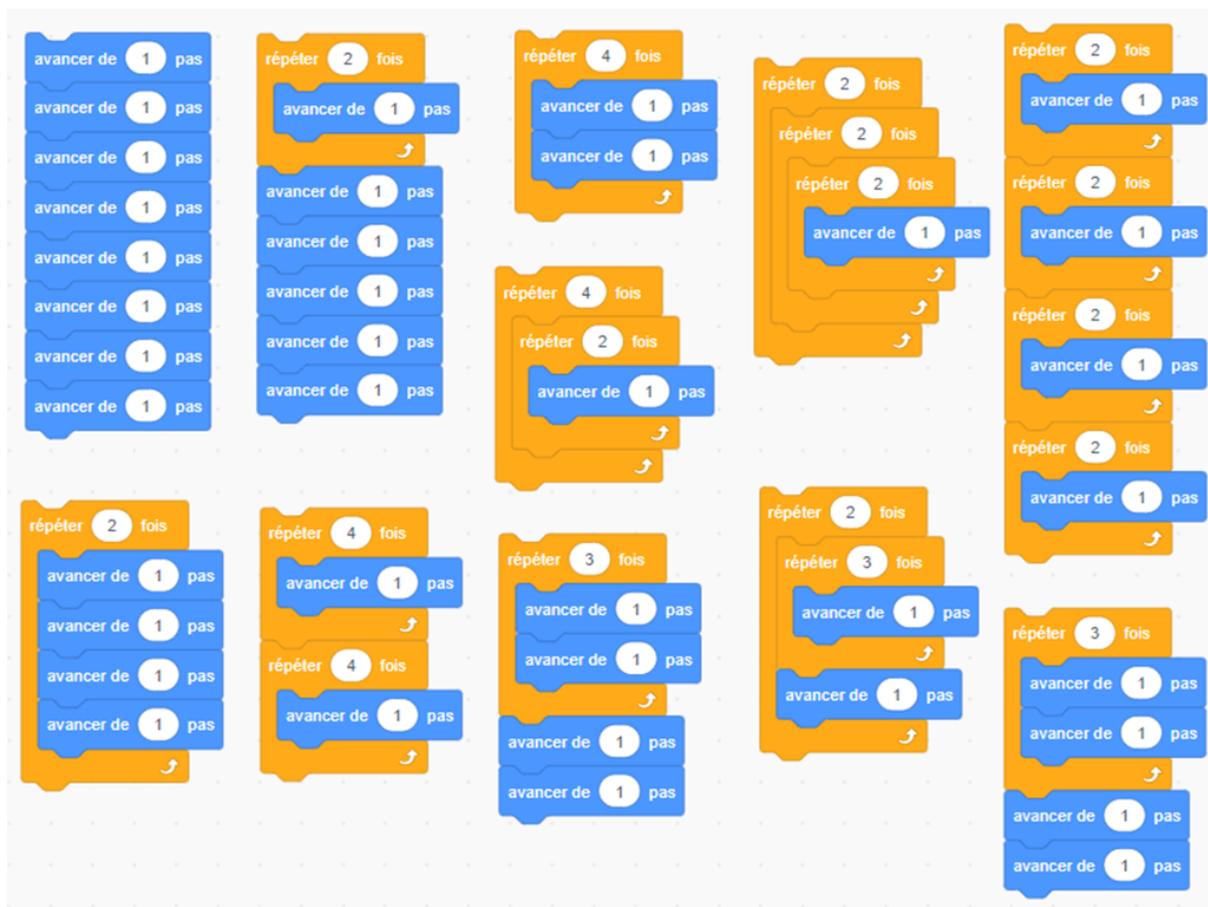


Figure 6. Exemples d'enchaînements d'instructions attendus pour le chemin 1.

Dans la figure 6, les exemples ont été représentés comme cela apparaîtrait sur le logiciel Scratch, ce qui permet de mettre « à l'intérieur » de la boucle les instructions répétées. Pour les élèves, ce sera à eux de le coder (ils n'ont plus les cartes matérielles), et plusieurs possibilités sont envisageables. Il s'agira de se mettre d'accord en classe (avant ou pendant la mise en commun). En tout cas, boucles successives et boucles imbriquées sont possibles.

Le deuxième chemin permet d'écrire des enchaînements d'instructions plus complexes que pour le premier déplacement et de manipuler les instructions < tourner à ... de 90 degrés > dans les deux sens. La répétition d'un motif bien visible et le grand nombre d'instructions nécessaires à la réalisation du déplacement a pour but d'inciter les élèves à décomposer le déplacement en sous-parties. Cette décomposition leur permettra d'introduire des boucles et par conséquent de réduire le nombre d'instructions. Les élèves devront malgré tout rester attentifs au fait qu'il y a cinq paliers horizontaux mais seulement quatre segments verticaux, ce qui complexifie

l'enchaînement d'instructions : en effet, le motif n'est pas répété un nombre entier de fois. Ce chemin a été choisi afin de permettre aux élèves, qui ne l'auraient éventuellement pas découvert lors de l'activité 1, de comprendre qu'il est possible et parfois nécessaire de mettre plusieurs instructions dans la boucle. Dans le chemin 2, en effet, trouver une suite d'instructions la plus courte nécessite d'utiliser l'imbrication de 5 instructions à répéter dans la boucle.

Lors de la mise en commun de toutes les solutions trouvées par les élèves, qui permet de rendre visible la multiplicité des enchaînements d'instructions possibles pour un déplacement donné, il est attendu que les élèves prennent conscience de deux éléments :

1. Plusieurs enchaînements d'instructions permettent d'obtenir le même déplacement ;
2. L'utilisation des instructions < répéter ... fois > peut permettre de réduire le nombre d'instructions, et par conséquent d'écrire un algorithme plus court.

Le vocabulaire spécifique de l'informatique est alors introduit : algorithme, programme, boucle. Ces notions sont construites à partir d'une décontextualisation des enchaînements d'instructions Scratch et de l'instruction < répéter ... fois >, découverts dans les activités. La trace écrite finale s'appuie sur les propositions des élèves avec leurs propres mots et le nouveau vocabulaire, après discussion à l'oral en classe.

3. Analyse *a posteriori* des expérimentations

Ces deux activités ont été expérimentées dans deux collèges publics de l'académie de Strasbourg, au mois de juin 2022. Après avoir présenté le contexte et les données récoltées, nous analyserons les expérimentations faites.

3.1. Contexte et récolte des données

L'activité 1 a été expérimentée auprès de trois demi-classes de sixième d'environ quinze élèves, en accompagnement personnalisé, et de deux classes de cinquième de 27 et 31 élèves. L'activité 2 a été proposée seulement à l'une des demi-classes de sixième (en accompagnement personnalisé) et aux deux classes de cinquième. Les élèves de ces différentes classes n'avaient jamais fait d'informatique dans leur scolarité. Chaque activité a été expérimentée sur une séance d'une heure. Les enseignantes ayant expérimenté dans leurs classes sont les deux premières autrices de l'article.

Pour un des groupes d'élèves de sixième, en plus de l'enseignante, des observateurs (une des autrices et des étudiants en stage) étaient présents (trois lors de l'activité 1 et deux lors de l'activité 2). Chaque observateur a assisté au travail d'un petit groupe de 3 élèves, sans intervenir. Ils ont pris des notes sur le déroulement des parties et les remarques et/ou discussions entre les élèves, ainsi que des photos des parties. Des traces écrites des déroulés de parties (activité 1) ou des réponses des élèves (activité 2) ont été collectées dans chaque classe.

Nous allons maintenant présenter les analyses des déroulements de l'activité 1 puis de l'activité 2.

3.2. Analyse des déroulements de l'activité 1

L'objectif de la première activité est notamment de découvrir l'instruction de boucle à travers le jeu. A travers nos analyses, nous essayons d'identifier et de comprendre les stratégies, les réussites et les erreurs des élèves, en lien avec le concept de boucle. Plusieurs constats sur la prise en main et la mise en œuvre du jeu ont été faits.

Le premier constat se rapporte à l'utilité des boucles. Les élèves ont rapidement compris que la carte < répéter ... fois > permet de se déplacer plus vite avec moins d'instructions. Le nombre

d'instructions étant limité (quatre au maximum), les élèves ont donc assez vite utilisé des boucles dans leur enchaînement. Ils ont bien identifié que la carte < répéter ... fois > ne pouvait être utilisée seule et que d'autres cartes s'y rapportaient. Certains posaient leur carte < répéter ... fois > et à sa droite les cartes à répéter (figure 7, à gauche) ou bien au-dessous (figure 7, à droite). En tout cas, bien qu'il ne soit pas possible avec les cartes de mettre les instructions à répéter « à l'intérieur » de la boucle, les élèves ont trouvé des façons de le faire comprendre.



Figure 7. Les boucles avec les cartes < instruction >.

Le deuxième constat se rapporte à la taille de la grille. Rappelons que nous avons choisi une grille suffisamment grande pour que les élèves puissent avancer sans être véritablement gênés par le chemin du joueur adverse (tout en gardant l'aspect compétitif). Nous avons observé que, dans deux des trois groupes de sixième, les élèves ont privilégié une traversée du plateau la plus rapide possible. Dans la troisième demi-classe en revanche, plusieurs groupes ont spontanément choisi d'essayer de bloquer leur adversaire (figure 8). La motivation à utiliser des boucles était plus grande lorsque l'objectif des élèves était la rapidité. Pour cela, le choix d'une grande grille semble avoir été un choix pertinent même si certaines parties ont duré plus de quinze minutes lorsque les joueurs tiraient peu de cartes < répéter ... fois > ou peu de cartes < avancer de 1 pas >. Il pourrait être judicieux de rééquilibrer les proportions de chaque type de cartes, afin d'accélérer les parties.

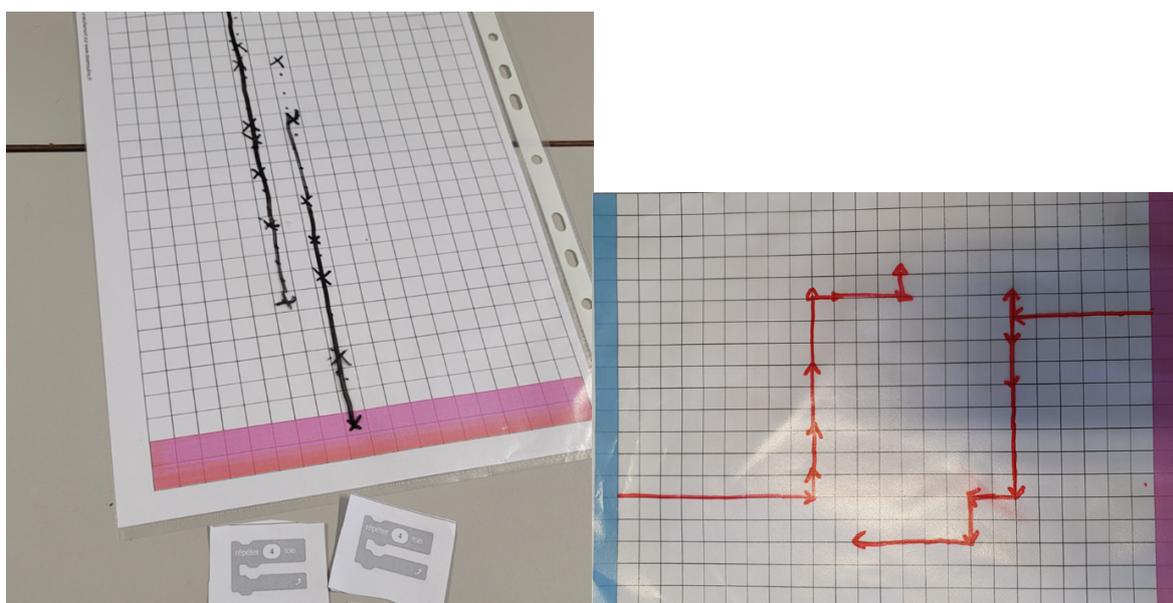


Figure 8. Exemples de parties où les élèves ont privilégié la stratégie de rapidité (gauche) ou la stratégie de blocage (droite).

Le troisième constat concerne l'imbrication de boucles. De nombreux groupes ont essayé d'imbriquer les boucles, mais les élèves n'étaient pas tous d'accord sur l'interprétation, comme nous l'avions prévu dans l'analyse *a priori*. Une boucle < répéter 3 fois > imbriquée dans une boucle < répéter 2 fois > correspond-elle à $3 + 2 = 5$ répétitions ou $3 \times 2 = 6$ répétitions ? Cela a engendré des débats dans la plupart des groupes, et certains se sont accordés sur l'interprétation erronée (5 répétitions). Ces débats sont manifestement féconds pour comprendre le concept de boucle. Ce concept a été bien compris par certains groupes au cours de l'activité 1 et bien identifié comme une stratégie efficace pour gagner en vitesse mais il semble essentiel de revenir sur cette question en grand groupe et de l'institutionnaliser. La durée des jeux n'a pas permis de le faire dans l'activité 1, mais cela a été fait lors de l'activité 2. Cela semble être pertinent de n'y revenir qu'en fin d'activité 2 car tous les élèves n'en sont pas au même stade avec les boucles en fin d'activité 1, donc une institutionnalisation sur l'imbrication de boucles à la fin de la première séance serait vraisemblablement prématurée pour certains élèves.

Le quatrième constat concerne l'utilisation de plusieurs instructions dans une boucle. De nombreux groupes ont utilisé des combinaisons comme < répéter 3 fois < avancer de 1 pas > < avancer de 1 pas > > pour gagner en rapidité. Il n'est cependant pas certain que tous les groupes l'aient fait lors de l'activité 1. Le tracé en escalier (figure 5) lors de l'activité 2 permet de s'assurer que tous les élèves l'ont compris.

Le cinquième constat se rapporte à l'apprentissage de l'orientation, du sens de rotation et de l'utilisation des boucles pour obtenir une orientation souhaitée. Par exemple, les élèves ont utilisé la boucle < répéter 3 fois < tourner vers la droite de 90° > > pour obtenir la même orientation qu'une carte < tourner vers la gauche de 90° > ce qui leur a permis de s'orienter correctement quand ils n'avaient pas le bon tirage, et révèle une autre motivation à utiliser la boucle.

Le sixième constat est l'utilisation d'une boucle pour simplement se défaire de cartes. L'objectif est d'aller le plus rapidement possible de l'autre côté du plateau, mais avec les cartes du tirage, il est intéressant aussi de se débarrasser de certaines de ses cartes : par exemple se débarrasser d'un maximum d'instructions < tourner vers la droite de 90° > ou < tourner vers la gauche de 90° > qui par moment les empêchaient d'avancer dans la direction de l'autre bord de la grille. Les élèves ont, par exemple, utilisé les instructions < répéter quatre fois < tourner vers la gauche de 90° > > pour garder la même orientation en se défaussant de la carte < tourner vers la gauche de 90° >. Pour certains groupes, l'utilisation d'un objet orienté (par exemple un crayon) sur la grille a été nécessaire pour bien choisir le nombre de répétitions de < tourner vers la gauche de 90° > pour obtenir l'orientation souhaitée.

Pour conclure, bien que des conceptions erronées (notamment sur les imbrications de boucles) puissent être utilisées à tort dans certains cas lors de l'activité 1, dans l'ensemble, la plupart des erreurs sont repérées et corrigées par la discussion dans le groupe. En cas de doute, l'accord doit être trouvé, ceci oblige les élèves à argumenter et à bien analyser l'enchaînement des instructions sur la grille pour convaincre les autres élèves du groupe. La présence de l'élève observateur est donc essentielle, car il est le garant du bon déroulement du jeu et, pour cela, il doit comprendre et valider les déplacements de ses camarades. Il est lui-même neutre et donc détaché de l'objectif à atteindre. Son rôle n'est pas de savoir si la stratégie choisie est la meilleure mais s'il y a cohérence entre les instructions posées et le déplacement sur la grille.

3.3 Analyse de l'activité 2

L'objectif de la deuxième activité est d'amener les élèves, par un travail individuel puis collectif, à réfléchir à la variété possible dans le choix d'instructions puis à l'optimisation de ce choix pour obtenir un algorithme plus court, pour ensuite en déduire le rôle et l'intérêt des boucles. Nous analysons cette activité sous trois angles : les réponses des élèves, en essayant d'identifier et comprendre les stratégies, les réussites et les erreurs des élèves ; le choix des conventions de codage, en présentant différentes propositions testées ; et l'institutionnalisation.

3.3.1. Réponses des élèves

Pour le chemin 1 et la recherche d'un maximum de suites d'instructions correspondantes à ce tracé, les élèves de sixième ont spontanément cherché à trouver la suite la plus courte, celle qui permettrait d'être le « plus rapide » dans le contexte de l'activité 1. Ils ont en effet essayé d'économiser au maximum le nombre d'instructions à retranscrire sur leur feuille. La recherche de plusieurs suites d'instructions, c'est-à-dire aussi d'algorithmes plus longs donc moins « performants » au regard de l'esprit du jeu était plus difficile pour eux.

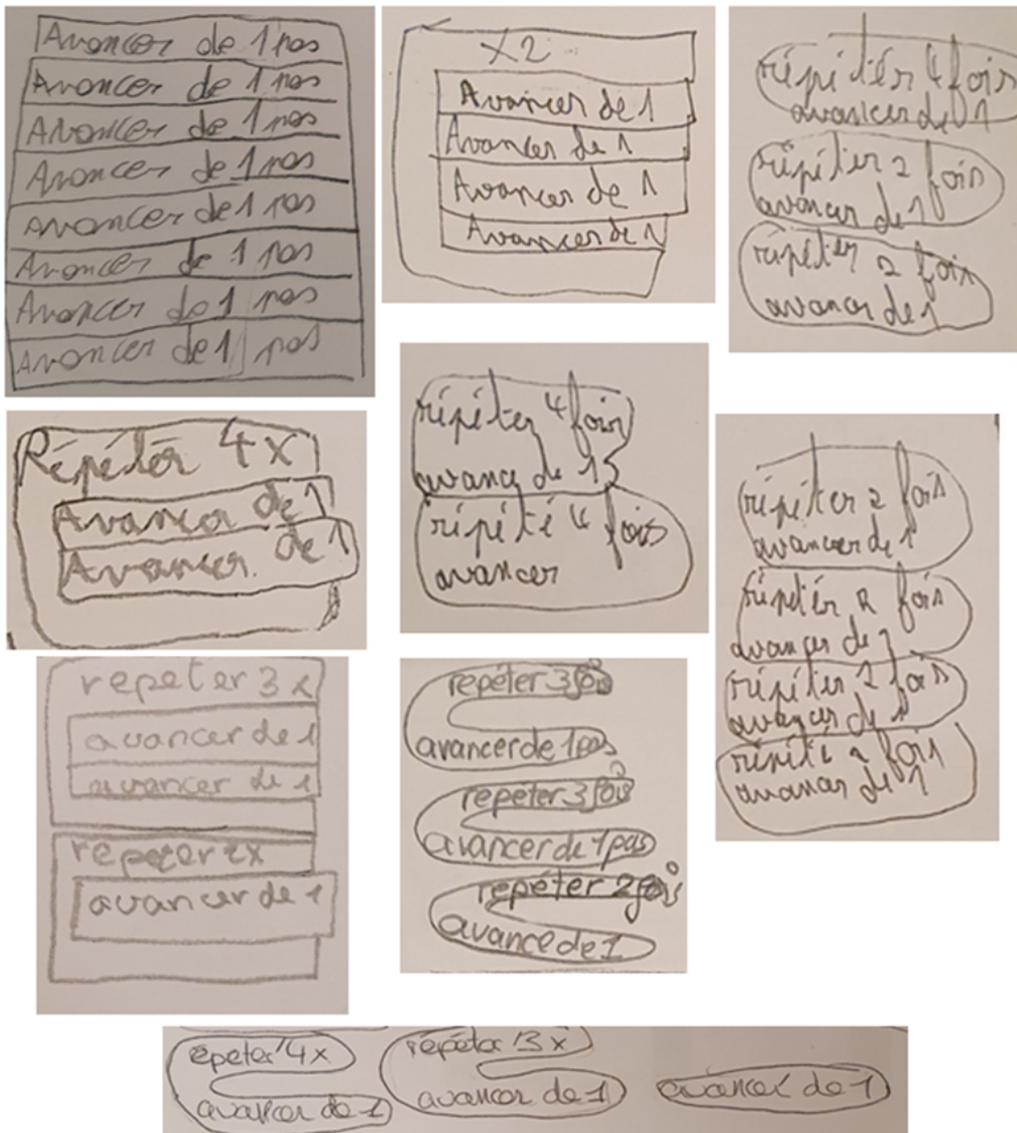


Figure 9. Exemples d'enchaînements d'instructions trouvés pour le chemin 1.

En leur présentant la recherche de solutions différentes comme un défi à relever, les élèves de cinquième se sont pris au jeu de trouver un maximum de solutions et ont proposé des enchaînements d'instructions variés (figure 9), ce qui a permis de comparer les longueurs des différents enchaînements et ainsi mettre en évidence l'intérêt des instructions < répéter ... fois > pour raccourcir l'enchaînement. Dans leur recherche du plus grand nombre de solutions différentes, certains élèves ont ajouté des instructions inutiles comme par exemple < tourner vers la gauche de 90° > (figure 10) alors que le trajet est rectiligne. C'est d'ailleurs en associant l'instruction < tourner vers la gauche de 90° > et l'instruction < tourner vers la droite de 90° > qu'ils s'étaient débarrassés de ces cartes lors de l'activité 1.

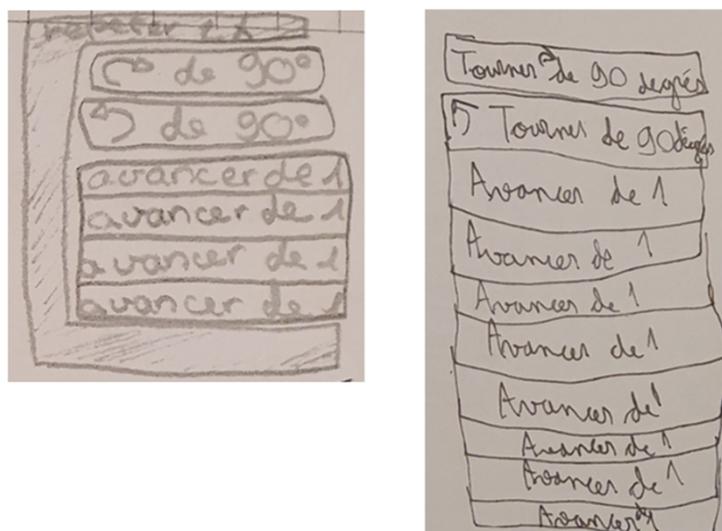


Figure 10. Exemples de propositions d'élèves pour le chemin 1.

Les élèves ont su, à travers leur codage, rendre visible les instructions qui étaient répétées dans une boucle, parfois en proposant de mettre les instructions « à l'intérieur » de la boucle comme sur le logiciel Scratch (figure 10, en bas à gauche), soit en proposant d'autres codages (par exemple, en bas de la figure 9). Les boucles imbriquées ont été spontanément utilisées par certains élèves qui avaient déjà compris lors de la séance précédente que leur utilisation permet de rendre le déplacement plus rapide lors de l'activité 1 (figure 11).

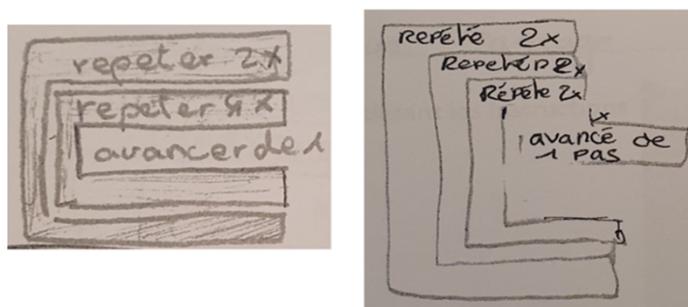


Figure 11. Exemples d'utilisations correctes de boucles imbriquées pour le chemin 1.

Des interprétations non conformes à l'usage des boucles imbriquées en informatique sont visibles, certains élèves additionnent les nombres de répétitions au lieu de les multiplier. Par exemple, dans l'algorithme de la figure 12, l'élève a additionné 3 et 3, ce qui donne 6 et il a ajouté deux instructions "avancer de 1" pour arriver aux huit pas du chemin.

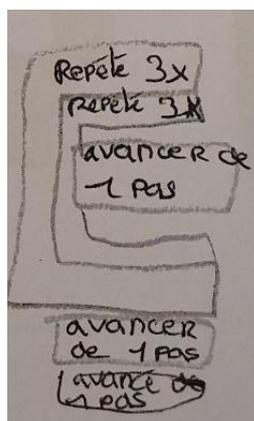


Figure 12. Exemple d'interprétation non conforme des boucles imbriquées pour le chemin 1.

<p>a. Suite d'instructions détaillée</p>	<p>b. Erreurs de décomposition</p>	<p>c. Les instructions dans la boucle sont incomplètes</p>

Figure 13. Exemples de propositions d'élèves pour le chemin 2.

Le deuxième chemin a posé davantage de difficultés et a été source d'erreurs. Dans cette partie, les élèves n'ont plus cherché à varier les propositions mais se sont concentrés sur l'écriture d'une suite d'instructions la plus courte possible. Ayant pris conscience de l'utilité des boucles, ils ont essayé de les utiliser au mieux. Certains d'entre eux ont rencontré des difficultés au moment de la décomposition du chemin en sous-parties, pour identifier le motif répété. Ils ont commencé

par écrire correctement une suite d'instructions détaillée (figure 13a) mais n'ont pas forcément repéré quelles instructions étaient répétées quand ils ont introduit des boucles. Ils ont ainsi utilisé l'instruction < répéter 4 fois > sans voir que le motif n'était pas répété quatre fois en entier (figure 13b). Ce choix constitue une difficulté qui devrait peut-être être évitée pour une première rencontre de motif (à plusieurs instructions). D'autres ont bien identifié que le motif était répété seulement trois fois en entier mais ont omis une partie des instructions permettant de tracer le motif (figure 13c).

Pour le deuxième élève de la figure 13b, on peut observer qu'il a identifié le motif (instructions à l'intérieur de la boucle < répéter 4 X >), cependant la syntaxe qu'il utilise est ambiguë à plusieurs endroits : le < répéter 2 X > est situé après le < avancer 1 > ; au niveau de l'instruction < tourner 90 > l'orientation n'est pas indiquée (tout comme pour l'élève en figure 13a). De plus, cet élève répète quatre fois les instructions et non 3 fois, avec des instructions supplémentaires pour finaliser le chemin. La verbalisation qui accompagne ces algorithmes est importante pour lever les éventuels ambiguïtés, et ensuite se mettre d'accord sur une syntaxe commune.

Quelques élèves ont néanmoins réussi à décomposer correctement ce chemin et ont produit chacun un enchaînement d'instructions valide, utilisant une ou plusieurs boucles (figure 14).

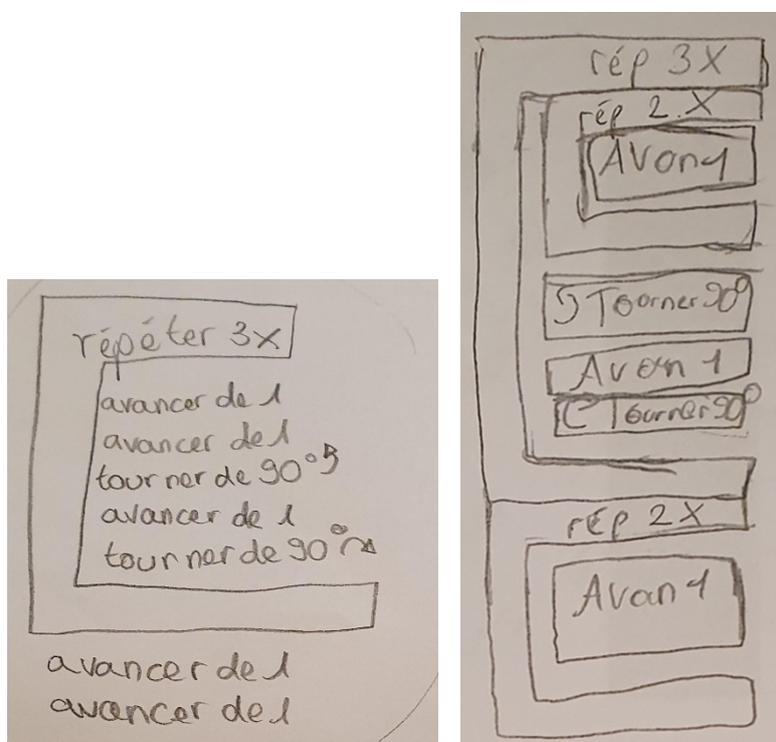


Figure 14. Exemples d'enchaînements d'instructions pour le chemin 2.

Nous avons pu repérer que l'identification d'un motif répété pouvait être source de difficultés pour les élèves. Il pourrait être envisagé un travail en amont en questions flash par exemple sur l'identification de motifs répétés sur un tracé pour que ceci ne soit pas un obstacle lors de cette activité, et un travail complémentaire après l'activité 2 serait pertinent (cf. par exemple les travaux de Peter et al., 2020 ; Leonard et al., 2020 ; Leonard et al., 2022).

3.3.2. Choix des conventions d'écriture

Lors de la conception de la fiche de consignes pour les élèves, notre première idée était de faire coller les cartes < instruction > sur la feuille pour écrire les différents algorithmes. Comme cela nous paraissait chronophage et risquait de nous faire perdre du temps de réflexion, il a été

proposé dans une demi-classe de sixième d'élaborer une convention d'écriture qui permettrait d'écrire plus rapidement les cinq instructions et donc les suites d'instructions, avant de commencer l'activité 2. La construction d'un nouveau codage a suscité un certain nombre de questions et a donné lieu à des discussions intéressantes :

- Vaut-il mieux mettre « $\times 3$ » ou plutôt « $3 \times$ » à la place de l'instruction < répéter trois fois > puisque l'instruction à répéter est écrite après ?
- Comment savoir si la suite d'instructions « $\times 3$, av de 1, av de 1 » signifie qu'il faut répéter trois fois l'instruction < avancer de 1 > puis encore < avancer de 1 > ou répéter trois fois les deux instructions < avancer de 1 > ? Les élèves ont alors eu l'idée de rajouter des parenthèses pour clarifier ce qui fait partie de la boucle.

Finalement, ce qui semblait pouvoir permettre de gagner en rapidité pour la mise en œuvre s'est révélé être une source de difficultés pour les élèves car cela leur demandait de s'approprier un troisième codage, en plus de la langue naturelle et des cartes < instruction > de Scratch. L'activité s'est un peu éloignée de son objectif de discussion sur l'intérêt de l'utilisation des boucles.

Suite à ces constats, lors des expérimentations dans les deux classes de cinquième, l'activité a été proposée en demandant simplement aux élèves de dessiner directement les instructions comme dans le jeu et ainsi de se rapprocher, sans forcément le savoir, d'une présentation comme cela apparaît à l'écran dans le logiciel Scratch en adaptant la taille de l'instruction < répéter... fois > pour entourer plusieurs instructions. Certains élèves ont pris l'initiative d'écrire des instructions simplifiées afin de limiter l'effort d'écriture (figure 15).

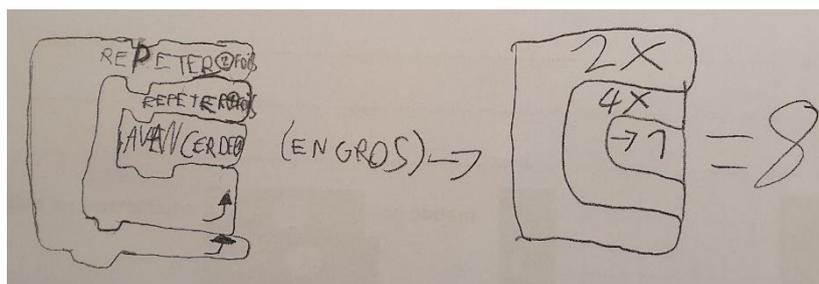


Figure 15. Simplification du codage par un élève.

3.3.3. Institutionnalisation

Nous avons prévu à la fin de l'activité 2 d'institutionnaliser les concepts de boucle et d'algorithme. Avec la demi-classe de sixième, le temps consacré à la réflexion nécessaire à l'institutionnalisation s'est révélé trop court. Le vocabulaire a été noté mais il n'a pas été possible de laisser les élèves réfléchir seuls avant la mise en commun. Dans les classes de cinquième, une vingtaine de minutes ont pu être consacrées à l'institutionnalisation et à l'élaboration d'une trace écrite « à retenir » (annexe 2).

Après la mise en commun des réponses à la première question, les élèves ont constaté qu'il existe plusieurs suites d'instructions qui permettent de tracer un chemin donné. Cela a amené la classe à l'énoncé suivant : « Plusieurs algorithmes différents permettent de résoudre un même problème ».

La classe s'est ensuite intéressée aux boucles. Après réflexion, les élèves ont majoritairement répondu à la question « Que remarques-tu en observant les suites d'instructions qui utilisent le moins de cartes possible ? » par « on utilise des cartes < répéter... fois > ». Nous avons sorti cette observation du contexte pour en conclure que « L'utilisation de boucles permet de diminuer la longueur d'un algorithme ».

La mise en commun a également été l'occasion de reprendre le concept de boucles imbriquées, en travaillant sur un cas concret de répétition (comme par exemple en figure 16).



Figure 16. Dépliage de la boucle externe pour expliquer le fonctionnement des boucles imbriquées.

Il aurait été possible de demander à un élève d'exécuter les instructions en effectuant le nombre de pas demandés ou alors de tracer sur la grille du plateau de jeu le déplacement obtenu à l'issue de chaque répétition de la boucle intérieure.

En conclusion, les élèves se sont impliqués avec beaucoup d'intérêt et leurs différentes remarques, questions, propositions, réponses et productions écrites ont montré qu'ils étaient réellement acteurs de la séance. L'activité 1 s'est révélée indispensable pour que l'activité 2 produise cet effet. En effet, le fait d'avoir déjà manipulé les instructions en matérialisant les chemins correspondants leur a permis de se lancer rapidement dans l'écriture des scripts et de repérer les instructions les plus utiles pour obtenir un script plus court, ce qui était l'objectif final.

Conclusion et perspectives

Les objectifs de cet article étaient de présenter et d'analyser des activités débranchées associées à un jeu, qui permettent de travailler les concepts d'algorithme et d'itération dans le contexte de déplacements à partir des instructions Scratch. Le jeu *Trace ton chemin* permet, dans un cadre ludique, de mettre en lumière la variété d'algorithmes possibles pour effectuer un déplacement (du fait des contraintes engendrées par les cartes que le joueur a en main, sans parler des blocages que peut ajouter l'adversaire) et l'avantage des boucles pour « avancer plus vite » (diminuer le nombre d'instructions pour effectuer un déplacement). Nous constatons ici la pertinence didactique d'aborder le concept de boucle comme une réponse à la recherche d'« économie de la communication homme - machine » soulignée par Nguyen et Bessot (2010, p. 47).

Cela reste à vérifier, mais nous pensons que ce jeu, en amont de toute utilisation du logiciel Scratch, pourrait permettre, au-delà de ce que nous avons identifié dans l'article, une meilleure entrée dans la programmation sur le logiciel. En effet, ces activités permettent une première familiarisation avec les instructions de déplacement (< avancer de 1 pas >, < tourner de 90° >) et de boucle, et une réflexion sur le sens des boucles et des imbrications de boucles.

Une alternative pourrait aussi être d'utiliser le logiciel Scratch en parallèle des activités. En effet, ces activités, comme nous l'avons vu, ne sont pas auto-validantes : par exemple, dans l'activité 1, c'est l'observateur qui joue le rôle de valideur, cependant il peut ne pas être en mesure d'invalider un raisonnement erroné (comme par exemple avec l'imbrication de boucles). Nous ne pensons pas que cela soit nécessaire et cela nous semble être très contraignant voire contre-productif d'utiliser le logiciel Scratch pour vérifier les algorithmes lors de l'activité 1. En revanche, nous pourrions l'envisager lors de l'activité 2 pour visualiser le tracé produit, et ainsi valider que chaque proposition permet bien de tracer le chemin demandé ou dans le cas contraire, permettre des ajustements basés sur la rétroaction que permet alors le logiciel. Nous n'avons pas testé cette modalité.

Au-delà du fait que l'on puisse enrichir l'activité 2 par de nouveaux chemins, plusieurs variantes du jeu pourraient être envisagées. Une variante du jeu Trace ton chemin pourrait être de ne plus tirer les quatre cartes du dessus de la pioche (au hasard) mais de laisser les élèves choisir 4 cartes (faces visibles), ou bien encore de piocher 2 cartes au hasard, et d'en choisir 2 (faces visibles). Cela permettrait de faire émerger de nouvelles stratégies. Une autre variante peut être de modifier la grille de départ, en remplaçant les carrés par pavage formé de triangles équilatéraux. La difficulté se renforce avec l'augmentation du nombre d'instructions : quatre rotations au lieu de deux avec deux angles différents, 60° et 120°, et l'impossibilité de tracer un chemin rectiligne. Le jeu ne peut pas, dans ce cas, être proposé en début de sixième mais il peut être proposé dans une classe de cycle 4.

Il nous semble que nos activités sont complémentaires des propositions présentées dans des ressources d'autres groupes IREM (cf. partie 1.3). Un travail sur le repérage des motifs semble un point à ne pas négliger pour renforcer la maîtrise des boucles. Dans le prolongement de ces deux activités, une nouvelle expérimentation a été menée. Les élèves devaient en même temps concevoir un chemin qui respecte certaines contraintes (points de passage et obstacles) et écrire l'algorithme. Ceci a conforté les résultats des recherches de Léonard et al. (2022) sur les difficultés que peuvent rencontrer les élèves à identifier des motifs.

Références bibliographiques

- Baron G.-L., & Drot-Delange, B. (2016). L'informatique comme objet d'enseignement à l'école primaire française ? Mise en perspective historique. *Revue française de pédagogie*, 195, 51-62.
- Branthôme, M. (2021). Apprentissage de la programmation informatique à la transition collège-lycée. *STICEF (Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation)*, 28 (3), 1-35.
- 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*.
- Jutand, O., & Kirch, C. (2020). Algorithmique débranchée. APMEP, *Au Fil des Maths*, 533.
- Lagrange, J.B., & Rogalski, J. (2017). Savoirs, concepts et situations dans les premiers apprentissages en programmation et en algorithmique. *Annales de didactique et de sciences cognitives*, 22, 119-158.
- Léonard, M., Peter, Y., & Secq, Y. (2020). Reconnaissance et synthèse de motifs redondants avec des élèves de 6-7 ans. MOTIFS.MOTIFS.MOTIFS. \Leftrightarrow 3 x MOTIFS. *Colloque Didapro 8 - DidaSTIC - L'informatique, objets d'enseignements enjeux épistémologiques, didactiques et de formation*, février 2020, Lille.
- Léonard, M., Secq, Y., Peter, Y., & Fluckiger, C. (2022). Pensée informatique : approche didactique de l'identification de motifs. Dans C. Declercq, J. Lehuen, P. Leroux, V. Renault & A. Sejourne. *L'informatique, objets d'enseignement et d'apprentissage. Quelles nouvelles perspectives pour la recherche ? Actes du colloque DIDAPRO 9* (pp. 113-125).
- Ministère de l'Éducation nationale et de l'Enseignement supérieur et de la Recherche (MENESR) (2016). *Algorithmique et programmation*. Eduscol. Consulté le 24 février 2023. <https://eduscol.education.fr/document/17311/download>
- Ministère de l'Éducation nationale et de la Jeunesse et des Sports (MENJS) (2020). Cycle des apprentissages fondamentaux (cycle 2), cycle de consolidation (cycle 3) et cycle des

approfondissements (cycle 4) : modification. *Bulletin officiel n°31 du 30 juillet 2020*.
https://www.education.gouv.fr/pid285/bulletin_officiel.html?pid_bo=39771

Nguyen, C. T., & Bessot, A (2010). Introduire des éléments d’algorithmique et de programmation dans l’enseignement secondaire ? Une ingénierie didactique. *Petit x*, 83, 27-49.

Peter, Y., Secq, Y., Léonard, M. (2020). Reconnaissance de motifs redondants et répétitions : Introduction à la Pensée Informatique. *STICEF (Sciences et Technologies de l’Information et de la Communication pour l’Éducation et la Formation)*, 27(2).

Annexe 1. Enoncé de l'activité 1 distribué aux élèves

Trace ton chemin !

Objectif du jeu : être le premier à atteindre le bord opposé du plateau de jeu en traçant un chemin qui suit les instructions de programmation.

Règles du jeu : Chaque joueur choisit son point de départ et son orientation. Il dispose de 4 cartes d'instructions visibles devant lui, les cartes restantes constitueront la pioche.

À son tour :

- le joueur pose au maximum 4 cartes d'instructions, trace sur le plateau le chemin correspondant puis indique son orientation finale.
- Si son chemin est validé par l'adversaire, les instructions utilisées sont alors déposées sur un autre tas à côté de la pioche.
- Le joueur re-pioche le nombre nécessaire d'instructions pour en avoir à nouveau 4 devant lui.

C'est alors au joueur suivant de commencer son tour.

Attention :

- Il est interdit de croiser le chemin du joueur adverse mais on peut croiser son propre chemin.
- Il est obligatoire de laisser au moins un accès au côté opposé à chaque joueur.

Annexe 2. Trace écrite

A retenir :

L'instruction



s'appelle **une boucle**.

Pour réaliser un déplacement, on utilise une suite d'instructions, c'est **un algorithme**.

Plusieurs algorithmes différents permettent de tracer un même chemin.
L'utilisation de boucles permet d'obtenir un algorithme plus court.

Si besoin, on peut placer une boucle à l'intérieur d'une autre boucle.

A chaque passage dans la 1^{ère} boucle, la 2^{ème} boucle est exécutée entièrement.

Exemple :



Cela revient à avancer de $2 \times 3 \text{ pas} = 6 \text{ pas}$.

